

Dissertation zur Erlangung des Doktorgrades der Fakultät für
Angewandte Wissenschaften der Albert-Ludwigs-Universität
Freiburg im Breisgau

Mapping with Mobile Robots

Dirk Hähnel



December 2004

Betreuer:
Prof. Dr. Wolfram Burgard

Dekan der Fakultät für Angewandte Wissenschaften:
Prof. Dr. Jan G. Korvink

1. Gutachter: Prof. Dr. Wolfram Burgard
2. Gutachter: Prof. Dr. Bernhard Nebel

Tag der Disputation: 3. Februar 2005

ZUSAMMENFASSUNG

Einer der entscheidenden Faktoren bei der Konstruktion eines vollständig autonomen mobilen Roboters ist das Erstellen von Karten. Diese Arbeit bietet Lösungsmöglichkeiten für unterschiedliche Fragestellungen, die beim Kartenbau auftreten.

Üblicherweise beinhalten Karten nur die Beschreibungen von statischen Objekten in der Umgebung, wie zum Beispiel Wände, Gebäude oder Straßen. Dynamische Elemente, umhergehende Menschen oder ähnliches sollen im Allgemeinen nicht in Karten auftauchen. In dieser Arbeit stellen wir Verfahren vor, die es erlauben, die Messungen, die durch diese dynamischen Objekte verursacht werden, zu erkennen und zu filtern. Neben der Reduzierung von unerwünschten Gegenständen in der resultierenden Karte kann dadurch auch die Genauigkeit der Karte erhöht werden. In zahlreichen Experimenten wird demonstriert, dass die gefundenen Methoden effektiv Messungen von diversen dynamischen Objekten, wie zum Beispiel von Personen oder Autos detektieren und filtern können.

Der inkrementelle Vergleich von Abstandsmessungen, das so genannte “Incremental Scan Matching”, ist ein elementarer Bestandteil vieler hier gezeigter Ansätze. Diese Methode ermöglicht die Korrektur lokaler Fehler und erlaubt die Erzeugung örtlich konsistenter Karten. Kleinere Ungenauigkeiten in der Positionsbestimmung können sich allerdings über die Zeit akkumulieren und es kann daher leicht passieren, dass eine globale Inkonsistenz eintritt, sobald ein zuvor gesehener Bereich wieder betreten wird. Das korrekte Schließen von solchen Zyklen ist eine der großen Schwierigkeiten beim Kartenbau. Hierbei muss der Roboter den akkumulierten Positionsfehler erkennen und korrigieren, auch wenn dieser Fehler erheblich sein sollte. Für diese Problematik werden zwei unterschiedliche Lösungen vorgestellt. Die Erste basiert auf dem Einsatz von Rao-Blackwellized Partikelfiltern, der allerdings die Notwendigkeit von erheblichen Berechnungen impliziert. Es wird gezeigt, dass durch eine Kombination mit dem vorgenannten Scan Matching, eine Realisierung in Echtzeit möglich wird. Daneben wird ein Verfahren vorgestellt, das Kartenabschnitte kombinatorisch verwaltet und Beziehungen zwischen den Abschnitten nachträglich ändern kann, sofern ein Fehler er-

kannt wird. In Experimenten wird gezeigt, dass mit Hilfe dieser Methoden Karten von Umgebungen mit zahlreichen, auch verschachtelten Schleifen erstellt werden können.

Neben dem Problem der Erstellung von geometrischen Karten werden in der vorliegenden Arbeit Techniken zur Ermittlung von Position und Form einzelner Objekte vorgestellt. Zur eindeutigen Identifizierung der Objekte werden in der Praxis RFID-Systeme immer populärer. Die genaue Ortung der so genannten RFID-Marken ist jedoch schwierig, da sie ihrerseits keine Abstandsinformationen liefern und der Detektionsbereich der Antennen mehrere Meter betragen kann. Es wird ein Verfahren gezeigt, das mit Hilfe von mobilen Robotern detaillierte Karten dieser RFID-Marken erstellt. Eine andere Technik ermöglicht es, aus Teilansichten eines deformierbaren Objekts ein komplettes Modell zu generieren. Zur Berechnung der Deformation werden die genauen Konfigurationen von beweglichen Verbindungen zwischen benachbarten Messpunkten herangezogen. Um das resultierende Optimierungsproblem zu vereinfachen, wird eine hierarchische Darstellung benutzt. Experimentelle Resultate zeigen die Realisierbarkeit des Ansatzes.

Schließlich werden, als Anwendungen der dargestellten Techniken, zwei komplexe Systeme vorgestellt, die in der Lage sind, autonom ihre Umgebung zu modellieren. Das erste System, das entwickelt wurde, um dreidimensionale Karten von stillgelegten Minen zu erstellen, ist ein 800 kg schwerer Roboter mit dem Namen "Groundhog". Das Basisnavigationssystem beinhaltet ein Softwarepaket, das das Problem des Kartenbaus mittels zweidimensionaler Karten löst. Daneben wurde ein mit unterschiedlichen Sensoren ausgestatteter Modellhelikopter entwickelt, der eine dreidimensionale Karte eines Geländes erzeugen kann. Ein zweidimensionales Lasermesssystem, das senkrecht zur Flugrichtung orientiert ist, erlaubt es, in Echtzeit Karten zu generieren.

ACKNOWLEDGMENT

First and foremost I would like to thank my advisor, Professor Wolfram Burgard, for his continued support and guidance of both my research and career. His advice has been a great help, and his assistance during the preparation of this thesis was extremely generous. I also want to express my gratitude to the committee members for reviewing my dissertation.

Many people have helped me with the accomplishment of this thesis. I want to thank Professor Sebastian Thrun and Professor Dieter Fox for the hospitality, while visiting Pittsburgh, Stanford, and Seattle respectively. I have learned a lot from their extensive knowledge and their brilliant and creative ideas. I thank Sebastian for giving me the opportunity to be part of several very exciting projects like the mine mapping robot or the Helibot project.

There are many people from whom I have benefited also beyond academics. I thank my colleagues Maren Bennewitz, Cyril Stachniss, and Rudolph Triebel for years of fruitful discussions, advice and support. They helped me to succeed in the daily struggle at the University of Freiburg. I would like to thank Dirk Schulz for his help and the great time we shared in the projects. I thank Mike Montemerlo and Nick Roy. Their kind and all-aspect help has made my visit in Pittsburgh a good memory in my life.

I would like to thank to William “Red” Whittaker, Scott Thayer, David Ferguson, Aaron Morris, and all the other Groundhog team members for assistance and support on the Groundhog project. I thank Mark Diel and Andrew Y. Ng for the chance to “play” with an autonomous flying helicopter. Thanks also go to the members of the Intel Research Lab in Seattle, who provided me an excellent environment for my research.

I thank Susanne Bourjaillat for her kind help with frequent office affairs, Michael Keser for the IT-support, and Dirk Zitterel for helping me to fix our robot Albert.

Last but not least, I want to express my gratitude to my parents and my sisters, who have been enormously supportive of me for my entire life. This thesis is as much theirs as it is mine.

DEDICATION

To my parents, for their guidance, support, and encouragement.

NOTATION

x_t	state at time t (often pose (x, y, θ))
u_t	motion at time t
z_t	measurement or measured feature at time t
$m(x_{1:t}, z_{1:t})$ or m	map
$\Theta_t = \theta_t^1, \dots, \theta_t^n$	feature set at time t
$c_{t,n}$	binary correspondence variable

CONTENTS

1	Introduction	1
1.1	Fundamental Problems of Mapping with Mobile Robots	1
1.2	Structure of the Thesis	6
1.3	Publications	7
2	Map Representations and Scan Matching	11
2.1	Map Representations in Robotics	11
2.1.1	Geometric Representations	11
2.1.2	Topological Representations	13
2.1.3	Evidence Grids	14
2.1.4	Comparison between Map Representations	16
2.2	Learning Grid Maps	17
2.2.1	Occupancy Probability Grid Maps with Bayesian Update .	17
2.2.2	Computation of the Occupancy Probability	19
2.2.3	Learning Grid Maps using the Counting Model	21
2.2.4	Fast Approximation using End-Points	25
2.3	Incremental Scan Matching	27
2.3.1	Probabilistic Matching for Scan Registration	28
2.3.2	Two-dimensional Scan Alignment with Occupancy Grid Maps	29
2.4	Scan Registration in 3D	30
2.4.1	The Iterative Closest Point (ICP) Algorithm	31
2.4.2	Probabilistic Three-dimensional Scan Registration using Geometric Maps	36

3 Map Building in Populated Environments	39
3.1 Introduction	39
3.2 Tracking People in Range Scans	40
3.2.1 Sample-based Joint Probabilistic Data Association Filters (SJPDAFs)	40
3.2.2 Dealing with Variable Numbers of Persons	41
3.2.3 Laser-based Implementation	43
3.3 Computing Consistent Maps	44
3.3.1 Integrating People Tracking Results into the Map Building Process	45
3.4 Experiments	47
3.4.1 Learning 2D Maps	47
3.4.2 Improved Robustness	49
3.4.3 Learning 3D Maps	50
3.5 Related Work	53
3.6 Summary	54
4 Mapping in Dynamic Environments	55
4.1 Introduction	55
4.2 Learning Maps in Dynamic Environments	56
4.3 Experiments	63
4.3.1 Filtering People	63
4.3.2 Improved Localization Accuracy	64
4.3.3 Generating Large-Scale Outdoor Maps	65
4.3.4 Generating Textured 3D Maps	66
4.3.5 Extracting Textured 3D Objects	68
4.4 Related Work	69
4.5 Summary	71
5 Mapping with Rao-Blackwellized Particle Filters	73
5.1 Introduction	73
5.2 Mapping with Rao-Blackwellized Particle Filter	74
5.3 Combination with Scan Matching	76
5.4 Experimental Results	79
5.4.1 Mapping Large-Scale Environments with Multiple Cycles	81
5.4.2 Comparison to Previous Online Techniques	81
5.5 Related Work	84
5.6 Summary	85

6 Lazy Data Association in SLAM	87
6.1 Introduction	87
6.2 Preliminaries	88
6.2.1 SLAM with Known Data Association	88
6.2.2 Incremental Data Association	89
6.3 Lazy Data Association	91
6.3.1 Tree Search	91
6.3.2 Equivalency Constraints	91
6.3.3 Recovering the Path Posterior under Equivalency Constraints	92
6.3.4 Incorporating Negative Measurement Information	94
6.4 Experimental Results	94
6.5 Summary	98
7 Mapping and Localization with RFID Technology	99
7.1 Introduction	99
7.2 Learning a Probabilistic Sensor Model for the RFID Antenna	102
7.3 Mapping RFID Tags	104
7.4 Localization with RFID Tags	105
7.5 Experimental Results	106
7.5.1 Mapping RFID tags	107
7.5.2 Localization with RFID Tags	108
7.5.3 Improving Global Localization with RFID Tags	110
7.6 Related Work	111
7.7 Summary	112
8 Modeling Nonrigid Objects with Mobile Robots	113
8.1 Introduction	113
8.2 ICP Variant for 3D Scan Registration	114
8.2.1 Three-dimensional Scans taken from a Mobile Robot	115
8.2.2 Measurement Model	116
8.2.3 Registration as Likelihood Maximization	117
8.2.4 Optimization Procedure	118
8.3 Recovering the Surface Configuration of Nonrigid Objects	121
8.3.1 Links	121
8.3.2 Target Function	123
8.3.3 Optimization Procedure	123
8.3.4 Efficient Variable Resolution Optimization	124
8.4 Setup and Experimental Results	125
8.5 Summary	128

9	Mapping Applications with Autonomous Robots	129
9.1	Mine Mapping	129
9.1.1	Introduction	129
9.1.2	Robotic System for Mine Mapping	131
9.1.3	SLAM Technique in Mines	134
9.1.4	Mine Mapping Experiments	134
9.1.5	Summary	137
9.2	Helicopter Mapping	139
9.2.1	Introduction	139
9.2.2	3D Modeling Approach	143
9.2.3	Experimental Results	146
9.2.4	Summary	149
10	Conclusions	151

CHAPTER 1

Introduction

1.1 Fundamental Problems of Mapping with Mobile Robots

This dissertation explores fundamental issues of mapping with mobile robots. Maps are an essential part for safe and efficient navigation and can be preconditions for other techniques like, for example, localization. A successful realization of the map learning process has to address several sub-problems, including building globally consistent maps with nested loops and dealing with dynamic environments.

To understand these problems, we give an simplistic example in the following paragraphs. In this example, a human is faced with the problem to create a map of a central train station and its close surrounding. The problems encountered by this person typically also occur when creating maps with mobile robots.

Consider a person gets the task to draw a detailed map of a central train station and its close surrounding. Let us assume that after he arrives at the train station he starts his work directly from this location. Obviously he cannot see the whole area from one position, therefore he has to create the map from observations received at several places. After describing the part of the environment, which is visible from the first place, he moves to another position and continues his work from there. After a while he leaves the hall and explores the area outside the building. Then he expands this region to the streets going to the train station and finally he returns to the central station. Mainly we will address here three important challenges in robotic mapping.

Typically, the person builds the map from observations perceived at several

places. He starts from his initial pose and changes his position to incrementally enlarge the map by adding new information. In principle he could use every observation to create a local map of his current vicinity. He could do this for all the places, but at the end he wants to have one single big map instead of a collection of small maps. This raises the problem of how to combine the local maps to a larger consistent map. One way to solve this problem is to fit the local observations to the previous ones and to integrate both parts into a common map. This process can easily be repeated to enlarge the map further and further. Because of its stepwise nature this approach is usually called *incremental mapping*. In this algorithm we combine a new observation by finding the position of the measurement, which best fits in the map constructed so far. In robotics the observations are often so called *distance scans*, which are a combination of multiple distance measurements. Therefore this technique of incremental fitting or matching of the measurements will be denoted as *incremental scan matching*.

Beside this, another problem increases the difficulty to create maps. The human is probably not alone in the central station and a lot of other people are walking by. Typically, maps only include *static* parts of the world, like walls, buildings, or streets. Therefore, all dynamic parts like other people have to be neglected. Unfortunately the people can disturb his observations, so that he carefully has to distinguish between static objects and dynamic ones. In our example, most of the dynamic elements are humans, but they are not the only ones. If the person builds the map of the central station then, after removing all the effects caused by the other people, there will still be other dynamic objects in his drawing. For example he does not want to include the cars, which are parking in front of the station. At a later time, when he visits the place again, they are probably gone. We can find more examples like this, which are less obvious. Tables and chairs in front of a cafe or sometimes the booths of small traders can change their places. In the context of robotics this problem is often described as *mapping in dynamic environments*. We have to decide carefully which measurements are caused by dynamic objects and therefore should not be included in our map. The example with the parking cars shows us that in robotics this problem can be extremely hard. Typically the robot does not recognize a detected object. Most times it can only detect the shape or color. So it does not have the knowledge of the properties of objects and like in our examples, it does not know that the big object in front of it is a parked car which will likely be gone at later points in time.

Assuming the person can handle all of these difficulties and permanently increases the size of the map by adding local observations to it. In our example he will leave the building and perhaps he will go around it. However, at one point in time he will return to a previous seen area. Unfortunately we cannot expect that he perfectly draws the map. He has to live with the fact that the map contains slight errors which accumulate over time. What easily can happen is that

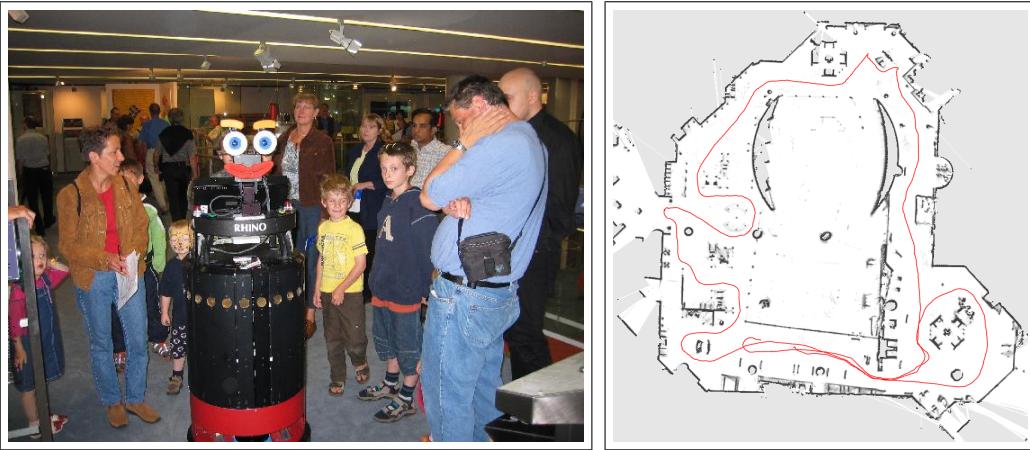


Figure 1.1: The left image shows the mobile robot Rhino in the Deutsches Museum Bonn. The robot is usually surrounded by people. Maps should only contain static parts of the world, therefore all measurement of the people has to be filtered. The right image show the map of the museum created by the robot. Dark areas means that this space is occupied by an object, white means that the space is free and usually save for navigation. The red lines shows the way of the robot collecting the data to create this map.

these accumulated errors lead to a global inconsistency when the person reenters a previously seen area. When he identifies a place seen before, his position in the current map should be identical to that of the previous observation. Note that it is often easy for humans to recognize a previously seen building. However, we also sometimes get lost in unknown cities and pass the same locations several times before we recognize them. Especially if places look alike it is very difficult to decide whether we have seen them before or whether they only look very similar to the previous ones.

In robotics this problem occurs when the robot closes a loop. A lot of techniques can close small loops, but when the loop gets larger, many of these become unreliable. For example, our incremental scan matching technique is not the appropriate way to handle large loops. Due to its incremental nature and its inability to retrospectively correct errors this approach is prone to failure when it comes to the task of handling large cyclic environments. Therefore, we need an alternative method for this problem. In the context of robotics the problem of closing large loops becomes especially difficult as the perception is more limited than visual system of humans and many places appear similar. The assignment of places to previous seen ones is known as *data association*. Data association can be, for example, between locations but also between objects that we have seen more than



Figure 1.2: Different application of robotic mapping: the robot “Dumbo” should detect the locations of objects, which are labeled with special tags. The white squared panels are additional antennas to detect these tags. The robot itself is a standard robotic platform used by many universities and research centers.

once. The problem is to decide whether a given assignment is a good choice or whether there are better ones. Dealing with multiple potential data associations can be done in a *pro-active* way by maintaining multiple hypotheses or in a lazy way by maintaining a single hypothesis and by changing the assignments retrospectively and only when an error is detected.

A successful and efficient system for mapping should address the following aspects.

- During the process we need an efficient technique that can combine successive measurements.
- People and other dynamic objects should be detected as they can disturb the mapping process and should not be included in the map.
- Whenever the robot returns to a previous seen area the algorithm should detect this and potentially correct errors in the map and its position.
- The detection and correction of previously seen places can fail. Therefore the process of data association should be correctable and decisions should be changeable at later times.

Additionally, we do not want to restrict ourselves to maps with detailed geometric descriptions of all objects in the environment. Sometimes we are only interested in places of a certain sets of objects of interest. This problem of mapping the locations of these objects would be straightforward if we could detect them easily in our regular mapping process. Unfortunately sometimes we



Figure 1.3: The left image shows the “Groundhog” robot, a 1,500 pound custom-built vehicle equipped with on-board computing, laser range sensing, gas and sinkage sensors, and video recording equipment. The right image shows a instrumented helicopter platform. The system has a modified laser range finder, GPS, compass, and an inertial sensor to measure the position. The system is equipped with on-board data collection and processing capabilities and a wireless digital link to the ground station.

cannot tell the exact location of the origin of an observation. For example, if our person in the central station hears his name called by a friend, he immediately wants to figure out the exact position of his friend. When the hall is crowded he cannot see the friend directly, so he has only a direction of the friend’s position. Moreover echos in halls can lead him to a completely wrong direction. Although he knows his own position he cannot easily estimate the location of his friend. He probably needs several observations from different locations to find him. We can observe a similar characteristic of this behavior in radio frequency identification technologies (RFID). These systems mainly consist of two parts, the marker and the reader. The markers can be uniquely identified by the reader, but the precision of their locations cannot be determined up to the vicinity of the used antenna. In recent years automatic identification procedures have become very popular in many service industries, purchasing and distribution logistics, industry, manufacturing companies and material flow systems. Due to the technological improvements in this field the detection range of these systems could be enlarged from a few centimeters to several meters in the last years. Because of the procedures used for the transfer of power and data, contact-less ID systems are called RFID systems (Radio Frequency Identification). RFID is a proven technology that has been around since the 1970s. Up to now, it has been too expensive and too limited to be practical for many applications. There are several methods of identification, but the most common is to store a serial number that identifies a person or object, and perhaps other information, on a microchip that is attached to an antenna (the chip

and the antenna together are called an RFID tag). The antenna enables the chip to transmit the identification information to a reader. The RFID tag reader can read the information from a certain distance. Typically we can assume that the place of the RFID tag will be in the vicinity of the machine’s antenna. Unfortunately this can be a region of several meters and sometimes, in the case of reflections, the place of the RFID tag can be outside the usual vicinity of the antenna. This shows that building maps of special objects like these RFID tags can be a difficult problem.

Second, there are objects that can change their shape although they are not typically dynamic. Trees, for example, will not change the location, but their branches can move. We have to find a way to deal with such *non-rigid* objects. We need a special model for describing typical deformation of objects. The branch from a tree can change the position, but the overall structure will not change so much. We need a special technique which can integrate multiple views of such non-rigid objects.

In this thesis we address several of the aspects of the mapping problem which have been described above. We extensively validated all techniques on different mobile robotic platforms. Additionally we present two applications of systems for mapping with mobile robots. The first application is the robotic system, designed to autonomously explore and acquire 3D maps of abandoned mines, is a 1,500 pound vehicle, nicknamed “Groundhog.” Groundhog acquires large-scale consistent maps of the voids it explores. The second application shows the feasibility of acquiring high-resolution ground models using active laser range sensors on a low-flying helicopter platform. Low-flying air vehicles, such as helicopters, promise to overcome limitations which are inherent to ground-based vehicles: they are much less constraint than ground vehicles with regards to their navigational capabilities, yet they can fly low enough to acquire data of vertical structures at high resolution.

1.2 Structure of the Thesis

The remainder of the this thesis is organized as follows. In the next chapter we describe different map representations in robotics and we introduce an incremental scan matching technique. In Chapter 3 we present a combination of this mapping approach with a people tracking technique. Our algorithm tracks people in the vicinity of the robot. The belief about the potential positions of persons is used to improve the registration process and to remove spurious objects from the map being learned. In Chapter 4 we introduce a more general way for filtering dynamic elements, which is not only restricted to humans. After this we focus on the problem of loop closure in Chapter 5. We present an efficient way to build maps

of large environments with multiple loops. In Chapter 6 we introduce a lazy data association technique that can “repair” past data association techniques back into the past. In Chapter 7 we investigate how RFID technology can be enhanced by location information. We use a mobile robot equipped with RFID antennas to determine the locations of RFID tags attached to objects in an indoor environment. Then we propose an approach suited for scan registration and 3D modeling of nonrigid objects in Chapter 8. Finally, in Chapter 9 we present two applications of the proposed techniques.

1.3 Publications

Parts of thesis have been published in the following articles, conference and workshop papers:

REFEREED JOURNAL/MAGAZINE ARTICLES

- P. Trahanias, W. Burgard, A. Argyros, D. Hähnel, H. Baltzakis, P. Pfaff, and C. Stachniss. Tourbot and webfair: Web-operated mobile robots for tele-presence in populated exhibitions. *IEEE Robotics and Automation Magazine*, Special Issue on Robotics & Automation in Europe: Projects funded by the Commission of the European Union. Forthcoming.
- S. Thrun, S. Thayer, W. Whittaker, C. Baker, W. Burgard, D. Ferguson, D. Hähnel, M. Montemerlo, A. Morris, Z. Omohundro, C. Reverte, and W. Whittaker. Autonomous exploration and mapping of abandoned mines. *IEEE Robotics and Automation Magazine*. Forthcoming.
- D. Hähnel, D. Schulz, and W. Burgard. Mobile robot mapping in populated environments. *Advanced Robotics*, 17(7), 2003.
- W. Burgard, P. Trahanias, D. Hähnel, M. Moors, D. Schulz, H. Baltzakis, and A. Argyros. Tele-presence in populated exhibitions through web-operated mobile robots. *Autonomous Robots*, 15, 2003.
- S. Thrun, C. Martin, Y. Liu, D. Hähnel, R. Emery-Montemerlo, D. Chakrabarti, and W. Burgard. A real-time expectation maximization algorithm for acquiring multi-planar maps of indoor environments with mobile robots. *IEEE Transactions on Robotics and Automation*, 2003.
- D. Hähnel, W. Burgard, and S. Thrun. Learning compact 3d models of indoor and outdoor environments with a mobile robot. *Robotics and Autonomous Systems*, 2002.

REFEREED CONFERENCE ARTICLES

- D. Hähnel, W. Burgard, D. Fox, K. Fishkin, and M. Philipose. Mapping and localization with rfid technology. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2004.
- C. Stachniss, D. Hähnel, and W. Burgard. Exploration with active loop-closing for FastSLAM. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2004.
- C. Baker, A. Morris, D. Ferguson, S. Thayer, C. Whittaker, Z. Omohundro, C. Reverte, W. Whittaker, D. Hähnel, and Thrun. S. A campaign in autonomous mine mapping. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2004.
- S. Thrun, Mark Diel, and D. Hähnel. Scan alignment and 3d surface modeling with a helicopter platform. In *Proceedings of the International Conference on Field and Service Robotics (FSR)*, 2003.
- D. Ferguson, A. Morris, D. Hähnel, C. Baker, Z. Omohundro, C. Reverte, S. Thayer, W. Whittaker, W. Whittaker, W. Burgard, and S. Thrun. An autonomous robotic system for mapping abandoned mines. In *Proceedings of Conference on Neural Information Processing Systems (NIPS)*, 2003.
- D. Hähnel, W. Burgard, B. Wegbreit, and S. Thrun. Towards lazy data association in slam. In *In Proceedings of the 10th International Symposium of Robotics Research (ISRR)*, 2003.
- D. Hähnel, S. Thrun, and W. Burgard. An extension of the ICP algorithm for modeling nonrigid objects with mobile robots. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*, 2003.
- D. Hähnel, W. Burgard, D. Fox, and S. Thrun. An efficient FastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2003.
- D. Hähnel, R. Triebel, W. Burgard, and S. Thrun. Map building with mobile robots in dynamic environments. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2003.
- S. Thrun, D. Hähnel, D. Ferguson, M. Montemerlo, R. Triebel, W. Burgard, C. Baker, Z. Omohundro, S. Thayer, and W. Whittaker. A system

for volumetric robotic mapping of abandoned mines. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2003.

- D. Hähnel, D. Schulz, and W. Burgard. Map building with mobile robots in populated environments. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2002.

REFEREED SYMPOSIUM/WORKSHOP ARTICLES

- S. Thrun, D. Hähnel, and W. Burgard. Modeling nonrigid objects from range data. In *Proceedings of the Snowbird Workshop 'Machines That Learn'*, Snowbird, UT, 2003. NIPS Foundation. Extended abstract.
- W. Burgard, D. Hähnel, R. Triebel, and S. Thrun. Mapping with mobile robots in dynamic environments. In *Proceedings of the Snowbird Workshop 'Machines That Learn'*, Snowbird, UT, 2003. NIPS Foundation. Extended abstract.
- W. Burgard, P. Trahanias, D. Hähnel, M. Moors, D. Schulz, H. Baltzakis, and A. Argyros. Tourbot and webfair: Web-operated mobile robots for telepresence in populated exhibitions. In *In Proc. of the IROS 02 Workshop on Robots in Exhibitions*, 2002.
- D. Hähnel and W. Burgard. Probabilistic matching for 3d scan registration. In *In Proc. of the VDI-Conference Robotik 2002 (Robotik)*, 2002.
- D. Hähnel, W. Burgard, and S. Thrun. Learning compact 3d models of indoor and outdoor environments with a mobile robot. In *Proc. of the European workshop on advanced mobile robots (EUROBOT)*, 2001.

CHAPTER 2

Map Representations and Scan Matching

2.1 Map Representations in Robotics

In this section we give a short survey of map representations often found in the robotic literature. Among the huge variety of different representations three concepts prevail: geometric maps, topological maps, and evidence grids.

2.1.1 Geometric Representations

Geometric models use geometric primitives, like points, lines, etc., for representing the environment. Introducing new primitives for every measurement (see the left image in Figure 2.1) constantly increases the complexity of the map. So we need an another representation of the environment with a smaller number of primitives, so called features. Mapping then amounts to estimating the parameters of the primitives as to best fit the observations. In the past different geometric representations have successfully been used. They typical consist of compact and efficient representations of an environment. In terms of spatial information content, feature maps are limited to parametric landmarks or modeled objects (such as points, lines, planes, and curves). An example of a two-dimensional feature map is presented in the right image of Figure 2.1.

The geometric representation does suffer from not being able to represent more complex environments, such as the space in between the features, and natural structures. Most geometric representations do not explicitly model the free space. Furthermore they usually can only approximation of natural structures.

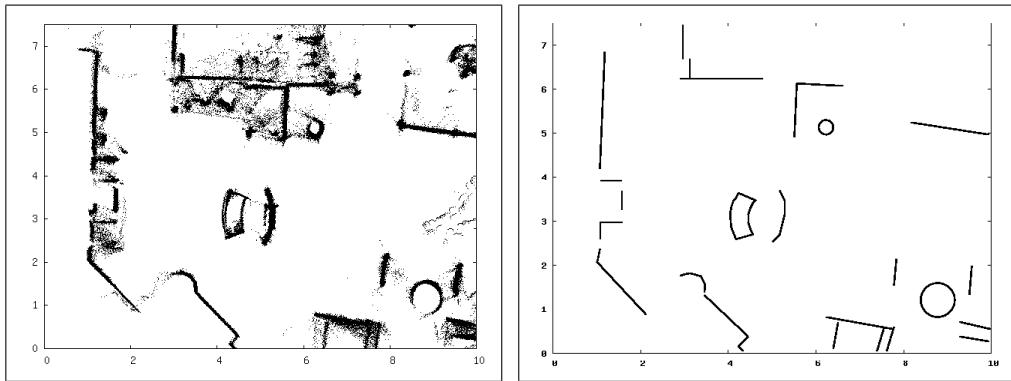


Figure 2.1: Two-dimensional geometric maps. The left image shows a part of a map in which every observation is described by a point. Therefore the resulting map includes a large number of points. The right image shows the same environment described by a few line segments and circles. This map is more compact, but lacks some of the details.

Higher dimension parametric models, on the other hand, are typically employed for approximating more complex features. Many of these models for representing complex structures from spatial data have been developed in the field of computer graphics and are usually computationally expensive. Consequently, such models can be inappropriate representations for time critical, incremental map building algorithms. Furthermore, these models are often limited in representing the uncertainty.

In robotics there is a wide variety of algorithms which deals with geometric feature maps. Since the measurements usually do not provide features directly from the observations, we need techniques to detect them. According to the time restriction for online systems this detection can only be done for relatively simple features. Often line segmentation or edge detection is used for this purpose. These features are easy to extract and can be found frequently in man-made environments. The description of an environment with, for example, line-segments can be very compact, but the representation suffers essentially from two problems. First, many environments can be very complex and it is in advance not known which set of geometric primitives is appropriate for describing the scene. While in a typical office environment line segments are often very successful, the same technique can fail for other structures with less straight parts. Second, the sensor does not always directly provide the features. Therefore the algorithm has to extract them from one or multiple measurements. Depending on the complexity of the feature and on the quality of the sensor this can be error-prone, why it may result in missing or wrong features.

On the other hand we cannot always avoid feature detection. As an example, camera images cannot be used directly for map building. Since the pixels of an image do not provide a three dimensional location, this information has to be extracted from a sequence of images. To achieve this we need to detect the same feature in several images and we need to know the location where the pictures were taken. This is a complex problem even if the association of the features is always correct. If we allow that the association is incorrect, this problem becomes even harder. As we often cannot distinguish features, the detection and association can be unreliable and is many times restricted to special environments.

In certain situations geometric representations are a very compact way to describe the environment. A few features, structures, or objects can represent the environment in a very compact way. Feature detection can be very difficult, especially in the case of more complex structures. The detection rate and the quality often depends on assumptions about the environment. For example, if we assume that the robot operates in a typical office environment, we probably get good results with line or edge detection. In general, however, we do not know this in advance. Feature detection will fail, if we look for the wrong features or if they are too difficult to extract. On the other side, representing every measurement by a single geometric primitive leads to a large number of geometric objects. Then the number of primitives outruns very quickly the capability of the computer.

Another problem is that the uncertainty in the detection is often only incorporated in the estimation of the geometric object. Once the geometric primitive or feature is selected and the location is estimated the uncertainty is typically not be represented any longer.

2.1.2 Topological Representations

Topological maps, based on models like the TOUR model [Kuipers, 1978], vary from evidence grids and geometric maps in the sense that they represent an environment without implicit use of metric information. Spatial information is given instead through the notion of distinctive places. Some examples of typically used distinctive places in topological representations are geometric features like Equal-Distance-to-Near-Objects [Kuipers and Byun, 1991] or rooms, and corridors [Chatila and Laumond, 1985]. Topological maps are generally represented as graphs, where nodes represent distinctive places and arcs that connect nodes represent path information between places. An example of a topological map is shown in Figure 2.2. With topological maps, algorithms such as localization and navigation are performed by the use of place recognition techniques. Topological maps are more efficient representations for structured environments, where distinctive places are more frequent. Conversely, in unstructured environments, where place recognition is more complex, a robot using purely topological in-

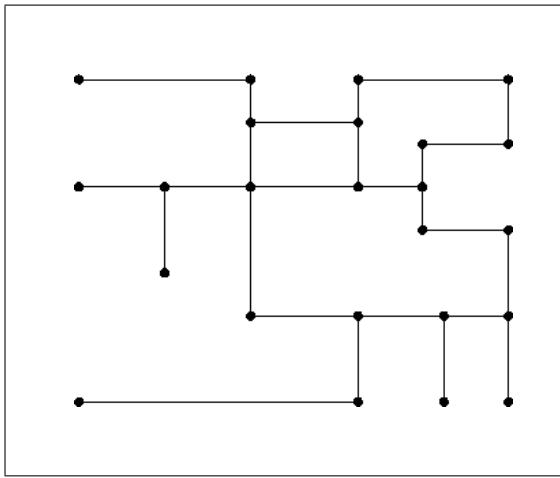


Figure 2.2: A two-dimensional topological map. Nodes represent distinctive places and arcs that connect nodes represent path information between places.

formation for localization can easily become lost. Given the absence of metric information, location uncertainty is represented as the uncertainty of being at a distinctive place. The ability of localization is highly dependent on how reliable the place recognition technique employed can distinguish and match sensor data to stored topological information. Hybrid topological-metric maps combine topological information with metric information. This enables the map representation to benefit from the efficiency of topological maps, and from the spatial consistency and accuracy of metric maps.

2.1.3 Evidence Grids

Two dimensional evidence grids were introduced by Elfes and Moravec [Moravec and Elfes, 1985]. In this representation, the environment is subdivided into a regular array or a grid of rectangular cells. The resolution of the environment representation directly depends on the size of the cells. In addition to this discretization of space, a probabilistic measure of occupancy is associated with each cell. This measure takes on any real number in the interval $[0, 1]$ and describes one of the two possible cell states: occupied or unoccupied. The occupancy probability of 0 means definitely unoccupied space and the probability of 1 means definitely occupied space. A value of 0.5 refers to an unknown state of occupancy.

Many methods have been employed for updating the state of occupancy for each cell, such as Bayesian [Matthies and Elfes, 1988, Elfes, 1992], Dempster-Shafer [Gambino *et al.*, 1997, Pagac *et al.*, 1998], and Fuzzy Logic [Oriolo *et al.*, 1997]. An example of an occupancy grid map representation is shown in



Figure 2.3: Occupancy grid map: In this representation, the environment is subdivided into a regular array or a grid of rectangular cells. A probabilistic measure of occupancy is associated with each cell, which describes one of the two possible cell states: occupied or unoccupied. In this image, black corresponds to occupied areas and white corresponds to unoccupied areas.

Figure 2.3, where white regions correspond to occupancy probability of 0, and black regions to a probability of 1. The evidence grid is an efficient approach for representing uncertainty and for fusing multiple sensor measurements. It is also ideal for incorporating different models of sensor uncertainty. Until recently, evidence grids were only considered suitable for map building given robot location or for localization given an a priori map. However, map building has been successfully applied using occupancy grids. The scalability and extensibility of the evidence grid representation is limited to a predefined fixed resolution and area. Moreover, the increase in resolution will result at the expense of additional storage space and increase in computation. Evidence grids represent both occupied space and empty space of an environment, which is useful for performing tasks such as path-planning and obstacle avoidance. Although, this representation can result in computationally expensive algorithms due to the large search spaces typically associated with large-scale environments and high grid resolutions. Some of the issues inherent to the traditional occupancy grid representation can be overcome by the use of quad-trees. As usual evidence grids quad-trees also represent an environment by an array of rectangular cells. However, with this representation scheme scalability is obtained by a recursive subdivision of existing cells. As the name suggests, each rectangular cell can be subdivided into four elements and a quad-tree structure is used for storage and reasoning. In addition, cells are allowed to have a partial state of occupancy. By expanding in depth the child nodes

of this tree structure, a local increase in resolution and finer inference in the state of occupancy of specific regions of the environment can be achieved. Finally, with quad-trees, changing the resolution of the map at a local scale results with a relatively small increase in storage and computation resources. The idea behind evidence grids and quad-trees has also been extended to the three dimensional space. In a 3D occupancy grid representation, the environment is subdivided into a regular 3D array of cells, or otherwise referred to as voxels. Likewise, oc-trees allow the recursive subdivision of voxels, into eight smaller elements.

2.1.4 Comparison between Map Representations

Either metric, geometric or evidence grid based, or topological representations have significant drawbacks. In principle, topological maps should scale better than metric maps to large-scale environments, because a coarse-grained, graph-structured representation is much more compact than a dense array, and more directly suited to problem-solving algorithms. However, purely topological maps have difficulties to distinguish adequately among different places, and have not been applied successfully to large environments in practice. Recent progress in metric mapping has made it possible to build useful and accurate metric maps of reasonably large-scale environments, but memory and time complexity cause problems.

While metric representations are very useful in robotic applications, human beings do not operate in such discrete ways. Rather, an individual tends to remember localities with reference to other localities, even though certain relations may well have distances associated with them. For instance, a human will likely not know the exact bearing and distance of Frankfurt with respect to Freiburg, but know that he needs to pass Karlsruhe on the way, and that it takes roughly 50 minutes to get from Freiburg to Karlsruhe. Furthermore, topological representations reveal relations between localities and rely on the agent to employ separate navigational techniques.

While probabilistic techniques have been considered extensively in the context of metric maps, no general purpose of probabilistic methods exist for topological maps. In this thesis we will not use topological map representations, as we strongly use probabilistic methods as well as the metric information. Most times we will use evidence grids for the map representation. Unfortunately the memory requirement prevents us to use the same representation in the three dimensional case. Therefore in those cases we will switch to a metric representation with geometric primitives. The number of possible measurements is also very restricted using geometric primitives, but the necessary memory requirement is significantly smaller compared to grid maps.

2.2 Learning Grid Maps

As we mentioned before many methods have been employed for updating the state of occupancy for each cell, such as Bayesian, Dempster-Shafer, and Fuzzy Logic. Here we will describe two ways for updating grid cells: *Bayesian Update* and the *Counting Model*. The evidence grid provides elegant ways to incorporate different models of sensor uncertainty. We do not need any feature extraction and therefore we can easily apply the update technique to different kinds of sensors.

2.2.1 Occupancy Probability Grid Maps with Bayesian Update

In this section we will discuss how a mobile robot can learn an occupancy grid map from the sensor data using Bayesian Updating. Given the positions x_t of the vehicle at each point in time t is known, suppose $x_{1:t} = x_1, \dots, x_t$ are the positions of the robot at the individual steps in time, and $z_{1:t} = z_1, \dots, z_t$ are the perceptions of the environment. Occupancy probability grids determine for each cell m_l of the grid the probability that this cell is occupied by an obstacle. Thus, occupancy probability grids seek to find the map m that maximizes $P(m | x_{1:t}, z_{1:t})$. If we apply Bayes rule using $x_{1:t}$ and $z_{1:t-1}$ as background knowledge, we obtain

$$P(m | x_{1:t}, z_{1:t}) = \frac{P(z_t | m, x_{1:t}, z_{1:t-1}) \cdot P(m | x_{1:t}, z_{1:t-1})}{P(z_t | x_{1:t}, z_{1:t-1})}.$$

If we assume that z_t is independent from $x_{1:t-1}$ and $z_{1:t-1}$ given we know m , the right side of this equation can be simplified to

$$P(m | x_{1:t}, z_{1:t}) = \frac{P(z_t | m, x_t) \cdot P(m | x_{1:t}, z_{1:t-1})}{P(z_t | x_{1:t}, z_{1:t-1})}. \quad (2.1)$$

Applying Bayes rule we determine

$$P(z_t | m, x_t) = \frac{P(m | z_t, x_t) \cdot p(z_t | x_t)}{P(m | x_t)}. \quad (2.2)$$

If we insert Equation (2.2) into Equation (2.1) assuming that x_t does not carry any information about m if there is no observation z_t , we obtain

$$P(m | x_{1:t}, z_{1:t}) = \frac{P(m | z_t, x_t) \cdot p(z_t | x_t) \cdot P(m | x_{1:t-1}, z_{1:t-1})}{P(m) \cdot P(z_t | x_{1:t}, z_{1:t-1})} \quad (2.3)$$

If we exploit the fact that each cell m_l is a binary variable, we derive the following equation in an analogous way.

$$\begin{aligned} & P(\neg m | x_{1:t}, z_{1:t}) \\ &= \frac{P(\neg m | z_t, x_t) \cdot p(z_t | x_t) \cdot P(\neg m | x_{1:t-1}, z_{1:t-1})}{p(\neg m) \cdot P(z_t | x_{1:t}, z_{1:t-1})} \end{aligned} \quad (2.4)$$

By dividing Equation (2.3) by Equation (2.4), we obtain

$$\frac{P(m \mid x_{1:t}, z_{1:t})}{P(\neg m \mid x_{1:t}, z_{1:t})} = \frac{P(m \mid z_t, x_t) \cdot p(\neg m) \cdot P(m \mid x_{1:t-1}, z_{1:t-1})}{P(\neg m \mid z_t, x_t) \cdot P(m) \cdot P(\neg m \mid x_{1:t-1}, z_{1:t-1})}.$$

Finally, we use the fact that $P(\neg A) = 1 - P(A)$ which yields to

$$\begin{aligned} & \frac{P(m \mid x_{1:t}, z_{1:t})}{1 - P(m \mid x_{1:t}, z_{1:t})} \\ &= \frac{P(m \mid z_t, x_t)}{1 - P(m \mid z_t, x_t)} \cdot \frac{1 - P(m)}{P(m)} \cdot \frac{P(m \mid x_{1:t-1}, z_{1:t-1})}{1 - P(m \mid x_{1:t-1}, z_{1:t-1})}. \end{aligned} \quad (2.5)$$

If we define

$$\text{Odds}(x) = \frac{P(x)}{1 - P(x)} \quad (2.6)$$

Equation (2.5) turns into

$$\begin{aligned} & \text{Odds}(m \mid x_{1:t}, z_{1:t}) \\ &= \text{Odds}(m \mid z_t, x_t) \cdot \text{Odds}(m)^{-1} \cdot \text{Odds}(m \mid x_{1:t-1}, z_{1:t-1}). \end{aligned} \quad (2.7)$$

The corresponding log Odds representation of Equation (2.7) is

$$\begin{aligned} & \log \text{Odds}(m \mid x_{1:t}, z_{1:t}) = \\ & \log \text{Odds}(m \mid z_t, x_t) - \log \text{Odds}(m) + \log \text{Odds}(m \mid x_{1:t-1}, z_{1:t-1}). \end{aligned} \quad (2.8)$$

To incorporate a new scan into a given map we multiply its Odds-ratio with the Odds-ratio of a local map constructed from the most recent scan and divide it by the Odds-ratio of the prior. Often it is assumed that the prior probability of m is 0.5. In this case the prior can be canceled so that Equation (2.8) simplifies to

$$\begin{aligned} & \log \text{Odds}(m \mid x_{1:t}, z_{1:t}) \\ &= \log \text{Odds}(m \mid z_t, x_t) + \log \text{Odds}(m \mid x_{1:t-1}, z_{1:t-1}). \end{aligned} \quad (2.9)$$

To recover the occupancy probability from the Odds representation given in Equation (2.7) we use the following law which can easily be derived from Equation (2.6):

$$P(x) = \frac{\text{Odds}(x)}{1 + \text{Odds}(x)} \quad (2.10)$$

This leads to:

$$\begin{aligned} & P(m \mid x_{1:t}, z_{1:t}) \\ &= \left[1 + \frac{(1 - P(m \mid x_t, z_t))}{P(m \mid x_t, z_t)} \cdot \frac{P(m)}{(1 - P(m))} \cdot \frac{1 - P(m \mid x_{1:t-1}, z_{1:t-1})}{P(m \mid x_{1:t-1}, z_{1:t-1})} \right]^{-1} \end{aligned} \quad (2.11)$$

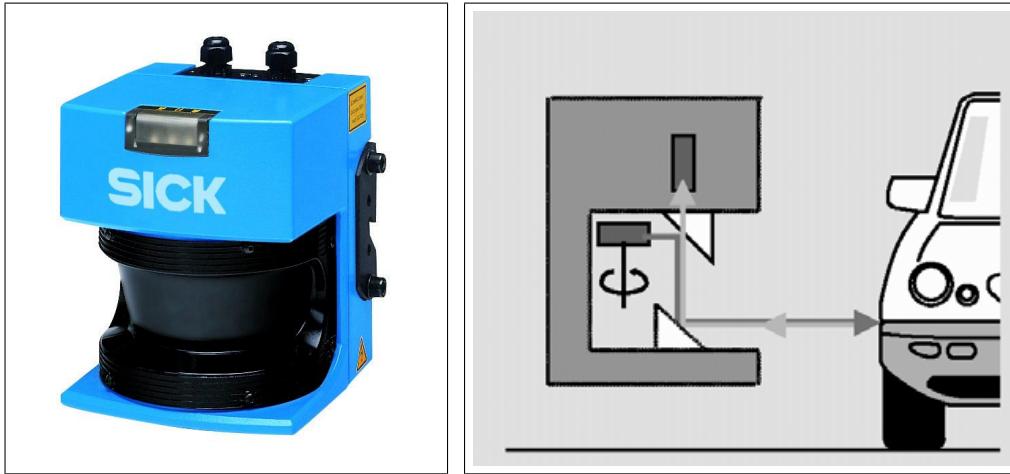


Figure 2.4: The SICK LMS is a non-contact laser measurement system that scan the surrounding two-dimensionally (laser radar). They require neither reflectors nor position markers.

This equation tells us how to update our belief about the occupancy probability of a grid map given sensory input. All we have to do to incorporate a new measurement z_t taken at location x_t is to multiply the odds ratio of the current belief about m with the odds ratio of the map constructed from the most recent measurement and divide the result by the prior probability of m . In practice one often assumes that the initial Belief $P(m)$ about the map is 0.5 so that the prior can be reduced from the equation.

2.2.2 Computation of the Occupancy Probability

It remains to describe how we actually compute $P(m \mid x_t, z_t)$. According to Equation (2.11) a measurement z_t has no influence on a cell m_l if $P(m_l \mid x_t, z_t) = P(m_l)$. Therefore, cells in areas in which a measurement does not change the belief about the state of the world carry the value $P(m_l)$. Often $P(m_l) = 0.5$ is assumed to be the prior, but it can be regarded as a special instance of this situation.

Our approach assumes that the occupancy probability of a cell m_l of the grid can be computed independently for all sensor measurements. Although this is a strong assumption, the resulting maps still are of satisfactory quality.

In our case we use mainly laser-range finders (see Figure 2.4) as the robotic sensor. This sensor is very common in robotics and currently the state-of-the-art sensor for distance measurements. The signal of a laser-range finder is emitted in a beam (see right image of Figure 2.4). The sensor uses a rotating mirror

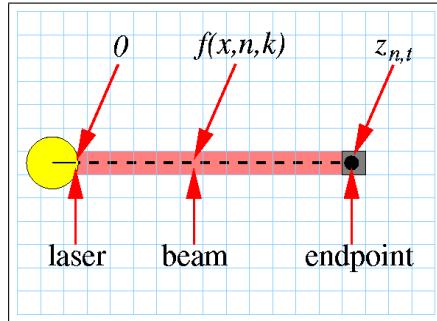


Figure 2.5: Beam covering $z_{t,n}$ cells of a map. The occupancy of the cell where the beams ends should be increased. On the other side the occupancy of the cells between the robot and the end-point, which are here marked with the color red, should be decreased.

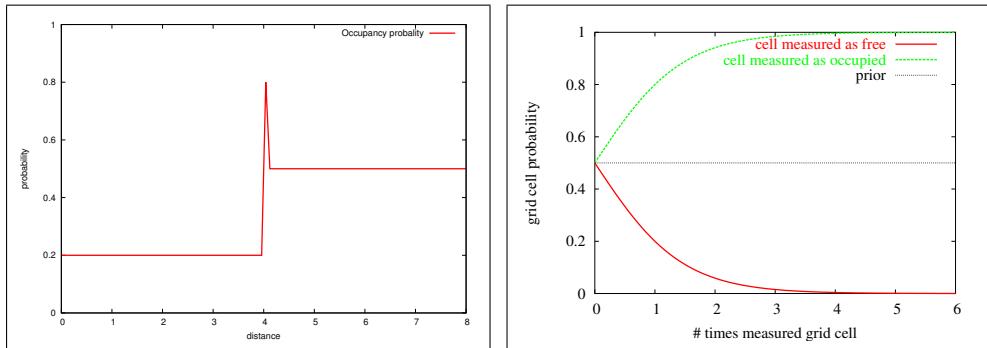


Figure 2.6: The left image shows the occupancy probability of a beam measuring a distance of 4m. Between the sensor and the measured distance it is more likely to be free, at the measured distance we expect the grid cell to be occupied. Please note that because of the small error of the sensor this function has only a narrow peak. The right image shows the probability of a occupancy grid cell seen several times as occupied (green) or unoccupied (red). We use the odds-model described in Equation 2.11 to compute the probabilities. The prior of the map is 0.5.

to combine several distance measurements to a two dimensional scan. At each time t we receive a complete scan z_t , which is a combination of several distance measurements $z_{t,n}$. We assume that this distance measurements are independent and therefore consider the beams individual.

The left side of Figure 2.6 shows the function we use to compute the occupancy probability measuring a specific distance (in this case 4m). We can divide the function in three parts. Let f be a function that returns for each pose x_t of the robot, each beam number n , and each $k \leq z_{t,n}$ the index $f(x_t, n, k)$ of k -th

field covered by that beam in the map (see Figure 2.5). Now we can describe the function as follow:

$$P(m | x_t, z_t) = \begin{cases} 0.2 & : |f(x_t, n, k), x_t| < z_{t,n} \\ 0.8 & : |f(x_t, n, k), x_t| = z_{t,n} \\ 0.5 & : \text{else} \end{cases} \quad (2.12)$$

We use the Equation 2.11 to update the cells. Figure 2.6 (right) shows the function of the probability, that we see the grid cell several time as free or occupied. The left image of Figure 2.7 shows the case that a cell is seen several times as occupied and unoccupied. We can describe this function using following preconditions: first, the prior of the map is 0.5 and second, P_{free} is the probability for cells with $|f(x_t, n, k), x_t| < z_{t,n}$ and P_{occ} for the cell with $|f(x_t, n, k), x_t| = z_{t,n}$ (see Equation 2.12). Finally we count the number of times the cells is updated by P_{free} or P_{occ} and describe this number with c_{free} or respectively c_{occ} . Applying Equation 2.11 we compute the probability directly with:

$$f(c_{occ}, c_{free}) = \frac{\exp(c_{occ}P_{occ} + c_{free}P_{free})}{1 + \exp(c_{occ}P_{occ} + c_{free}P_{free})}. \quad (2.13)$$

It is easy to see that the steepness of the functions is related to the values of P_{occ} and P_{free} . Figure 2.8 shows, as a practical example, how fast the cell approximate to 0 or 1. This Figure shows the creation of a typical map. In the beginning some parts are gray, but after the integration of a few measurements most parts are black or white. In this example we used $P_{occ} = 0.8$ and $P_{free} = 0.2$. To visualize the differences in the resulting map, we created several maps with this data set using for different values for P_{occ} and P_{free} . We always used $P_{free} = 1 - P_{occ}$ as a simplification. Figure 2.9 (a-g) shows the resulting maps. In the upper left part of each map we can see the used value.

2.2.3 Learning Grid Maps using the Counting Model

In Chapter 4 we will discuss a different approach to update grid maps. In the last section we described a technique to learn occupancy grids using Bayesian Updating. The goal is to compute the probability that the grid cell is occupied by an obstacle. In contrast to this we will discuss a way to compute the probability that a beam will be reflected. Later we will go into the details of the derivation for this technique. At this point we briefly introduce the *counting model*, described in detail in Section 4.2, and compare this model with the odds model to learn occupancy grid maps.

First we define the likelihood of a measurement given the current map m of the environment, the pose x of the robot, and the information about whether $z_{t,n}$

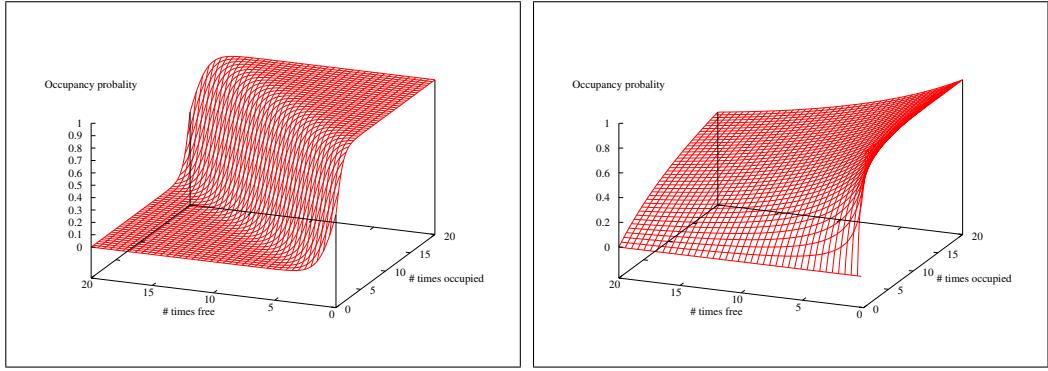


Figure 2.7: The images show the occupancy probability for a cell seen several times as occupied *and* free. The left side shows the result using the odds model with the probability function shown in Figure 2.6 (left). The right side shows the resulting probabilities using the counting model (see Equation 2.15)

is reflected by a maximum range reading. We introduce indicator variables $\zeta_{t,n}$ which are 1 if and only if $z_{t,n}$ is a maximum range reading and 0 otherwise. The likelihood of a measurement $z_{t,n}$ given the map m can thus be computed as:

$$p(z_{t,n} \mid x_t, m) = \prod_{k=0}^{z_{t,n}-1} (1 - m_{f(x_t, n, k)})^{\zeta_{t,n}} \cdot \left[m_{f(x_t, n, z_{t,n})} \cdot \prod_{k=0}^{z_{t,n}-1} (1 - m_{f(x_t, n, k)}) \right]^{(1-\zeta_{t,n})} \quad (2.14)$$

We can separate Equation 2.14 in two parts. The first part is not zero, if we have a maximum range reading. In this case we expect that all cells up to the last grid cell should be free. In the other case, which is represented by the second part, we additional expect that the last grid cell is occupied.

$$p(z_{t,n} \mid x_t, m) = \begin{cases} \prod_{k=0}^{z_{t,n}-1} (1 - m_{f(x_t, n, k)}) & : \text{max. range} \\ m_{f(x_t, n, z_{t,n})} \cdot \prod_{k=0}^{z_{t,n}-1} (1 - m_{f(x_t, n, k)}) & : \text{else} \end{cases}$$

In Chapter 4 we will show that, given this sensor model, the closed-form solution for the most likely map m for given poses x corresponds to the naive counting technique in which one counts for each cell how often a beam has ended in that cell and how often a beam has covered it without ending in it. By straightforward mathematical transformations we obtain

$$f(c_{occ}, c_{free}) = \frac{c_{occ}}{c_{free} + c_{occ}}. \quad (2.15)$$

Figure 2.7 shows the difference between the function of the odds model (Equation 2.13) and the counting model (Equation 2.15). The left image shows the probabilities seen the cell several times as free (x-axis) as well as occupied (y-axis)

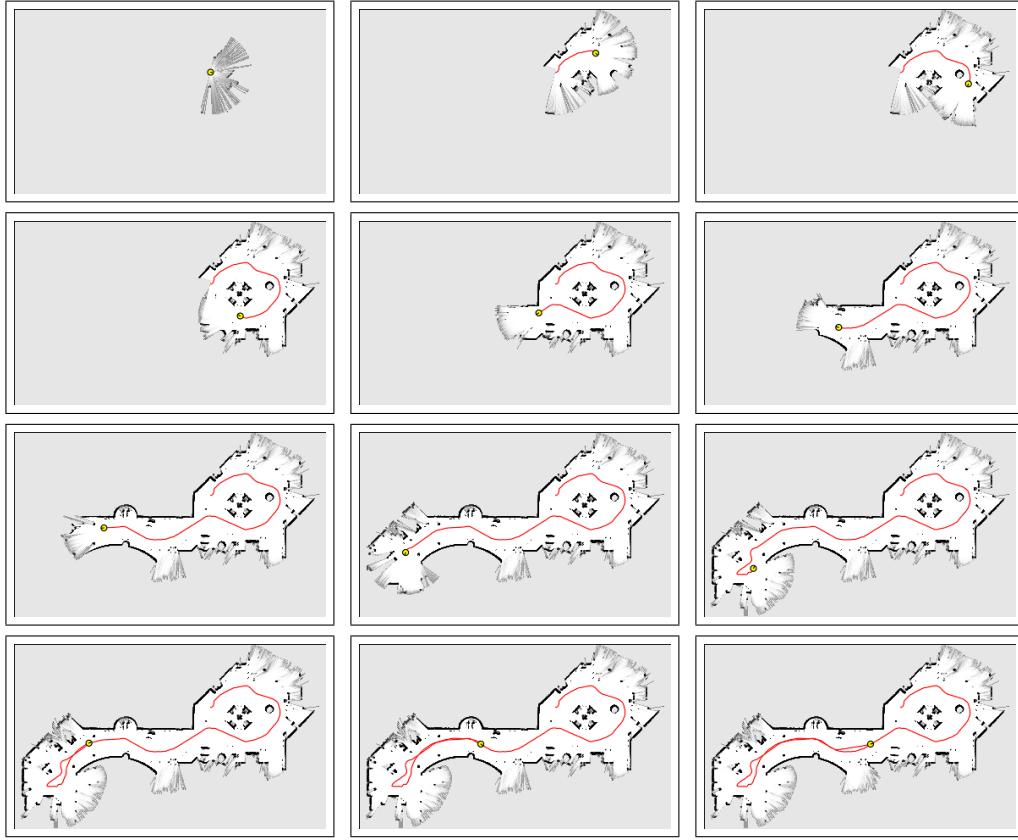


Figure 2.8: Incremental mapping of an indoor environment. The upper left image show the initial map and the lower map show the resulting map. We used the odds-model for creating the map with the measurement function shown in Figure 2.6 (left).

using the odds model. The right image shows the probabilities computed with the counting model. Obviously the function on the left maps most parts of the domain to 0 or 1. As we mentioned before the steepness of this function depends on the used values for P_{occ} and P_{free} . Figure 2.9 (a-g) shows maps of the same data set with different values for P_{occ} and P_{free} . In contrast to this Figure 2.7 (right) shows that the function for the counting models covers smoothly all values between 0 and 1. Figure 2.9 (h) shows the resulting map using the counting model. Although the map of Figure 2.9 (g) seems to be more accurate, some details got lost. For example glass (one side of the room in the upper right part of the map is out of glass) is only visible in map (h), but in none of the other maps (a-g).

The following example also show the difference between both approaches. Let us assume that we will sense the environment in blocks of 100 measurements. 70% of these measurements will tell us that a specific grid cell is seen as occupied or being able to reflect the beam. The remaining 30% of the measurements have

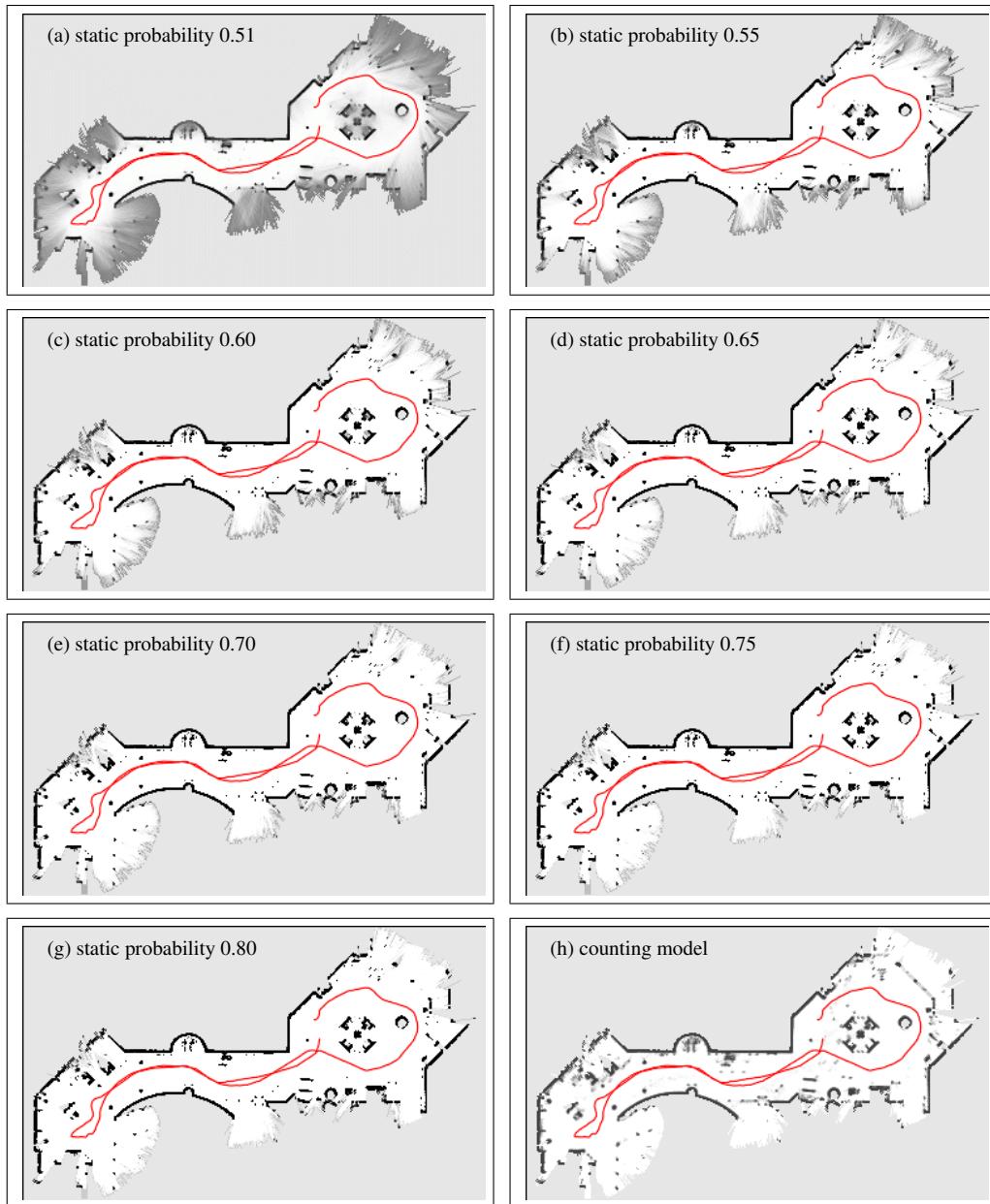


Figure 2.9: This figure shows occupancy probability grid maps created with the odds-model and the counting-model. The images (a)-(g) show the maps using the odds-model for different measurement functions (see left image of Figure 2.6). In comparison image (h) shows the resulting map from the same data set using the counting model. The red path shows the trajectory of the robot collecting the data.

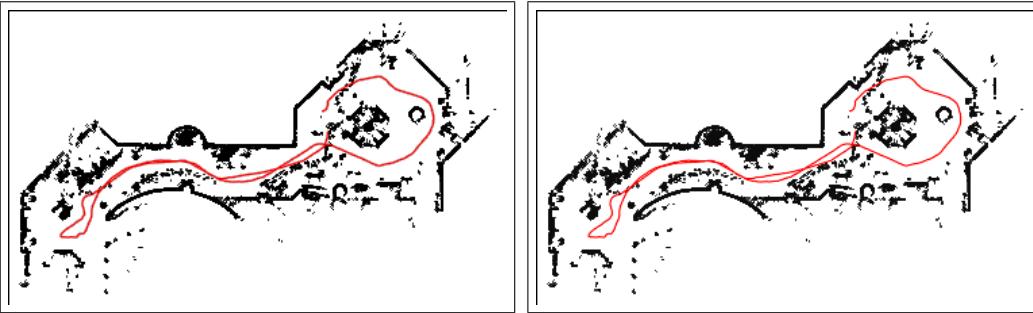


Figure 2.10: Maps created with end-points information. On the left side the simplified odds model was used, the right image shows the result using the counting model. The maps show no visible difference.

seen the cell as free space. While the probability of the cell using the odds-model would approximate 1, in the case of the counting-model the probability of the cell would always be 0.7. Therefore the difference between the models is more than the speed of convergence, they also can differ in the approximated value.

2.2.4 Fast Approximation using End-Points

Independently of the used technique for updating the occupancy grids, they have one disadvantage in common. In all cases the complete beam has to be considered for the computation. This means that in high resolution grid maps the number of cells which has to be updated would be large. Additional data processing, for example convolutions to approximate the uncertainty in the position, has then a high computational cost. One way to speed up the process, is to consider only the end-points of the beams. Obviously, in that case we do not use the information of a beam about free space. This is a major drawback. Once a spurious measurement is included, for example caused by a person walking by, we do not have the possibility to correct this by integrating multiple measurements. Therefore these kinds of measurements can have strong influence and we have to be very carefully with spurious measurements. On the other side this approximation makes it possible to handle high resolution grid maps. Using end-points only drastically reduces the computation of the occupancy grid cells. Setting c_{free} always to zero means for the odds-model

$$f(c_{occ}) = \frac{\exp(c_{occ}P_{occ})}{1 + \exp(c_{occ}P_{occ})} \quad (2.16)$$

and for the counting model

$$f(c_{occ}) = \frac{c_{occ}}{c_{occ}} = 1. \quad (2.17)$$

Algorithm 1 Technique for efficient local map building

Input: measurements z_i with poses \hat{x}_i , maximal number of integrated scans α , maximal length of the used history β , and minimal distance between integrated scans γ .

Output: local map build with at most α scans and a minimal distance γ between used scans.

```

 $idx = t - 1, count = 1$ 
integrate measurement  $z_{idx}$  with pose  $\hat{x}_{idx}$ 
for  $i = t - 2$  to  $i = 0$  do
    if ( $count \geq \alpha \vee t - i \geq \beta$ ) then
        abort local map building
    else
        if ( $\|\hat{x}_{idx}, \hat{x}_i\| \geq \gamma$ ) then
             $idx = i, count++$ 
            integrate measurement  $z_{idx}$  with pose  $\hat{x}_{idx}$ 
        end if
    end if
end for
```

Obviously, in the case of the counting model we just mark the cells where the beam ends. The difference of the result using Equation 2.16 or Equation 2.17 is often marginal. Usually many beams end up in the same cell (this depends on the used resolution) and therefore the values of the grid cells are often close to 1. Figure 2.10 shows two maps created in such ways. While for the left map we used the odds model, we cannot see any difference to the simplified version using the probability one for every marked cell.

The usage of such a simplified model drastically speeds up the process. Another computational limitation in mapping is the required space. High resolution grid maps of bigger areas needs a large amount of space. Due to the fact that in incremental mapping we do not need the whole map for correcting the position of a single scan, we use only the area close to current position. In contrast to storing the whole map we can alternatively generate local views on the fly. This is possible if the process of generating these views can be executed fast. This is the case if we consider only the end-points. Since that such maps will be typically sparse, we can accelerate the computation by using efficient data structures like quad-trees. We then have the advantage, that the memory requirement of such a system does not grow over time and depends only on the resolution of the local map. As we have to integrate all measurements from the previous scans, the number of neces-

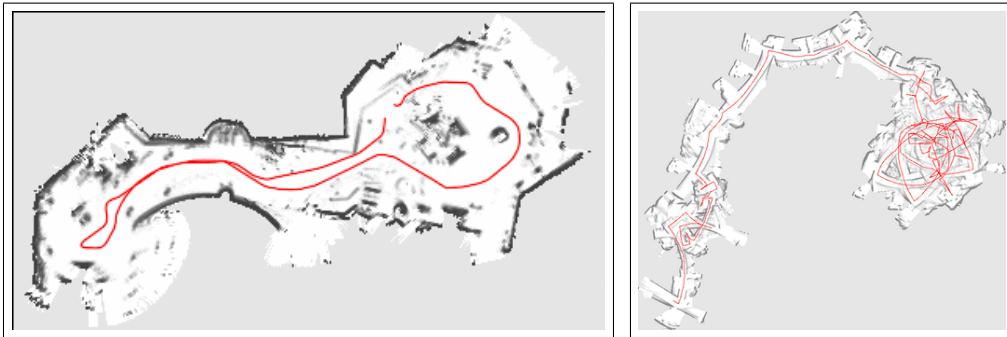


Figure 2.11: Mapping of the Deutsches Museum Bonn with raw odometry data of a B21 robot (left) and the Intel Research Lab with raw odometry data of a Pioneer robot (right). In both cases we used the counting model to update the grid cells.

sary computations increases with the length of the history. In our system we limit the number of possible scans by considering only a time slot instead of the whole history.

$$x_{1:t} \approx x_{t-\Delta t}, \dots, \hat{x}_{t-1} \quad z_{1:t} \approx z_{t-\Delta t}, \dots, z_{t-1} \quad (2.18)$$

Algorithm 1 gives us a detailed description how we generate such local maps.

2.3 Incremental Scan Matching

In the previous section we described a technique to generate occupancy grid maps with known positions. In this section we will show how to estimate the position using incremental scan matching. Most of the robots use wheel encoders to measure their movements. By integrating the movements we can compute the current of the position of the robot. Unfortunately the measurements are most of the time not precise. Errors are caused by several factors. First, the measurement device has only limited accuracy, second the robot can skid on the ground. Especially in case of skid-steer platforms this leads to large measurement errors in rotation. Figure 2.11 shows two maps generated with the odometry data of two different robots. The left map was created by iRobot B21 robot with a synchro drive. This robot measures additional to the wheel rotation and their orientation. The right map was generated from the data of a ActivMedia Pioneer robot. This robot has a skid-steering and is more sensitive to rotational errors. In both cases the map is not correct and unusable for navigation tasks like path planning. We therefore need a technique which helps us to get a better position estimation. One of the most popular techniques to correct position errors is incremental scan matching.

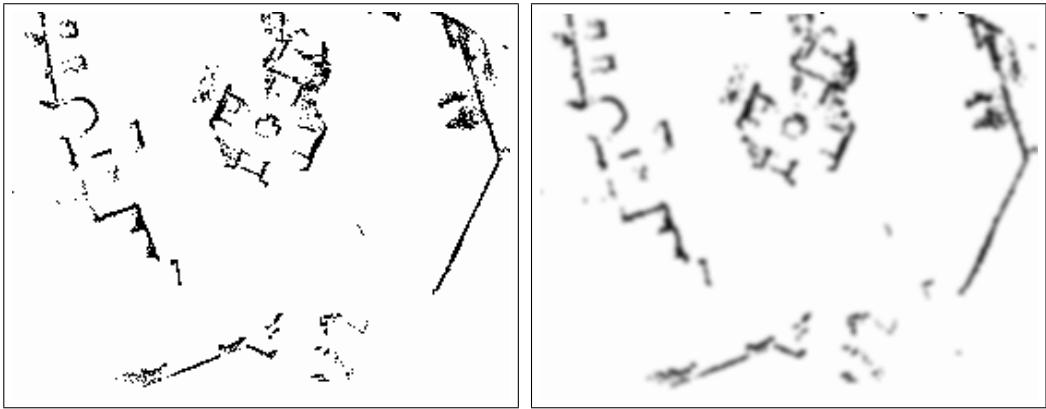


Figure 2.12: Local map built with 20 scans (left) and after integration of small Gaussian errors in the robot pose when computing the local map (right).

In this technique consecutive scans are compared and the positions of the scan is changed to minimize the alignment error to the other corresponding scans. The advantage of this technique is obvious. First, the computation can be carried out very fast and second, the algorithm is incremental and therefore able to run online.

2.3.1 Probabilistic Matching for Scan Registration

Mathematically, we calculate a sequence of poses $\hat{x}_1, \hat{x}_2, \dots$ and corresponding maps by maximizing the marginal likelihood of the t -th pose and map relative to the $(t-1)$ -th pose and map:

$$\hat{x}_t = \underset{x_t}{\operatorname{argmax}} \{ p(z_t | x_t, \hat{m}(\hat{x}_{1:t-1}, z_{1:t-1})) \cdot p(x_t | u_{t-1}, \hat{x}_{t-1}) \} \quad (2.19)$$

In this equation the term $p(z_t | x_t, \hat{m}(\hat{x}_{1:t-1}, z_{1:t-1}))$ is the probability of the most recent measurement z_t given the pose x_t and the map $\hat{m}(\hat{x}_{1:t-1}, z_{1:t-1})$ constructed so far. The term $p(x_t | u_{t-1}, \hat{x}_{t-1})$ represents the probability that the robot is at location x_t given the robot previously was at position \hat{x}_{t-1} and has carried out (or measured) the motion u_{t-1} . The resulting pose \hat{x}_t is then used to generate a new map \hat{m} via the standard incremental map updating function presented in the last section:

$$\hat{m}(\hat{x}_{1:t}, z_{1:t}) = \underset{m}{\operatorname{argmax}} p(m | \hat{x}_{1:t}, z_{1:t}) \quad (2.20)$$

In our implementation we use the fast approximation (see Section 2.2.4) to build the maps with end-points and a local views. In that case one disadvantage of the approach lies in the fact, that the complexity of a single maximization step is in

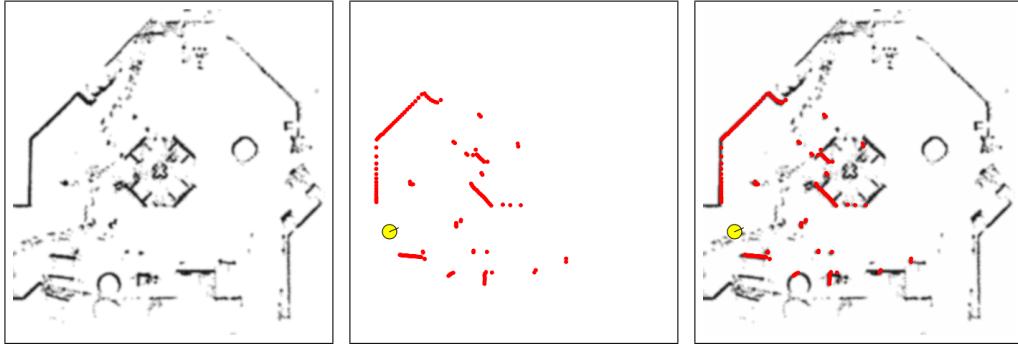


Figure 2.13: Two-dimensional scan alignment. Map created out of the Δt most recent scans (left image), measurement z_t obtained at time t (center image) and resulting alignment (right image).

$O(t)$, since every measurement is compared with all previous measurements. In our system, we therefore use a map

$$\hat{m}(\hat{x}_{t-\Delta t:t-1}, z_{t-\Delta t:t-1}) = \hat{m}(\hat{x}_{t-\Delta t}, \dots, \hat{x}_{t-1}, z_{t-\Delta t}, \dots, z_{t-1}) \quad (2.21)$$

that is constructed based of the Δt most recent measurements only (see also Equation 2.18). This is motivated by two observations. First, proximity sensors have a limited range only so that the system generally cannot cover the whole environment with a single scan. Additionally, objects in the environment lead to occlusions so that many aspects of a given area are invisible from other positions. Therefore, measurements obtained at distant places often provide no information to maximize (2.19).

The overall approach can be summarized as follows: At any point $t - 1$ in time the robot is given an estimate of its pose \hat{x}_{t-1} and a map $\hat{m}(\hat{x}_{t-\Delta t:t-1}, z_{t-\Delta t:t-1})$. After the robot has moved further on and after it has taken a new measurement z_t , the robot determines the most likely new pose \hat{x}_t . It does this by trading off the consistency of the measurement with the map (first term on the right-hand side in (2.19)) and the consistency of the new pose with the control action and the previous pose (second term on the right-hand side in (2.19)). The map is then extended by the new measurement z_t , using the pose \hat{x}_t as the pose at which this measurement was taken.

2.3.2 Two-dimensional Scan Alignment with Occupancy Grid Maps

It remains to describe how we actually maximize Equation (2.19). To align a scan relative to the Δt most recent scans, we first construct a local grid map

$\hat{m}(\hat{x}_{t-\Delta t:t-1}, z_{t-\Delta t:t-1})$ out of the Δt most recent scans. Additionally we integrate over small Gaussian errors in the robot pose when computing the maps. This avoids that many cells remain unknown especially if t is small. Additionally, it increases the smoothness of the map and corresponding likelihood function to be optimized and thus facilitates the range registration. As an example, consider the Figure 2.12 which shows a typical map constructed out of 20 scans on the left side. The darker a location, the higher the likelihood that the corresponding place in the environment is covered by an obstacle. On the right side of Figure 2.12 one may notice that the map appears slightly blurred according to the integration over small pose errors (see also left image in Figure 2.13). To maximize the likelihood of a scan with respect to this map, we apply a hill climbing strategy. A typical scan is shown in the center of Figure 2.13. The optimal alignment of this scan with respect to the map is shown in the right image of Figure 2.13.

2.4 Scan Registration in 3D

The topic of learning 3D models of buildings (exterior and interior) and man-made objects has received considerable attention over the past few years. 3D models are useful for a range of applications. For example, architects and building managers may use 3D models for design and utility studies using virtual reality (VR) technology. Emergency crews, such as fire fighters, could utilize 3D models for planning as to how to best operate at a hazardous site. 3D models are also useful for robots operating in urban environments. In all of these application domains, there is a need for methods that can generate 3D models at low cost, and with minimum human intervention.

In Section 2.3.1 of this chapter we described a technique to maximize Equation 2.19 using occupancy grid maps. Although the proposed implementation is two-dimensional, we can apply the same technique in three dimensions. Unfortunately there are practical limitations that prevent us from doing this efficiently. Mainly the memory requirements and the time needed for updating or computing the map prevent an efficient implementation. Obviously a (high resolution) 3D occupancy grid map needs more memory than a typical computer can provide. Outdoor scenarios often need three-dimensional maps, but especially in such cases the map would be very large. In the previous section we used several local maps instead of one large global map. This decreases the necessary amount of memory, but on the other side introduces higher computational costs.

Researchers in several robotic labs extended the capability of the 2D laser range finder to acquire three-dimensional scans. The laser is typically mounted on a tilt unit (see Figure 2.14) and can therefore measure three-dimensional structures by changing the tilt. The range of such a system is 60m. Even if we use several



Figure 2.14: Robots which are equipped with a SICK LMS laser and an AMTEC wrist unit. The 3D data are acquired by changing the tilt of the AMTEC module.

local maps we are currently not able to handle such high resolution occupancy grid maps.

Therefore we have to find an alternative map representation to apply our scan matching technique. In Section 2.1.1 we described the geometric map representation in contrast to occupancy grid maps. Geometric maps consists of geometric primitives, often points or triangles. Before we show how we can apply the maximization of Equation 2.19 using a geometric map representation, we will describe the most popular method for three dimensional range data registration: the ICP algorithm.

2.4.1 The Iterative Closest Point (ICP) Algorithm

The Iterative Closest Point (ICP) Algorithm [Besl and McKay, 1992, Greenspan and Godin, 2001] is a popular method of 3D range data processing. Given two sets of partially overlapping range data and an initial estimate of their relative positions, the ICP approach computes a registration of these two sets in an iterative fashion. In each round, the ICP algorithm determines for each point in the first set the closest point in the second set. It then tries to find a relative position of the

two scans such that the mean squared error (MSE) given by the sum of the squared distances between the corresponding points is minimized. The process is continued until either the MSE falls below a given threshold or no further improvement can be achieved.

Given two independently acquired sets of 3D points, X (model set, $X = \{x_i\}$) and P (data set, $P = \{p_i\}$) where $N_x = N_p$ and where each point p_i corresponds to the point x_i with the same index. We want to find the transformation consisting of a rotation R and a translation t which minimizes the following cost function:

$$E(R, t) = \frac{1}{N_p} \sum_{i=1}^{N_p} \|x_i - Rp_i - t\|^2 \quad (2.22)$$

The center of mass μ_p of the measured point set P and the center of mass μ_x of the model set X are given by

$$\mu_p = \frac{1}{N_p} \sum_{i=1}^{N_p} p_i \quad \text{and} \quad \mu_x = \frac{1}{N_x} \sum_{i=1}^{N_x} x_i$$

To determine the transformation, the mean values of the paired points μ_x and μ_p are subtracted from all points in X and P , respectively, resulting in the sets $X' = \{x_i - \mu_x\} = \{x'_i\}$ and $P' = \{p_i - \mu_p\} = \{p'_i\}$.

$$\begin{aligned} E(R, t) &= \frac{1}{N_p} \sum_{i=1}^{N_p} \|x_i - Rp_i - t\|^2 \\ &= \frac{1}{N_p} \sum_{i=1}^{N_p} \|(x'_i + \mu_x) - R(p'_i + \mu_p) - t\|^2 \\ &= \frac{1}{N_p} \sum_{i=1}^{N_p} \|x'_i - Rp'_i - \underbrace{(t - \mu_x - R\mu_p)}_{=:t'}\|^2 \\ &= \frac{1}{N_p} \left(\sum_{i=1}^{N_p} \|x'_i - Rp'_i\|^2 - 2t' \sum_{i=1}^{N_p} (x'_i - Rp'_i) + \sum_{i=1}^{N_p} \|t'\|^2 \right) \quad (2.23) \end{aligned}$$

To minimize the sum we have to minimize all of the terms. Because of $\sum_{i=1}^{N_p} x'_i = 0$ and $\sum_{i=1}^{N_p} p'_i = 0$ we know that the second term is zero. The third term has a minimum for $t' = 0 \Leftrightarrow t = \mu_x + R\mu_p$. Therefore we minimize the following new function

$$E(R, t) = \sum_{i=1}^{N_p} \|x'_i - Rp'_i\|^2. \quad (2.24)$$

As rotation does not change the length of a vector, we can conclude that $\|Rp'_i\|^2 = \|p'_i\|^2$. Given this equality we can extend the Equation 2.24 to

$$E(R, t) = \sum_{i=1}^{N_p} \|x'_i\|^2 - 2 \sum_{i=1}^{N_p} (x'_i - Rp'_i) + \sum_{i=1}^{N_p} \|p'_i\|^2, \quad (2.25)$$

in which only the second term depends on rotation. To minimize Equation 2.25 we have to maximize:

$$\sum_{i=1}^{N_p} (x'_i - Rp'_i) \quad (2.26)$$

At this point we use quaternions to represent the rotation. A quaternion can be thought of as a vector with four components, as the composite of a scalar and an ordinary vector, or as a complex number with three different imaginary parts. Quaternions will be denoted here by using symbols with points above them. Thus the complex number rotation, we have

$$\dot{q} = q_0 + iq_x + jq_y + kq_z,$$

a quaternion with real part q_o and three imaginary parts, q_x , q_y , and q_z . If $q_0^2 + q_x^2 + q_y^2 + q_z^2 = 1$, $q_0 \geq 0$ we call it a unit quaternion. Multiplications of quaternions can be defined in terms of the products of their components. Suppose that we let

$$\begin{aligned} i^2 &= -1, & j^2 &= -1, & k^2 &= -1 \\ ij &= k, & jk &= i, & ki &= j \end{aligned}$$

and

$$ji = -k, \quad kj = -i \quad ik = -j$$

Then, if

$$\dot{r} = r_0 + ir_x + jr_y + kr_z,$$

we get

$$\begin{aligned} \dot{r}\dot{q} &= (r_0q_0 - r_xq_x - r_yq_y - r_zq_z) \\ &\quad + i(r_0q_x + r_xq_0 + r_yq_z - r_zq_y) \\ &\quad + j(r_0q_y - r_xq_z + r_yq_0 + r_zq_x) \\ &\quad + k(r_0q_z + r_xq_y - r_yq_x + r_zq_0) \end{aligned}$$

The product $\dot{q}\dot{r}$ has a similar form, but six of the signs are changed, as can easily be verified. Thus $\dot{r}\dot{q} \neq \dot{q}\dot{r}$, in general. The product of two quaternions can also be

expressed in terms of product of a 4×4 matrix and a vector with four components. One may choose to expand either the first or the second quaternion in a product in a 4×4 matrix as follows:

$$\dot{r}\dot{q} = \underbrace{\begin{bmatrix} r_0 & -r_x & -r_y & -r_z \\ r_x & r_0 & -r_z & r_y \\ r_y & r_z & r_0 & -r_x \\ r_z & -r_y & r_x & r_0 \end{bmatrix}}_{=:Q} \dot{q}$$

or

$$\dot{q}\dot{r} = \underbrace{\begin{bmatrix} r_0 & -r_x & -r_y & -r_z \\ r_x & r_0 & r_z & -r_y \\ r_y & -r_z & r_0 & r_x \\ r_z & r_y & -r_x & r_0 \end{bmatrix}}_{=:Q} \dot{q}$$

The dot product of two quaternions is the sum of products of corresponding components:

$$\dot{p} \cdot \dot{q} = p_0 q_0 + p_x q_x + p_y q_y + p_z q_z$$

The square of the magnitude of a quaternion is the dot product of the quaternion with itself:

$$||\dot{q}||^2 = \dot{q} \cdot \dot{q}$$

Taking the conjugate of a quaternion negates its imaginary part; thus

$$\dot{q}^* = q_0 - iq_x - jq_y - kq_z$$

It can be shown that finding the rotation matrix in Equation 2.26 is equal to find the unit quaternion \dot{q} , that maximizes

$$\sum_{i=1}^{N_p} \dot{x}'_i \cdot (\dot{q} \dot{p}'_i \dot{q}^*) = \sum_{i=1}^{N_p} (\dot{q} \dot{x}'_i) \cdot (\dot{p}'_i \dot{q})$$

Suppose that $x'_i = (x'_{x,i}, y'_{x,i}, z'_{x,i})^T$ while $p'_i = (x'_{p,i}, y'_{p,i}, z'_{p,i})^T$; then

$$\dot{q} \dot{x}'_i = \begin{bmatrix} 0 & -x'_{x,i} & -y'_{x,i} & -z'_{x,i} \\ x'_{x,i} & 0 & z'_{x,i} & -y'_{x,i} \\ y'_{x,i} & -z'_{x,i} & 0 & x'_{x,i} \\ z'_{x,i} & y'_{x,i} & -x'_{x,i} & 0 \end{bmatrix} = \bar{Q}_x \dot{q}$$

while

$$\dot{x}'_i \dot{q} = \begin{bmatrix} 0 & -x'_{p,i} & -y'_{p,i} & -z'_{p,i} \\ x'_{p,i} & 0 & -z'_{p,i} & y'_{p,i} \\ y'_{p,i} & z'_{p,i} & 0 & -x'_{p,i} \\ z'_{p,i} & -y'_{p,i} & x'_{p,i} & 0 \end{bmatrix} = Q_p \dot{q}$$

The sum that we have to maximize can now be written in the form

$$\begin{aligned} \sum_{i=1}^{N_p} (\dot{q} \dot{x}'_i) \cdot (\dot{p}'_i \dot{q}) &= \sum_{i=1}^{N_p} (\bar{Q}_x \dot{q}) \cdot Q_p \dot{q} \\ &= \sum_{i=1}^{N_p} (\dot{q}^T \bar{Q}_x^T Q_p \dot{q}) \\ &= \dot{q}^T \underbrace{\left(\sum_{i=1}^{N_p} \bar{Q}_x^T Q_p \right)}_{=N} \dot{q} \end{aligned}$$

It is convenient at this point to introduce the 3×3 matrix

$$M = \sum_{i=1}^{N_p} (x'_i - p'^T_i) = \begin{bmatrix} S_{xx} & S_{xy} & S_{xz} \\ S_{yx} & S_{yy} & S_{yz} \\ S_{zx} & S_{zy} & S_{zz} \end{bmatrix}$$

where $S_{xx} = \sum_{i=1}^{N_p} x'_{x,i} x'_{p,i}$, $S_{xy} = \sum_{i=1}^{N_p} x'_{x,i} y'_{p,i}$, , and so on. Then

$$N = \begin{bmatrix} S_{xx} + S_{yy} + S_{zz} & S_{yz} + S_{zy} & S_{zx} + S_{xz} & S_{xy} + S_{yx} \\ S_{yz} + S_{zy} & S_{xx} - S_{yy} - S_{zz} & S_{xy} + S_{yx} & S_{zx} + S_{xz} \\ S_{zx} + S_{xz} & S_{xy} + S_{yx} & -S_{xx} + S_{yy} - S_{zz} & S_{yz} + S_{zy} \\ S_{xy} + S_{yx} & S_{yz} + S_{zy} & S_{zx} + S_{xz} & -S_{xx} - S_{yy} + S_{zz} \end{bmatrix}$$

It can be shown that the unit quaternion that maximizes $\dot{q}^T N \dot{q}$ is the eigenvector corresponding to the most positive eigenvalue of the matrix N . Using this quaternion we can compute the rotation matrix with

$$R = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1 q_2 + q_3 q_0) & 2(q_1 q_3 - q_0 q_2) \\ 2(q_1 q_2 + q_0 q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2 q_3 - q_0 q_1) \\ 2(q_1 q_3 - q_0 q_2) & 2(q_2 q_3 - q_0 q_1) & q_0 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad (2.27)$$

and the translation

$$t = \mu_x + R\mu_p \quad (2.28)$$

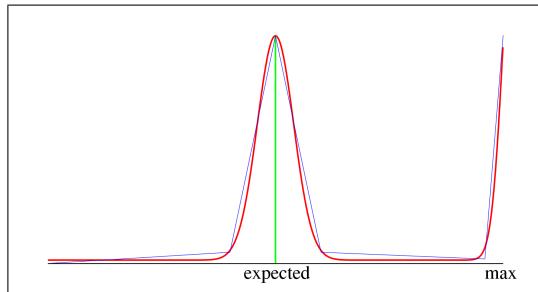


Figure 2.15: The probabilistic measurement model given as a mixture of a Gaussian and a uniform distribution (red) and its approximation by piecewise linear functions (blue).

The ICP algorithm calculates iteratively a minimum of Equation 2.22. In each iteration step, the algorithm selects the closest points as the model set, $X = \{x_i\}$ and calculates the transformation for minimizing Equation 2.22 according to Equation 2.27 and 2.28.

The key properties of the ICP algorithm can be summarized as follows. First, it converges to the nearest local minimum of the sum of squared distances between the closest points. Therefore, the final location is not necessarily equivalent to the optimal position. Furthermore, a good initial estimation of the transformation between point sets is required to ensure convergence to a good position. Finally, the ICP algorithm cannot deal with maximum-range measurements. Accordingly, it throws away a serious amount of information.

2.4.2 Probabilistic Three-dimensional Scan Registration using Geometric Maps

At the core of the probabilistic range scan registration is an approximative physical model of the range scanning process. Obviously, an ideal sensor would always measure the correct distance to closest obstacle in the sensing direction. However, sensors and models generated out of range scanners are noisy. Therefore, our current model incorporates measurement noise and random noise in order to deal with errors typically found in 3D range scans. First, we generally have normally distributed measurement errors around the distance “expected” according to the current position of the scanner and the given the model of the environment (e.g. the previous scan). Additionally, we observe randomly distributed measurements because of errors in the model and because of deviations in the angles between corresponding beams in consecutive scans. Therefore, our model consist of a mixture of a Gaussian with a uniform distribution (see Figure 2.15). The mode of the Gaussian corresponds to the distance expected according to the current state of the robot and the given surface. In practice, the values of the other

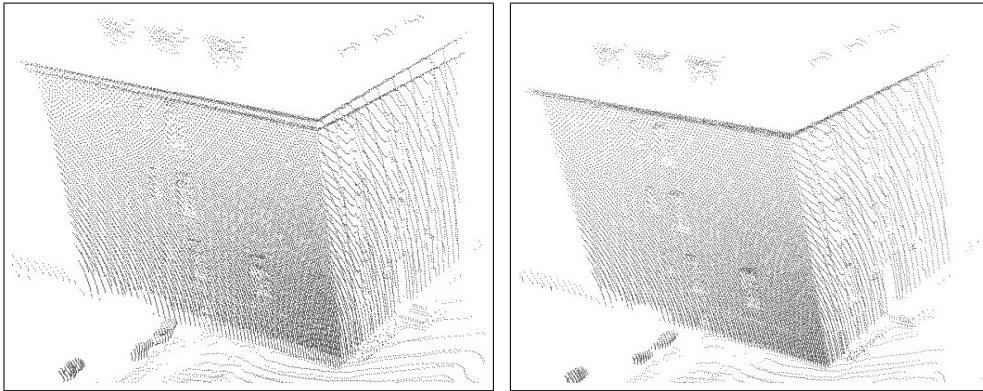


Figure 2.16: Details of two scans registered with ICP (left) and our probabilistic method (right).

two parameters, i.e. the variance of the Gaussian and the height of the uniform have to be adapted to the actual accuracy of the sensor uses (see also [Fox *et al.*, 1999b]). Additionally, to speed-up the computation, we use a piecewise linear approximation of this density.

To compute the likelihood of a beam, we compute the distance d_e to the closest obstacle in the measurement direction. Then we compute the probability of the measured distance d_m given the expected distance, i.e. we determine the quantity $P(d_m | d_e)$ using the mixture computed for d_e . Under the assumption that all beams in S are independent given the model S' , the likelihood of a complete sweep S given S' and a pose x is then computed according to

$$p(S | S', x) = \prod_{s \in S} p(s | d_e(s, S', x)) \quad (2.29)$$

Here $d_e(s, S', x)$ is the distance expected according to the beam direction of s , the previous scan S' , and the location of the scanner x . To compute this quantity, our current system constructs a surface out of S' by introducing a triangle between adjacent points whose distance does not exceed a certain threshold which itself depends on the length of the beams. To determine the most likely position of the scan S given a reference scan S' we apply a hill-climbing strategy similar to that of ICP. We repeatedly change the pose x of the scan S as long as the likelihood $p(S | S', x)$ increases.

Compared to the ICP algorithm and other scan-matching techniques, this approach has several advantages. First, it exploits the fact that each laser beam is a ray that does not go through surfaces and therefore does not require special heuristics for dealing with occlusions. Please note that there are variants of the ICP [Rusinkiewicz and Levoy, 2001] that match a scan to the corresponding closest points on the surface. However, even those approaches cannot correctly deal

with occlusions. Second, our approach exploits maximum range readings since beams going through surfaces reduce the likelihood of an alignment. Figure 2.16 shows an example of two scans registered with ICP and our probabilistic method. While the ICP algorithm stopped in a local maximum and did not converge to the correct position, the probabilistic method did. Although computing the expected distances can efficiently be computed using ray-tracing techniques based on a spacial tiling and indexing [Samet, 1990], computing the closest points, as it is necessary for the ICP algorithm, has less computational costs. Therefore the ICP algorithm is often faster than the probabilistic method, but as a drawback it may converge in a position with a higher mean squared error.

CHAPTER 3

Map Building with Mobile Robot in Populated Environments

3.1 Introduction

Whereas most of todays mapping systems are able to deal with noise in the odometry and noise in the sensor data, they usually assume that the environment is static during mapping. However, if a person walks through the sensor range of the robot during data acquisition, the resulting map will contain evidence about an object at the corresponding location. If the robot returns to the corresponding location later on and scans the area a second time, its pose estimate will be less accurate, since the new measurement does not contain features corresponding to the person. Thus, people walking through the scene result in spurious objects in the map and simultaneously make the localization problem harder. Moreover, the reduced accuracy of the resulting maps may have a negative influence on the overall performance of the robot, since it can obstruct the execution of typical navigation tasks such as localization and path planning.

In this chapter, we present a new algorithm for mapping with mobile robots in populated environments. Our approach uses sample-based joint probabilistic data association filters for tracking people in the vicinity of the robot. The belief about the potential positions of persons is used to improve the registration process and to remove spurious objects from the map being learned. We apply our approach to 2D and 3D data obtained with laser-range scanners. In practical experiments, we demonstrate that our algorithm yields accurate models of the environment by filtering out beams reflected by persons.

This chapter is organized as follows. After describing briefly our approach to

tracking multiple people in range scans in the following section, we will present our mapping technique and how the results of the people tracking approach are integrated into the mapping process in Section 3.3. Section 3.4 contains several experiments illustrating the advantages of our approach to learning accurate 2D and 3D maps with range scanners. Finally, Section 3.5 contains the related work.

3.2 Tracking People in Range Scans

To detect people and track them in the vicinity of the robot, our system applies a sample-based variant of Probabilistic Data Association Filters (JPDAFs) [Cox, 1993]. This approach is described in the following section. Additionally, we describe how we adapt the number of persons being tracked and how to implement this technique using data gathered with the laser-range finders of a moving mobile robot. Furthermore details above this people tracking technique can be found in [Schulz *et al.*, 2003].

3.2.1 Sample-based Joint Probabilistic Data Association Filters (SJPDAFs)

Suppose there are K persons in the environment and let $H_t = \{h_t^1, \dots, h_t^K\}$ be the states of these persons at time t . Note that we use h for the state instead of x . Later we will introduce x as the pose of the robot and we want to avoid additional indices. Therefore we will use two different variable names for states: h for the state of the person and x for the pose of the robot. Note that each h_t^i is a random variable ranging over the state space of a single person. Furthermore, let $\Theta_t = \{\theta_t^1, \dots, \theta_t^{m_t}\}$ denote a feature set observed at time t , where θ_t^j is one feature of such a set. $\Theta_{1:t}$ is the sequence of all feature sets up to time t . The key question when tracking multiple persons is how to assign the observed features to the individual objects.

In the JPDAF framework, a joint association event γ is a set of pairs $(j, i) \in \{0, \dots, m_t\} \times \{1, \dots, K\}$. Each γ uniquely determines which feature is assigned to which object. Please note, that in the JPDAF framework, the feature θ_t^0 is used to model situations in which an object has not been detected, i.e. no feature has been found for object i . Let Γ_{ji} denote the set of all valid joint association events which assign feature j to the object i . At time t , the JPDAF considers the posterior probability that feature j is caused by object i :

$$\gamma_{ji} = \sum_{\gamma \in \Gamma_{ji}} P(\gamma | \Theta_{1:t}). \quad (3.1)$$

According to [Schulz *et al.*, 2001], we can compute the γ_{ji} as

$$\gamma_{ji} = \sum_{\gamma \in \Gamma_{ji}} \alpha \gamma^{(m_t - |\gamma|)} \prod_{(j,i) \in \gamma} p(\theta_t^j | h_t^i). \quad (3.2)$$

It remains to describe, how the beliefs $p(h_t^i)$ about the states of the individual objects are represented and updated. In this approach [Schulz *et al.*, 2001], we use sample-based representations of the individual beliefs. The key idea underlying all particle filters is to represent the density $p(h_t^i | \Theta_{1:t})$ by a set Θ_t^i of N weighted, random samples or *particles*. A sample set constitutes a discrete approximation of a probability distribution. Each sample is a tuple $(h_t^{i,n}, w_t^{i,n})$ consisting of state $h_t^{i,n}$ and an importance factor $w_t^{i,n}$. The *prediction* step is realized by drawing samples from the set computed in the previous iteration and by updating their state according to the prediction model $p(h_t^i | h_{t-1}^i, \delta t)$. In the *correction* step, a feature set Θ_t is integrated into the samples obtained in the prediction step. Thereby we consider the assignment probabilities γ_{ji} . In the sample-based variant, these quantities are obtained by integrating over all samples:

$$p(\theta_t^j | h_t^i) = \frac{1}{N} \sum_{n=1}^N p(\theta_t^j | h_t^{i,n}). \quad (3.3)$$

Given the assignment probabilities we now can compute the weights of the samples

$$w_t^{i,n} = \alpha \sum_{j=0}^{m_t} \gamma_{ji} p(\theta_t^j | h_t^{i,n}), \quad (3.4)$$

where α is a normalizer ensuring that the weights sum up to one over all samples. Finally, we obtain N new samples from the current samples by resampling. For this purpose we select every sample $h_t^{i,n}$ with probability $w_t^{i,n}$.

3.2.2 Dealing with Variable Numbers of Persons

To adapt our algorithm to the number of people in the perceptual field of the robot we apply a Bayesian filtering process and maintain a density $P(N_t | m_{1:t})$ over the number of objects N_t at time t , where $m_{1:t} = m_1, \dots, m_t$ is the sequence of the numbers of features observed so far

$$P(N_t | m_{1:t}) = \alpha \cdot P(m_t | N_t) \cdot \sum_n P(N_t | N_{t-1} = n) \cdot P(N_{t-1} = n | m_{1:t-1}). \quad (3.5)$$

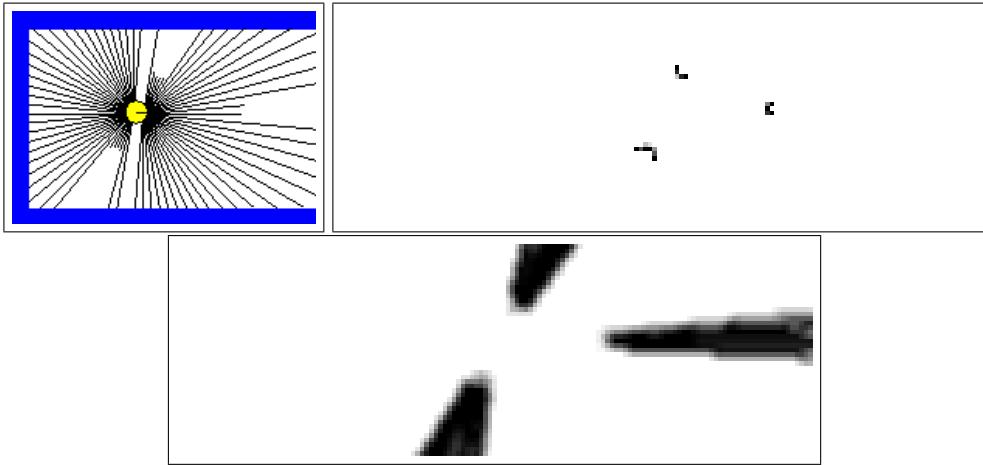


Figure 3.1: Typical laser range finder scan. Two of the local minima are caused by people walking by the robot (upper left image). Features extracted from the scan, the grey-level represents the probability that a person's legs are at the position (upper right image). Occlusion grid, the grey-level represents the probability that the position is occluded (lower image).

All we need to know to implement this equation are the quantities $P(m_t \mid N_t)$ and $P(N_t \mid N_{t-1})$. The term $P(m_t \mid N_t)$ represents the probability of observing m_t features, if N_t objects are in the perceptual field of the sensor. According to [Schulz *et al.*, 2001], this quantity is learned from data. To adapt the number of objects in the SJPDAF, our system uses the maximum likelihood estimate of $P(N_t \mid m_{1:t})$. If the number of particle filters in the SJPDAF is smaller than the new estimate for N_t , new filters need to be initialized. Since we do not know which of the features originate from new objects, we initialize the new filters using a uniform distribution. We then rely on the SJPDAF to disambiguate this distribution during subsequent filter updates.

If the most recent estimate for N_t is smaller than the current number of particle filters, the corresponding number of filters needs to be decreased. This, however, requires that we know which filter does not track an object any longer. To estimate the tracking performance of a sample set, we accumulate a discounted average \hat{w}_t^i of the sum of sample weights w_t^i before the normalization step:

$$\hat{w}_t^i = (1 - \delta)\hat{w}_{t-1}^i + \delta w_t^i. \quad (3.6)$$

Since the sum of sample weights decreases significantly, whenever a filter is not tracking any feature contained in the measurement, we use this value as an indicator that the corresponding object has left the perceptual field of the robot. Whenever we have to remove a filter, we choose that one with the smallest discounted average \hat{w}_t^i .

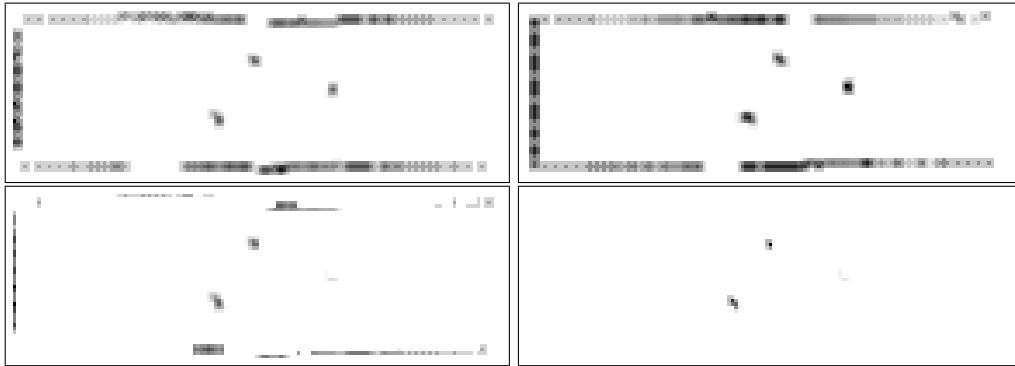


Figure 3.2: From left to right, top-down: the occupancy map for the current scan, the occupancy map for the previous scan, the resulting difference map, and the fusion of the difference map with the feature maps for the scan depicted in Figure 3.1.

3.2.3 Laser-based Implementation

In our system we apply the SJPDAF to estimate the trajectories of persons in range scans. Since the laser range scanners mounted on our platforms are at a height of approx. 40 cm, the beams are reflected by the legs of the people which typically appear as local minima in the scans. These local minima are used as the features for the SJPDAF (see upper images of Figure 3.1). Unfortunately, there are other objects which produce patterns similar to people. To distinguish these static objects from moving people our system additionally considers the differences between occupancy probability grids built from consecutive scans. This whole process is illustrated in Figure 3.2. Please note, that we also perform a scan-matching to align each pair consecutive scans. Therefore, the static aspects of the environment can be identified and filtered out accurately.

Finally, we have to deal with possible occlusions. We therefore compute a so-called “occlusion map” containing for each position in the vicinity of the robot the probability that the corresponding position is not visible given the current range scan. See right part of Figure 3.1. The information about occluded areas is used to avoid that the SJPDAF loses track of a person whenever it is temporarily occluded.

Figure 3.3 shows a typical situation, in which a robot equipped with two laser-range scanners is tracking up to four persons in its vicinity. As can be seen from the figure, our approach is robust against occlusions and can quickly adapt to changing situations in which additional persons enter the scene. For example, in the lower left image the upper right person is not visible in the range scan, since it is occluded by the person that is close to the robot. The knowledge that the samples lie in an occluded area prevents the robot from deleting the corresponding sample set. Instead, the samples only spread out, which represents the growing

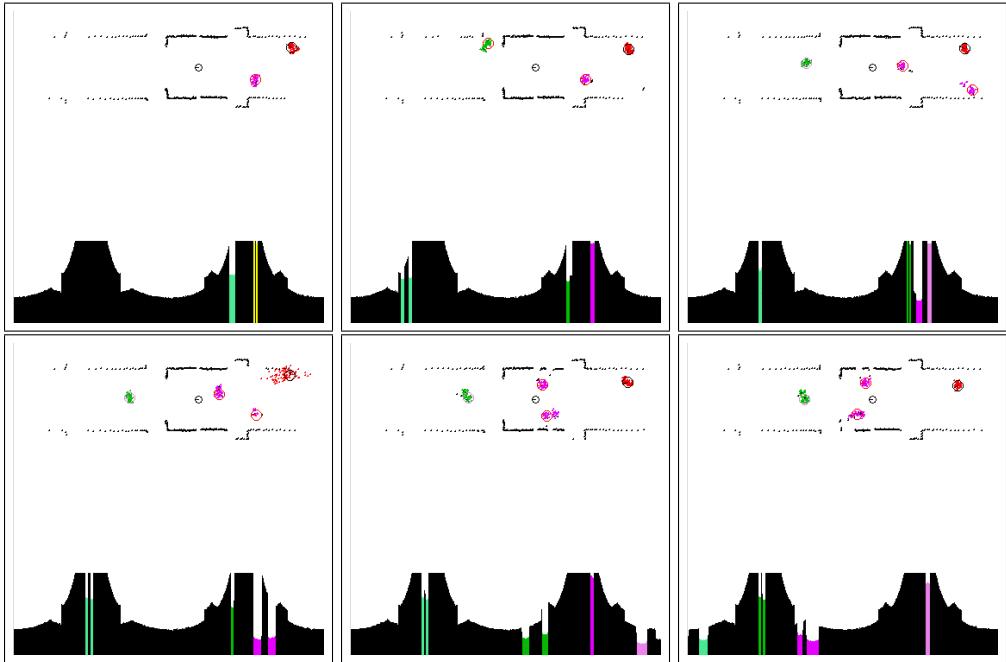


Figure 3.3: Tracking people using laser range-finder data. Upper part shows the raw sensor data with the colored sample sets. Lower part shows the range values with the local minima marked with the color of the corresponding sample set.

uncertainty of the robot about the position of the person.

3.3 Computing Consistent Maps

As described in Chapter 2 our current system is able to learn 2D and 3D maps using range scans recorded with a mobile robot. In both cases, the approach is incremental. Mathematically, we calculate a sequence of poses $\hat{x}_1, \hat{x}_2, \dots$ and corresponding maps by maximizing the marginal likelihood of the t -th pose and map relative to the $(t-1)$ -th pose and map:

$$\hat{x}_t = \underset{x_t}{\operatorname{argmax}} \{ p(z_t | x_t, \hat{m}(\hat{x}_{1:t-1}, z_{1:t-1})) \cdot p(x_t | u_{t-1}, \hat{x}_{t-1}) \} \quad (3.7)$$

In this equation the term $p(z_t | x_t, \hat{m}(\hat{x}_{1:t-1}, z_{1:t-1}))$ is the probability of the most recent measurement z_t given the pose x_t and the map $\hat{m}(\hat{x}_{1:t-1}, z_{1:t-1})$ constructed so far. The term $p(x_t | u_{t-1}, \hat{x}_{t-1})$ represents the probability that the robot is at location x_t given the robot was previously at position \hat{x}_{t-1} and has carried out (or measured) the motion u_{t-1} . The resulting pose \hat{x}_t is then used to generate a new map \hat{m} via the standard incremental map-updating function presented

in [Moravec, 1988]:

$$\hat{m}(\hat{x}_{1:t}, z_{1:t}) = \underset{m}{\operatorname{argmax}} p(m \mid \hat{x}_{1:t}, z_{1:t}) \quad (3.8)$$

The overall approach can be summarized as follows: At any point $t - 1$ in time the robot is given an estimate of its pose \hat{x}_{t-1} and a map $\hat{m}(\hat{x}_{1:t-1}, z_{1:t-1})$. After the robot moved further on and after taking a new measurement z_t , the robot determines the most likely new pose \hat{x}_t . It does this by trading off the consistency of the measurement with the map (first term on the right-hand side in (3.7)) and the consistency of the new pose with the control action and the previous pose (second term on the right-hand side in (3.7)). The map is then extended by the new measurement z_t , using the pose \hat{x}_t as the pose at which this measurement was taken.

More details on the used scan matching technique can be found in the previous Chapter 2. It remains to describe how we actually maximize Equation (3.7). Our system applies two different approaches depending on whether the underlying scans are 2D or 3D scans.

3.3.1 Integrating People Tracking Results into the Map Building Process

The goal of integrating the results of the people tracker into a mapping process can be divided in two subjects:

1. to improve the alignment between the scans and
2. to filter out corrupted measurements originating from people walking in the vicinity of the robot.

To consider the estimated states of the persons during the scan alignment, we need to know the probability $P(\text{hit}_{x,y} \mid H_t)$ that a beam ending at position $\langle x, y \rangle$ is reflected by a person. In our current implementation, we consider the individual persons independently:

$$P(\text{hit}_{x,y} \mid H_t) = 1 - \prod_{i=1}^K (1 - P(\text{hit}_{x,y} \mid h_t^i)). \quad (3.9)$$

In this equation $P(\text{hit}_{x,y} \mid h_t^i)$ describes the likelihood that a beam ending at position $\langle x, y \rangle$ is reflected by person i , given the state h_t^i of that person. To compute this quantity, we construct a two-dimensional and normalized histogram by counting how many samples representing the belief about h_t^i fall into each bin.

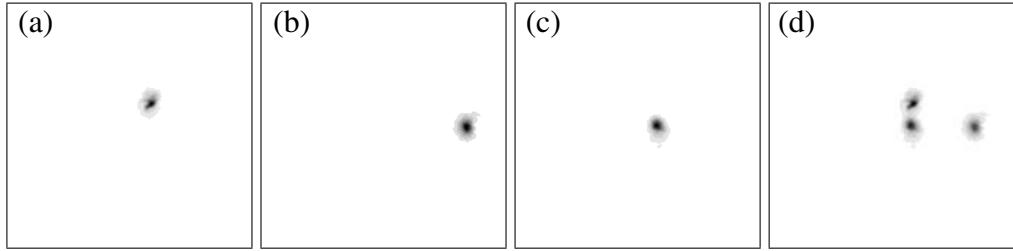


Figure 3.4: Example situation in which the quantity $P(\text{hit}_{x,y} | H_t)$ (image d) is computed by combining the histograms for three individual trackers (image a-c).

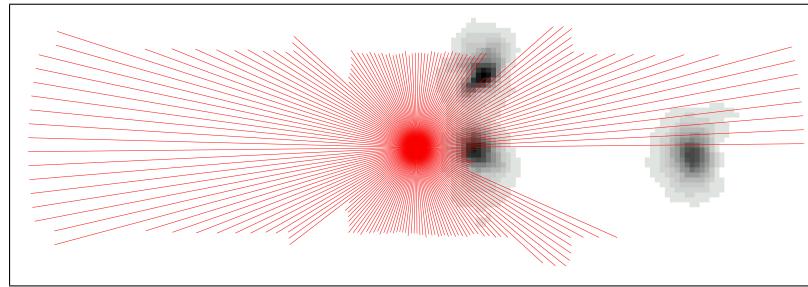


Figure 3.5: The weight of a laser beam is computed according to the value $P(\text{hit}_{x,y} | H_t)$ of the cell in which it ends.

Now suppose x_b and y_b are the coordinates of the cell in which the beam b ends. Accordingly, we can compute the probability $P(\text{hit}_b | h_t^i)$ that a beam b is reflected by a person as

$$P(\text{hit}_b | H_t) = P(\text{hit}_{x_b, y_b} | H_t). \quad (3.10)$$

It remains to describe, how we incorporate the quantity $q_b = P(\text{hit}_b | H_t)$ into the scan alignment and registration process. We consider all beams as independent and compute the likelihood $p(z_t | x_t, \hat{m}(\hat{x}_{1:t-1}, z_{1:t-1}))$ of the most recent measurement given all previous scans is obtained as

$$p(z_t | x_t, \hat{m}(\hat{x}_{1:t-1}, z_{1:t-1})) = \prod_{b \in z_t} p(b | \hat{m}(\hat{x}_{1:t-1}, z_{1:t-1})))^{(1-q_b)}. \quad (3.11)$$

Thus, during the scan alignment we weigh each beam b according to the probability $1 - P(\text{hit}_b | H_t)$. Please note that this is a general form of a situation in which it is exactly known whether or not b is reflected by a person. If b is known to be reflected by a person, q_b equals 1 such that b does not change the likelihood of the scan (see Figure 3.4 and Figure 3.5).

The second task is to filter out beams reflected by persons to avoid spurious objects in the resulting maps. In our current system we compute a bounding box



Figure 3.6: Robot Sam mapping the populated exhibition hall of the Byzantine Museum in Athens.

for each sample set w_t^i and integrate only those beams whose endpoints do not lie in any of the bounding boxes. To cope with the possible time delay of the trackers, we also ignore corresponding beams of several previous and next scans before and after the person was detected. Please note, that one generally can be more conservative during the map generation process, because the robot typically scans every part of the environment quite often. However, during scan alignment, a too conservative strategy may result in too few remaining beams which leads to reduced accuracy of the estimated positions.

3.4 Experiments

The approach has been implemented and tested on different robotic platforms and based on extensive off-line experiments carried out with recorded data. The goal of the experiments described in this section is to illustrate that the integration of people detection techniques into the mapping process leads to better maps since the resulting alignments are more accurate and since beams reflected by persons are filtered out which reduces the number of spurious objects. Please note that our current implementation can track several people in real-time, so that the time to map an environment is not influenced by using this information.

3.4.1 Learning 2D Maps

The first experiments were carried out using the Pioneer 2 robot Sam in an exhibition hall of the Byzantine Museum in Athens, Greece. The size of this environment is 30m x 45m. The robot traveled continuously 57m with an average speed of

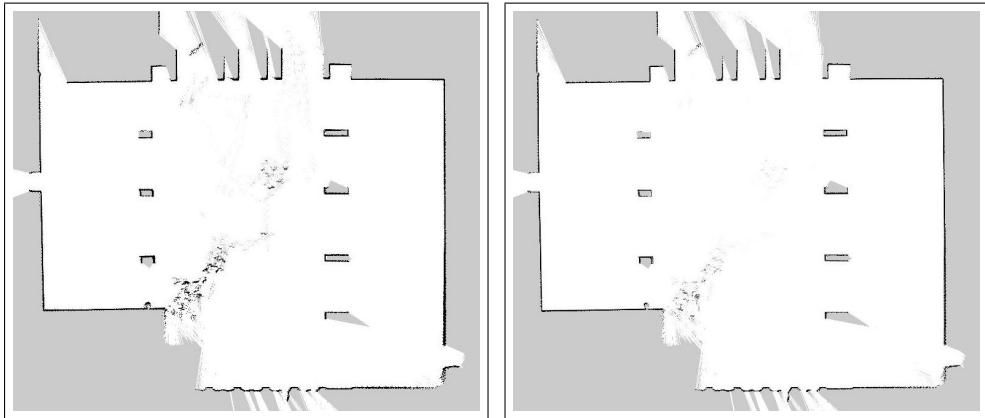


Figure 3.7: Maps of the Byzantine Museum in Athens created without (left image) and with people filtering (right image).

0.37m/s and a maximum speed of 0.96m/s. Figure 3.6 shows the robot during the mapping process. There were 15 people walking through the environment while the robot was mapping it. The map obtained without filtering measurements reflected by persons is shown in the left image of Figure 3.7. The result obtained with our new algorithm is shown in the right image of Figure 3.7. Both maps are high-resolution occupancy grid maps with a resolution of 2cm per cell. As can be seen from the figures the number of spurious readings is reduced considerably. The remaining spurious objects come from a crowd of people that did not move during the mapping process. Accordingly, they could not be filtered out by our algorithm. Furthermore, it should be mentioned, that the museum hall contains several columns which produce similar features in the range scans as people. Nevertheless, our approach could seriously reduce the number of readings corrupted by people.

We carried out a second experiment using our iRobot B21 robot Rhino in a 25m x 4m large corridor environment of the Computer Science Department in Bonn. Again, while the robot was gathering the data, there were several (up to five) persons walking through the environment (see Figure 3.8). The map obtained without people filtering is shown in the left image of Figures 3.9. As can be seen from the figure, there are a lot of cells in the resulting grid map, which have a high occupancy probability since people covered the corresponding area while the robot was mapping the environment. If, however, we use our new algorithm and filter out the beams corresponding to persons, the effect of the persons is seriously reduced in the resulting map (see lower image of Figures 3.9). In this experiment the overall reduction of beams reflected by the persons is 97.5%.



Figure 3.8: Robot Rhino mapping the populated corridor environment at the University of Bonn.

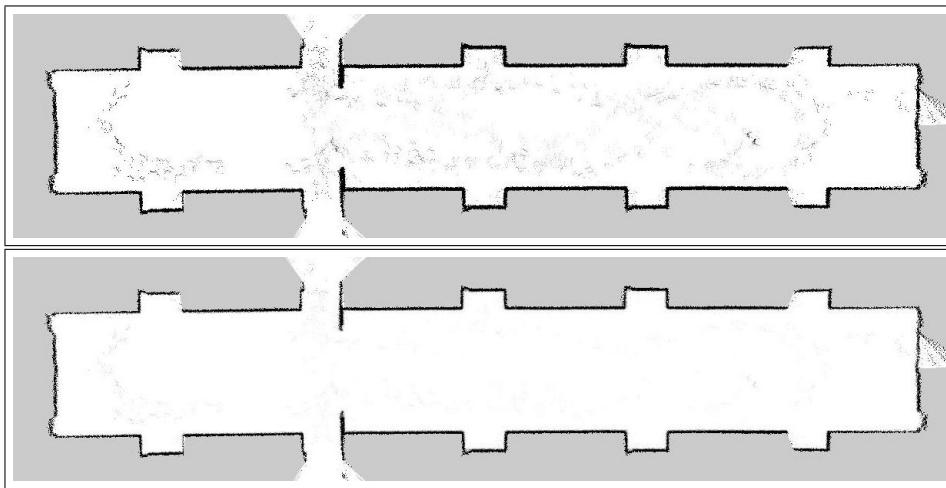


Figure 3.9: Occupancy grid maps created for the populated corridor environment of the University of Bonn without (upper image) and with people filtering (lower image).

3.4.2 Improved Robustness

Besides the fact that the resulting maps are better, filtering people increases the robustness of the mapping process. To demonstrate this we have carried out a series of experiments in which we added random noise to the poses of the robot in the input data and compared the performance of our mapping strategy with and without people filtering. We performed 50 experiments for each noise level. Figure 3.10 shows the numbers of maps containing a translational error larger than 2m for the different noise values. In this figure the x-axis corresponds to the standard deviation of the Gaussian noise added to each odometry reading. The

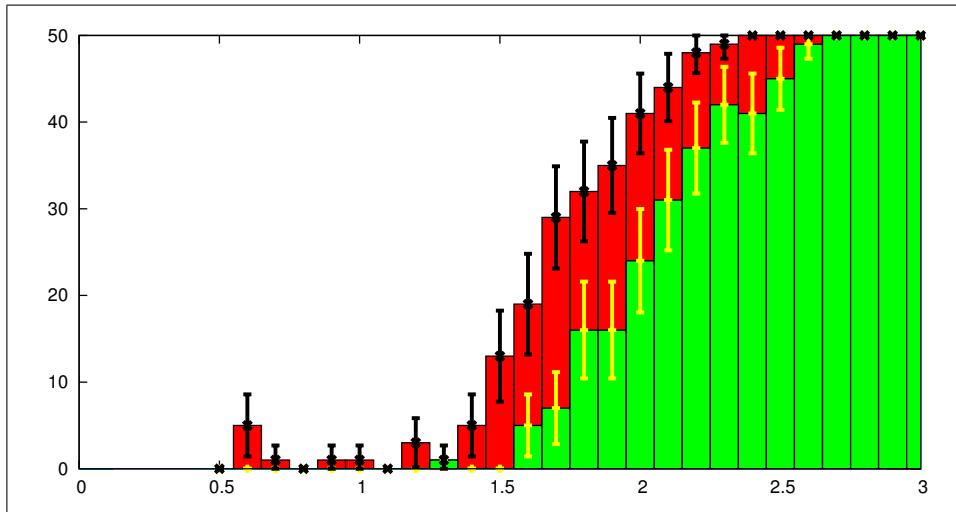


Figure 3.10: Number of maps with translational error larger than 2m computed without people filtering (red / dark grey) and using people filtering (green / light grey) for increasing levels of noise in odometry.

y-axis is the number of maps in which the translational error after registration exceeds 2m. As can bee seen by the figure, the use of the information provided by the people tracker significantly increases the accuracy of the position estimation during the mapping process.

3.4.3 Learning 3D Maps

The last experiment was carried out to analyze the performance of our system when learning three-dimensional maps. For this experiment we used the Pioneer 2 AT platform (see Figure 3.11 (left)) equipped with two laser range-scanners. Whereas the first scanner, that is mounted in front of the robot, is used for tracking people, the second scanner is mounted on an AMTEC wrist module to scan the 3D structure of the environment. Figure 3.11 (right) shows a typical scenario during this experiment performed on our university campus. Here, several people were walking through the scene while the robot was scanning it. Figure 3.13 (upper image) depicts the model obtained after aligning two scans of the same environment. In this model, the people appear as three-dimensional curves. Figure 3.12 contains a magnified view of the corresponding portion of the map. If we integrate the information obtained from the people tracker, however, these spurious objects are completely removed (see lower image of Figure 3.13). The number of triangles in these models are 416.800 without filtering and 412.500 with filtering. Please note, that this experiment also illustrates the advantage of using a



Figure 3.11: Pioneer 2 AT robot Herbert for 3D outdoor mapping (left) and typical situation in which people walk through the scene during mapping (right).

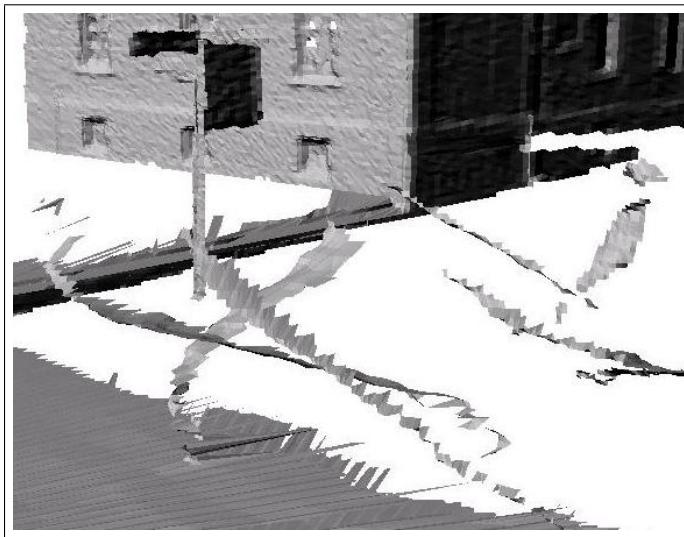


Figure 3.12: Spurious objects caused by people walking through the environment.

tracking system over a pure feature-based approach. Due to the displacement of the scanners, people are not always visible in both of them. Accordingly, a purely feature-based approach like [Wang and Thorpe, 2002] will add objects to the 3D model whenever they are not detected by the first scanner. Our system, however, can predict positions of persons in the case of occlusions and thus can filter out the corresponding readings even if the features are missing.

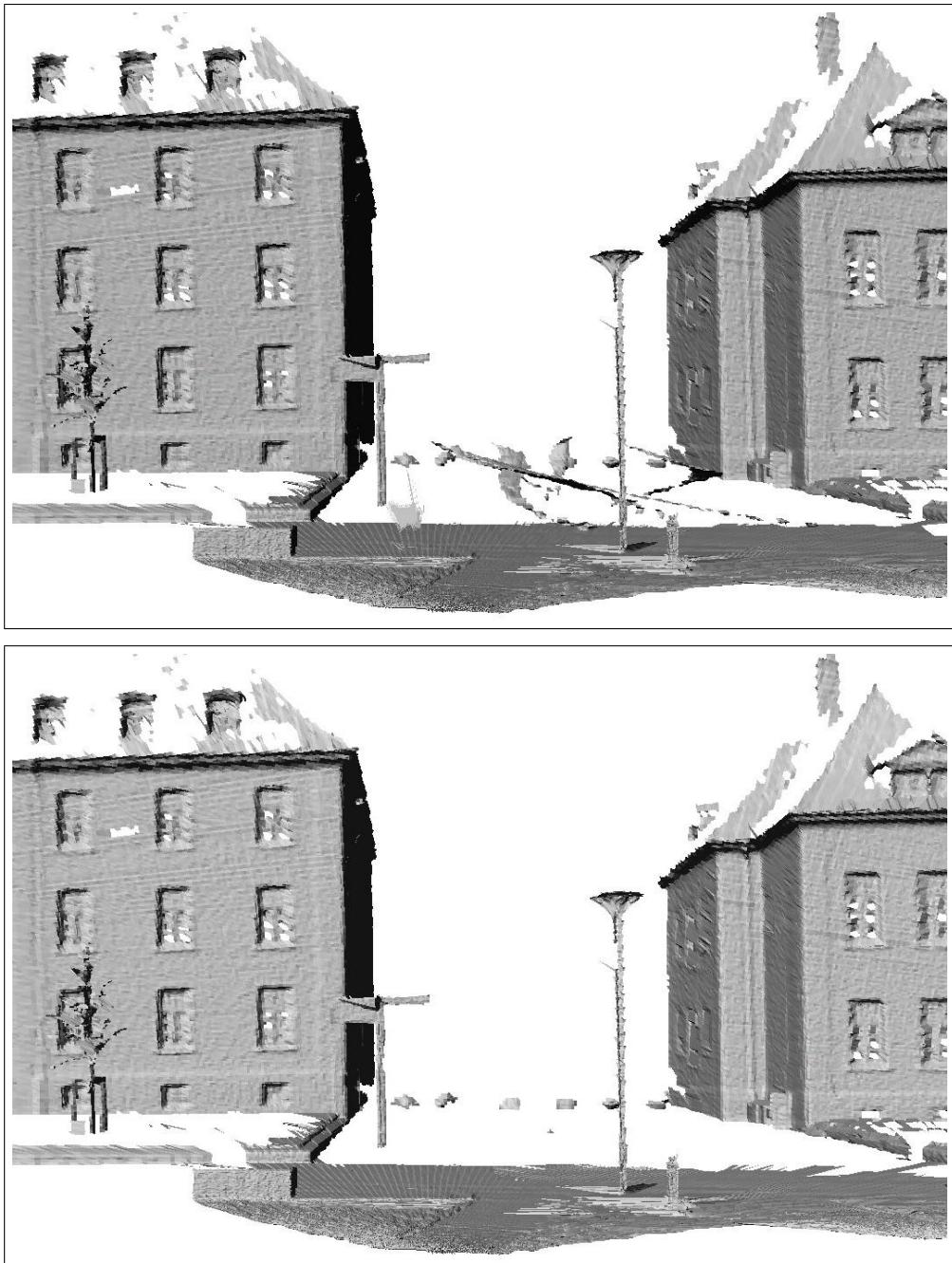


Figure 3.13: Three-dimensional map of a building (upper image) and people filtered (lower image).

3.5 Related Work

Approaches to concurrent mapping and localization can roughly be classified according to the kind of sensor data processed and the matching algorithms used. For example, the approaches described in [Shatkay, 1998, Castellanos *et al.*, 1999, Dissanayake *et al.*, 2000, Leonard and Feder, 1999] extract landmarks out of the data and match these landmarks to localize the robot in the map being learned. The other set of approaches such as [Lu and Milios, 1997, Gutmann and Konolige, 1999, Thrun, 2001] use raw sensor data and perform a dense matching of the scans. Although all approaches possess the ability to cope with a certain amount of noise in the sensor data, they assume that the environment is almost static during the mapping process. Especially in populated environments, additional noise is introduced to the sensor data which increases the risk of localization errors. Additionally, people in the vicinity of the robot appear as objects in the resulting maps and therefore make the maps not usable for path planning etc. Recently Wang and Thorpe [Wang and Thorpe, 2002] presented a heuristic and feature-based approach to identify dynamic objects in range scans. The corresponding measurements are then filtered out during 2D scan registration. Our approach instead uses a tracking technique and therefore is able to predict the positions of the persons even in situations in which the corresponding features are temporarily missing.

Additionally, there has been work on updating maps or improving localization in populated environments. For example, Burgard *et al.* [Burgard *et al.*, 2000] update a given static map using the most recent sensory input to deal with people in the environment during path planning. Montemerlo and Thrun [Montemerlo and Thrun, 2002] present an approach to simultaneous localization and people tracking. Arras *et al.* [Arras *et al.*, 2002] present a team of tour-guide robots that operates in a populated exhibition. Their system uses line features for localization and has been reported to successfully filter range-measurements reflected by persons. Fox *et al.* [Fox *et al.*, 1999b] present a probabilistic technique to identify range measurements which do not correspond to the given model of the environment. These approaches, however, require a given and fixed map which is used for localization and for the extraction of the features corresponding to the people. Our technique, in contrast, does not require a given map. Rather it learns the map from scratch using the data acquired with the robot's sensors.

Furthermore, a variety of techniques for tracking persons, for predicting future poses of persons, or for adopting the behavior of the robot according to the information obtained about the persons in its vicinity has been developed [Tadokoro *et al.*, 1995, Kruse and Wahl, 1998, Zhu, 1991, Lavalle *et al.*, 1997, Bui *et al.*, 2001, Kluge *et al.*, 2001, Kortenkamp *et al.*, 1996, Kahn *et al.*, 1996, Waldherr *et al.*, 1998, Beymer and K., 2001, Bennewitz *et al.*, 2002, Stachniss and Burgard, 2002]. All these approaches, however, do not filter the measurements correspond-

ing to persons in order to improve the model of the environment.

3.6 Summary

In this chapter we presented a probabilistic approach for mapping in populated environments. The key idea of this technique is to use Sample-based Joint Probabilistic Data Association Filters (SJPDAFs) to track people in the data obtained with the sensors of the robot. The results of the people tracking are integrated into the scan alignment and into the map generation process. This leads to two improvements. First, the resulting pose estimates are more accurate and second, the resulting maps contain less spurious objects compared to the maps created without filtering people.

Our technique has been implemented and tested on different robotic platforms as well as for generating 2D and 3D maps. In practical experiments we demonstrated that this technique is able to filter out beams corrupted by people walking through the environment. Additionally, extensive simulation experiments illustrated that the pose estimates are significantly better if the results of the tracking system are considered during the pose estimation.

CHAPTER 4

Mapping in Dynamic Environments

4.1 Introduction

The approach described in the previous chapter is mostly tailored to humans, in practice however one finds a wide variety of dynamic elements. For example, doors, cars, bicycles, or furniture, which change the location over time, are not modeled in this previous approach. All these objects do not fit to the shape of two human legs or to the motion behavior of persons.

In this chapter we present an alternative algorithm to mapping with mobile robots in dynamic environments. This approach does not rely on predicted features, rather it applies the popular Expectation Maximization (EM) algorithm. In the expectation step we compute a probabilistic estimate about which measurements might correspond to static objects. In the maximization step we use these estimates to determine the position of the robot and the map. This process is iterated until no further improvement can be achieved.

We apply this approach to 2D and 3D data obtained with laser-range scanners. In practical experiments we demonstrate that the algorithm can reliably filter out dynamic aspects and yields accurate models of the environment. A further advantage of the algorithm is that the filtered data can be extracted from the rest of all measurements. This way, we can obtain accurate textured 3D models of dynamic objects.

This chapter is organized as follows. We describe the EM-based procedure to learn which measurements correspond to static aspects of the environment in the following section. In Section 4.3 we will present several experiments illustrating that this approach can successfully learn 2D and 3D maps with range scanners in dynamic environments. Finally in Section 4.4 we will discuss the related work.

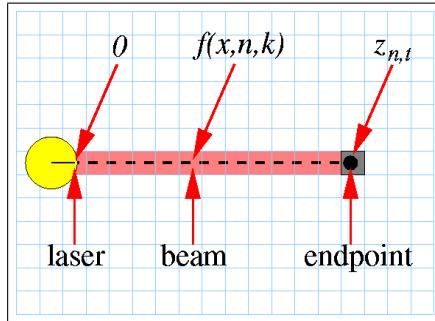


Figure 4.1: Beam covering $z_{t,n}$ cells of a map.

4.2 Learning Maps in Dynamic Environments

Our approach to discover measurements that correspond to dynamic objects is strictly statistical and does not rely on extracted features. We use the popular EM-algorithm to identify data items that cannot be explained by the rest of the data set. The input to this routine is a sequence of data items $z = \{z_1, \dots, z_T\}$. The output is a model m obtained from these data items after incorporating the estimates about spurious measurements. In essence, this approach seeks to identify a model m that maximizes the likelihood of the data. Often each measurement z_t consists of multiple data $z_{t,1}, \dots, z_{t,N}$, which is, for example, the case for laser-range scans. Throughout this chapter we assume that the data $z_{t,n}$ are the distance information obtained by the individual beams of a laser-range scan.

To accurately map a dynamic environment we need to know which measurements are caused by dynamic objects and therefore can safely be ignored in the alignment and map updating phase. To characterize spurious measurements in the data we introduce additional variables $c_{t,n}$ that tell us for each t and each n , whether the data item $z_{t,n}$ is caused by a static object or not. Each such variable $c_{t,n}$ is a binary variable, that is either 0 or 1. It is 1 if and only if the $z_{t,n}$ is caused by a static object. The vector of all these variables will be denoted by c .

In Section 2.2.3 we introduced the *counting model*. Here we will present an extended model, which is also able to deal with reflections caused by dynamic objects. In the case that all measurements are caused by static objects the previously introduced model can be deducted from the extended version.

For the sake of simplicity, we give the derivation for beams that are parallel to the x-axis of the map. In this case, the length $z_{t,n}$ directly corresponds to the number of cells covered by this beam. We will later describe how to deal with beams that are not parallel to the x-axis. Let f be a function that returns for each pose x_t of the robot, each beam number n , and each $k \leq z_{t,n}$ the index $f(x_t, n, k)$ of k -th field covered by that beam in the map (see Figure 4.1 or Figure 2.5 in

Chapter 2). To determine whether or not a beam is reflected by a dynamic object, we need to define the likelihood of a measurement given the current map m of the environment, the pose x of the robot, and the information whether or not $z_{t,n}$ is a maximum range reading. Maximum-range readings have to be treated differently, since those measurements generally are not reflected by any object. Like in Chapter 2 we use indicator variables $\zeta_{t,n}$ which are 1 if and only if $z_{t,n}$ is a maximum range reading and 0 otherwise. The likelihood of a measurement $z_{t,n}$ given the value of $c_{t,n}$ and the map m can thus be computed as:

$$\begin{aligned} p(z_{t,n} \mid c_{t,n}, x_t, m) &= \\ &\prod_{k=0}^{z_{t,n}-1} (1 - m_{f(x_t, n, k)})^{\zeta_{t,n}} \\ &\cdot \left[[m_{f(x_t, n, z_{t,n})}]^{c_{t,n}} \cdot [1 - m_{f(x_t, n, z_{t,n})}]^{(1-c_{t,n})} \cdot \prod_{k=0}^{z_{t,n}-1} (1 - m_{f(x_t, n, k)}) \right]^{(1-\zeta_{t,n})} \end{aligned} \quad (4.1)$$

The first term of this equation specifies the likelihood of the beam given it is a maximum range scan. In such a situation, we compute the likelihood as the product of the probabilities that the beam has covered the cells 0 to $z_{t,n}-1$. Please note, that the cell in which the beam ends does not provide any information, since we do not know, whether there is an object or not given the beam is a maximum range reading. Thereby the probability that a beam covers a cell $k < z_{t,n}$ is equal to $1 - m_{f(x_t, n, k)}$. The second row of the right hand side of this equation specifies how to deal with a non-maximum range reading. If $z_{t,n}$ is not reflected by a dynamic object, i.e. $c_{t,n} = 1$, the likelihood equals $m_{f(x_t, n, z_{t,n})}$. If, in contrast, $z_{t,n}$ is reflected by a dynamic object, the likelihood is $1 - m_{f(x_t, n, z_{t,n})}$. As well as for the maximum range measurements we have to consider in both cases that the beam has covered $z_{t,n} - 1$ cells before reaching cell $f(x_t, n, z_{t,n})$.

Based on the definition of the observation likelihood we will define the likelihood $p(z, c \mid x, m)$ of the data. We will try to maximize $p(z, c \mid x, m)$ in order to find the most likely map of the environment.

$$p(z, c \mid x, m) = \prod_{t=1}^T p(z_t, c_t \mid x_t, m) \quad (4.2)$$

$$= \prod_{t=1}^T p(z_t \mid x_t, m) \cdot p(c_t \mid x_t, m) \quad (4.3)$$

$$= \prod_{t=1}^T p(z_t \mid x_t, m) \cdot p(c_t) \quad (4.4)$$

$$= \prod_{t=1}^T \prod_{n=1}^N p(z_{t,n} | c_{t,n}, x_t, m) \cdot p(c_t) \quad (4.5)$$

We obtain Equation (4.3) from Equation (4.2) by assuming that the z_t and c_t are independent given x_t and m . We furthermore consider c_t as independent from the location x_t and the map m , which leads to Equation (4.4). Finally, Equation (4.5) is derived from Equation (4.4) under the usual assumption, that the beams of a single scan are independent given the map of the environment.

Maximizing $p(z, c | x, m)$ is equivalent to maximizing the corresponding log likelihood, which can be derived from Equation (4.5) and Equation (4.2) by straightforward mathematical transformations:

$$\begin{aligned} & \ln p(z, c | x, m) \\ &= \ln \prod_{t=1}^T \prod_{n=1}^N p(z_{t,n} | c_{t,n}, x_t, m) \cdot p(c_t) \\ &= N \cdot \sum_{t=1}^T \ln p(c_t) + \sum_{t=1}^T \sum_{n=1}^N \ln p(z_{t,n} | c_{t,n}, x_t, m) \\ &= N \cdot \sum_{t=1}^T \ln p(c_t) \\ &+ \sum_{t=1}^T \sum_{n=1}^N \left[(1 - \zeta_{t,n}) \cdot \left[c_{t,n} \cdot \ln m_{f(x_t, n, z_{t,n})} \right. \right. \\ &\quad \left. \left. + (1 - c_{t,n}) \cdot \ln(1 - m_{f(x_t, n, z_{t,n})}) \right] \right. \\ &\quad \left. + \sum_{k=0}^{z_{t,n}-1} \ln(1 - m_{f(x_t, n, k)}) \right] \end{aligned} \quad (4.6)$$

Since the correspondence variables $c_{t,n}$ are not observable in the first place, a common approach is to integrate over them. This is equal to optimize the expected log likelihood $E_c[\ln p(c, z | x, m) | x, m, d]$ instead. Since the expectation is a linear operator, we can move it inside the expression. By exploiting the fact that the expectation of $c_{t,n}$ only depends on the corresponding measurement $z_{t,n}$ and the pose x_t of the robot at that time we can derive the following equation:

$$\begin{aligned} E_c[\ln p(z, c | x, m) | z, x, m] &= \\ & \gamma + \sum_{t=1}^T \sum_{n=1}^N \left[e_{t,n} \cdot (1 - \zeta_{t,n}) \cdot \ln m_{f(x_t, n, z_{t,n})} \right. \\ &\quad \left. + (1 - e_{t,n}) \cdot (1 - \zeta_{t,n}) \cdot \ln(1 - m_{f(x_t, n, z_{t,n})}) \right] \end{aligned}$$

$$+ \left[\sum_{k=0}^{z_{t,n}-1} \ln(1 - m_{f(x,n,k)}) \right] \quad (4.7)$$

For the sake of brevity, we use the term

$$e_{t,n} = E_c[c_{t,n} | z_{t,n}, x_t, m] \quad (4.8)$$

in this equation. The term

$$\gamma = N \cdot \sum_{t=1}^T E_c[\ln p(c_t) | z, x, m] \quad (4.9)$$

is computed from the prior $p(c_t)$ of the measurements which is independent of z , x , and m . Accordingly, γ can be regarded as a constant.

Unfortunately, optimizing Equation (4.7) is not an easy endeavor. A typical approach to maximize log likelihoods is the EM algorithm. In the particular problem considered here this results in generating a sequence of maps m of increasing likelihood. In the E-Step, we compute the expectations about the hidden variables $c_{t,n}$. In the M-step we then compute the most likely map m using the expectations computed in the E-Step. Both steps are described in detail in the remainder of this section.

In the E-step we compute the expectations $e_{t,n} = E_c[c_{t,n} | z_{t,n}, x_t, m]$ for each $c_{t,n}$ given the measurement $z_{t,n}$, the location x_t of the robot and the current map m . Exploiting the fact that $e_{t,n}$ equals $p(c_{t,n} | z_{t,n}, x_t, m)$ and considering the two cases that $z_{t,n}$ is a maximum range reading or not, we obtain:

$$e_{t,n} = \begin{cases} p(c_{t,n}) & , \text{if } \zeta_{t,n} = 1 \\ p(c_{t,n})\epsilon_{t,n} & , \text{otherwise} \end{cases}$$

where

$$\epsilon_{t,n} = \frac{1}{p(c_{t,n}) + (1 - p(c_{t,n}))(\frac{1}{m_{f(x_t,n,z_{t,n})}} - 1)} \quad (4.10)$$

In the M-Step we want to determine the values for m and x that maximize Equation (4.7) after computing the expectations $e_{t,n}$ about the hidden variables $c_{t,n}$ in the E-step. Unfortunately, maximizing this equation is also not trivial since it involves a solution to a high-dimensional state estimation problem. To deal with the enormous complexity of the problem many researchers approximate it with an incremental maximum likelihood process [Gutmann and Konolige, 1999, Thrun, 2001]. The key idea of incremental approaches is to calculate the desired sequence

of poses and the corresponding maps by maximizing the marginal likelihood of the t -th pose and map relative to the $(t-1)$ -th pose and map. In the algorithm, we additionally consider the estimations $e_{t,n}$ that measurement n at time t is caused by a static object of the environment:

$$\hat{x}_t = \operatorname{argmax}_{x_t} \{p(z_t | c_t, x_t, \hat{m}(\hat{x}_{1:t-1}, z_{1:t-1})) \cdot p(x_t | u_{t-1}, \hat{x}_{t-1})\} \quad (4.11)$$

In this equation the term $p(z_t | c_t, x_t, \hat{m}(\hat{x}_{1:t-1}, z_{1:t-1}))$ is the likelihood of the measurement z_t given the pose \hat{x}_t and the map $\hat{m}(\hat{x}_{1:t-1}, z_{1:t-1})$ constructed so far. The term $p(x_t | u_{t-1}, \hat{x}_{t-1})$ represents the probability that the robot is at location x_t given it was previously at position \hat{x}_{t-1} and has carried out (or measured) the motion u_{t-1} . The registration procedure is then performed using the algorithm described in the previous Chapter 2.

It remains to describe how the measurement z_t is then used to generate a new map $\hat{m}(\hat{x}_{1:t}, z_{1:t})$ given the resulting pose \hat{x}_t and the expectations $e_{t,n}$. Fortunately, once x_1, \dots, x_t , have been computed, we can derive a closed-form solution for $m(x_{1:t}, z_{1:t})$. We want to determine the value of each field j of the map $m(x_{1:t}, z_{1:t})$ such that the overall likelihood of $m(x_{1:t}, z_{1:t})$ is maximized. To achieve this, we sum over the individual fields $j \in [1, \dots, J]$ of the map. Thereby we use an indicator function $I(y)$ which is 1, if y is true and 0, otherwise.

$$\begin{aligned} \hat{m}(\hat{x}_{1:t}, z_{1:t}) = \operatorname{argmax}_m & \left(\sum_{j=1}^J \sum_{t=1}^T \sum_{n=1}^N \left[I(f(x_t, n, z_{t,n}) = j) \right. \right. \\ & \cdot (1 - \zeta_{t,n}) \cdot (e_{t,n} \ln m_j + (1 - e_{t,n}) \ln(1 - m_j)) \\ & \left. \left. + \sum_{k=0}^{z_{t,n}-1} I(f(x_t, n, k) = j) \cdot \ln(1 - m_j) \right] \right) \end{aligned} \quad (4.12)$$

Now suppose, we define

$$\tilde{I}(x, n, k, j) := I(f(x, n, k) = j)$$

and

$$\begin{aligned} \alpha_j &:= \sum_{t=1}^T \sum_{n=1}^N \tilde{I}(x_t, n, z_{t,n}, j) \cdot (1 - \zeta_{t,n}) \cdot e_{t,n} \\ \beta_j &:= \sum_{t=1}^T \sum_{n=1}^N \left(\tilde{I}(x_t, n, z_{t,n}, j) \cdot (1 - \zeta_{t,n}) \right. \\ &\quad \cdot (1 - e_{t,n}) + \left. \sum_{k=0}^{z_{t,n}-1} I(f(x_t, n, k) = j) \right) \end{aligned}$$

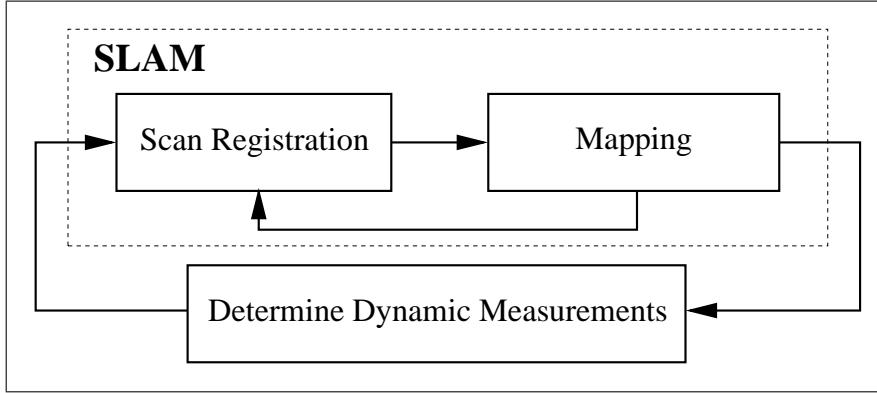


Figure 4.2: Iteration of SLAM and dynamic beam estimation.

The quantity α_j corresponds to the sum of the expectations $e_{t,n}$ of all beams that are reflected by static objects and end in cell j . Please note that we only consider the beams that are not maximum-range beams ($\zeta_{t,n} = 0$). The term β_j , on the other hand, is the sum of two terms. The first term is the sum of the expectations $1 - e_{t,n}$ that beam n of scan t is reflected by a dynamic object of all beams that end in cell j and that are not maximum-range beams. The second value of the sum simply is the number of times a beam covers j but does not end in j . Please note that this value is independent from whether or not the corresponding beam is reflected by a dynamic object or not. Please furthermore note that in a static world with $e_{t,n} = 1$ for all t and n the term α_t corresponds to the number of times a beam that does not have the maximum length ends in j . In contrast to that, β_j is the number of times a beam covers a cell.

Using the definitions of α_j and β_j , Equation (4.12) turns into

$$m(x_{1:t}, z_{1:t}) = \operatorname{argmax}_m \left(\sum_{j=1}^J \alpha_j \ln m_j + \beta_j \ln(1 - m_j) \right) \quad (4.13)$$

Since all m_j are independent, we maximize the overall sum by maximizing each m_j . A necessary condition to ensure that m_j is a maximum is that the first derivative equals zero:

$$\frac{\partial m}{\partial m_j} = \frac{\alpha_j}{m_j} - \frac{\beta_j}{1 - m_j} = 0 \quad (4.14)$$

By straightforward mathematical transformations we obtain

$$m_j = \frac{\alpha_j}{\alpha_j + \beta_j}. \quad (4.15)$$



Figure 4.3: Robot Sam mapping the populated exhibition hall of the Byzantine Museum in Athens (left). In the resulting map (right), the measurements labeled as dynamic are shown in orange / grey.

Please note that, given the sensor model specified in Equation (4.2), this closed-form solution for the most likely map m for given poses x and a static environment corresponds to the naive counting technique in which one counts for each cell how often a beam has ended in that cell and how often a beam has covered it without ending in it. We have previously seen in Section 2.2.3 maps built by using this technique.

The overall approach can be summarized as follows (see also Figure 4.2). We start with an initial map \hat{m} obtained by the incremental mapping approach. Thereby the expectations $e_{t,n}$ are initialized with the prior probability $p(c_{t,n})$ that a measurement is caused by a static object. Given the resulting map \hat{m} and the corresponding poses \hat{x} , we compute new expectations $e_{t,n}$ for each beam according to Equation (4.8). These expectations are then used to compute a new map. The overall process is iterated until no improvement of the overall likelihood (Equation (4.6)) can be achieved or a certain number of iterations has been exceeded.

Finally, we would like to discuss how to deal with beams that are not parallel to the x-axis. In this case we no longer can compute the likelihood that a beam covers a cell j of m as $(1 - m_j)$. Otherwise, transversal beams covering more cells would accumulate a lower likelihood. The solution to this is to weigh the beams according to the length by which they cover a cell. Suppose B is the set of cells in m covered by a beam. Furthermore suppose x_j is the length by which the beam covers field $j \in B$. Then, the likelihood of a covering all cells in B is computed as $\prod_{j \in B} (1 - m_j)^{x_j}$.

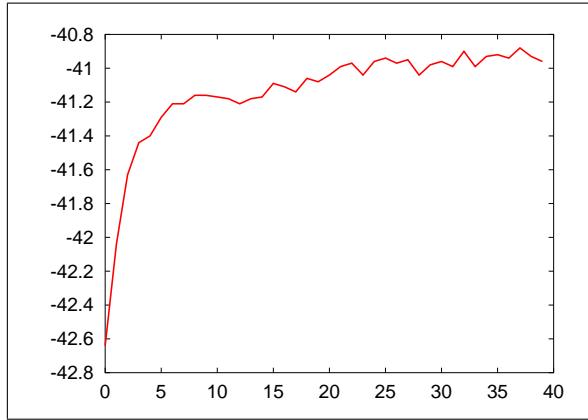


Figure 4.4: Evolution of the log likelihood (Equation (4.6)) during the individual iterations.

4.3 Experiments

This approach has been implemented and tested on different robotic platforms, in different environments and with 2D and 3D data. In all experiments, we figured out, that the system is robust even in highly dynamic environments. In one experiment carried out with a fast moving car, the system was able to accurately map the environment even if no odometry data was given.

4.3.1 Filtering People

For the first experiments we used the dataset from the experiment described in Section 3.4. The experiment was carried out using the Pioneer 2 robot Sam in the empty exhibition hall of the Byzantine Museum in Athens, Greece. Figure 4.3 (left) shows the robot during the mapping process. There were 15 people walking with a typical speed through the environment while the robot was building a map. Partially they stopped and later on continued moving. The most likely map computed by our approach is shown in the right image of Figure 4.3. The observations labeled as dynamic are drawn orange in this figure. As can be seen, this approach can reliably identify dynamic objects and is able to learn maps that include the static ones only. At this point we would also like to mention that the resulting map contains seriously less dynamic objects than the map obtained with the approach presented in the previous chapter.

Figure 4.4 plots the evolution of $E_c[\ln p(c, z | x, m) | x, m, d]$ over the different iterations of the algorithm. It illustrates that this algorithm in fact maximizes the overall log likelihood. Please note, that this curve generally is not monotonic, because of the incremental maximum-likelihood solution to the high dimensional

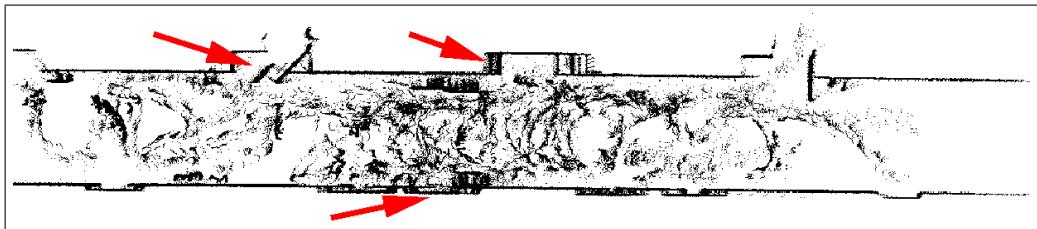


Figure 4.5: Map obtained in a populated corridor of the Wean Hall at Carnegie Mellon University using the raw input data.



Figure 4.6: Map generated by the algorithm. The filtered out measurements are plotted in orange / grey.

map building problem. Slight variations in the pose can have negative effects in future steps, so that the map likelihood can decrease. However, we never observed a significant decrease of the log likelihood.

4.3.2 Improved Localization Accuracy

Besides the fact that the resulting maps contain less spurious objects, this approach also increases the localization accuracy. If dynamic objects are not handled appropriately during localization, matching errors become more likely. Figure 4.5 shows a typical map we obtained when mapping a densely populated environment. In this case we mapped a part of the Wean Hall Corridor at Carnegie Mellon University during peak office hours when many persons were around. Some of them were trying to block the robot, so that the robot had to make detours around them. Therefore the robot traveled 74m with an average speed of 0.15m/s (0.35m/s maximum). Despite the fact, that the huge amount of spurious objects make the map useless for navigation tasks, it also shows serious errors in the alignment. Some of the errors are indicated by arrows in the corresponding figure.

Figure 4.6 shows the map generated by the algorithm. As the figure illustrates, the spurious measurements (indicated by orange dots) have been filtered out completely. Additionally, the alignment of the scans is more accurate.

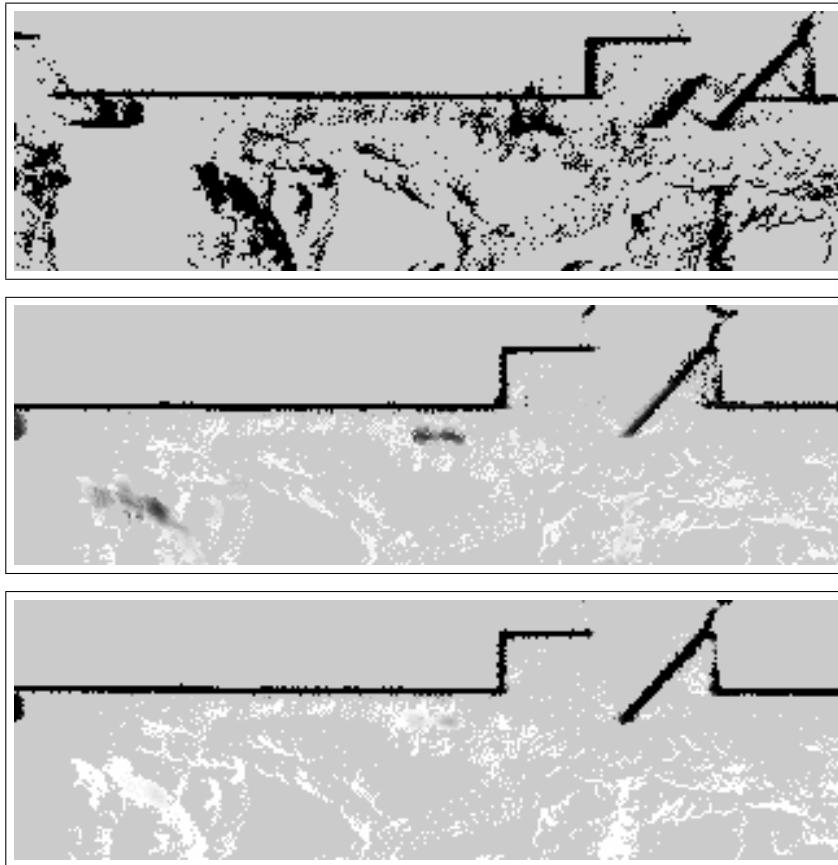


Figure 4.7: Evolution of the map during EM. The images corresponds to iteration 1, 2, and 6.

Figure 4.7 depicts the evolution of a part of the map in the different rounds of the EM. It shows how the beams corresponding to dynamic objects slowly fade out and how the improved estimates about these beams improve the localization accuracy.

4.3.3 Generating Large-Scale Outdoor Maps

To evaluate the capability of this technique to deal with arbitrary features, we mounted a laser-range scanner on a car and drove approximately 1km through Pittsburgh, PA, USA (Corner between Craig Street and Forbes Avenue). The maximum speed of the car was 35 MPH in this experiment. We then applied our approach to the recorded data. The map generated by the algorithm is shown in Figure 4.8. Whereas the black dots correspond to the static objects in the scene,

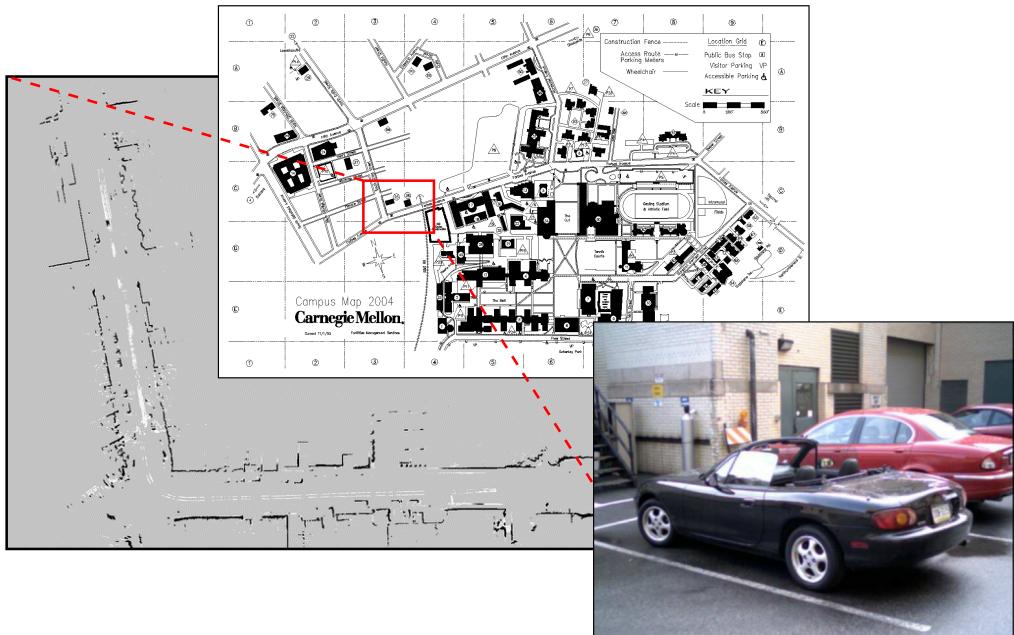


Figure 4.8: Map of an outdoor scene after filtering dynamic objects. The upper image shows the corresponding street map. The area covered by the grid map is marked with a red / grey box. The right image shows the car which was used for the experiment.

the white dots are those which are filtered out using this technique. Again, most of the dynamics of the scene could be removed. Only a few cars could not be identified as dynamic objects. This is mainly because we quickly passed cars waiting for turns and because we drove along the road only once. Please also note, that due to the lack of a GPS, the map had to be computed without any odometry information using only the laser measurements.

4.3.4 Generating Textured 3D Maps

To demonstrate that this approach is not limited to 2D range data, we carried out several experiments with the mobile robot Robin (see upper left image in Figure 4.9) which is equipped with a laser-scanner mounted on an AMTEC pan/tilt unit. On top of this scanner we installed a camera which allows us to obtain textured 3D maps of an environment. Additionally, this robot provides a horizontally scanning laser range finder which we used in the experiments to determine dynamic objects. To label the beams in the 3D data as dynamic we use a bounding box around the dynamic 2D points. To filter dynamic objects in the textures recorded with Robin's cameras we choose for every polygon that image which has

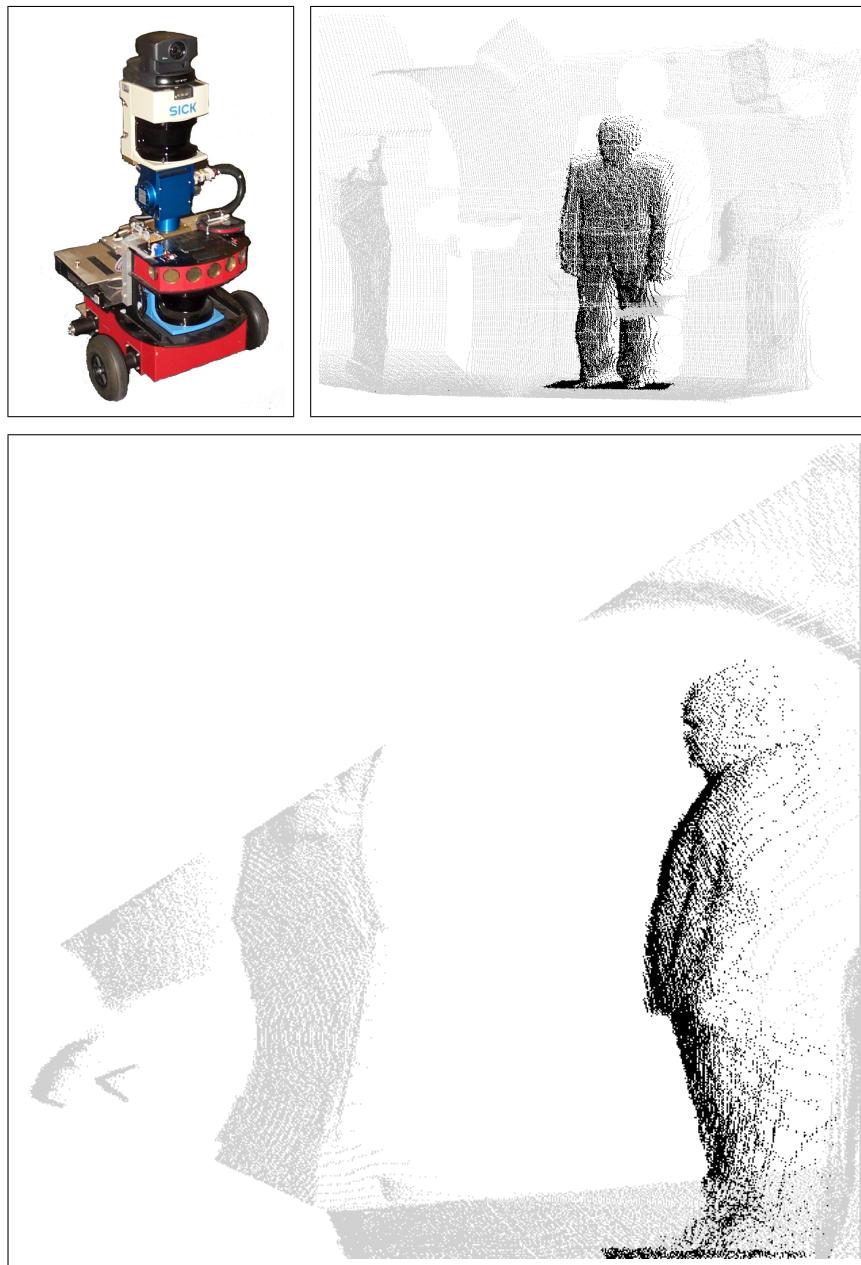


Figure 4.9: The mobile robot Robin used to generate textured 3D models (see also Figure 4.10). Beams reflected by a person are isolated from the rest of the data. This is achieved by computing a bounding box around those beams perceived with the horizontal scanner that are identified as corresponding to dynamic objects (upper right and lower image).

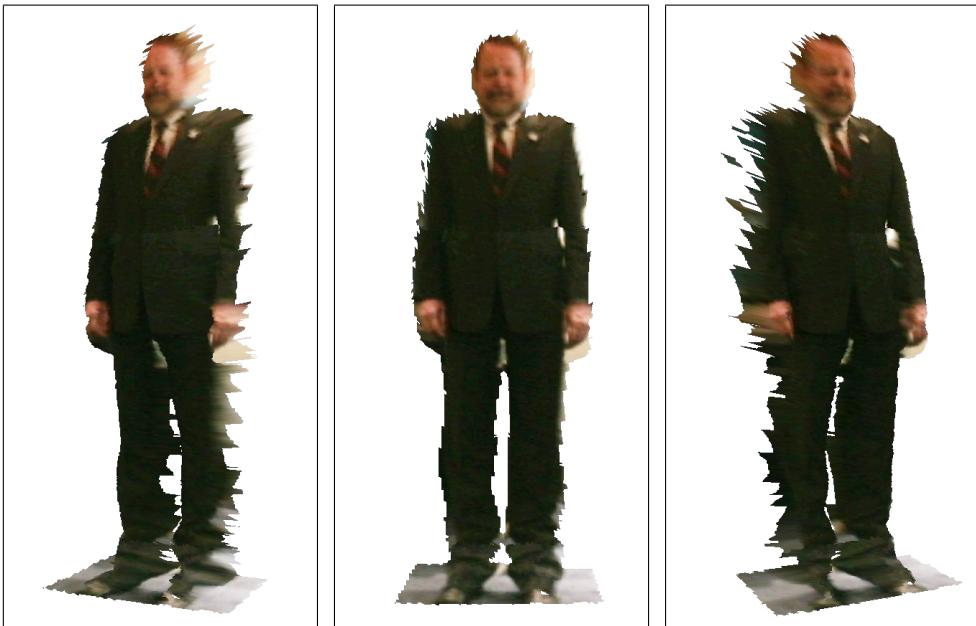


Figure 4.10: Textured 3D model of a person identified as a dynamic object.

the highest likelihood of containing static aspects only. The upper image of Figure 4.11 shows one particular view of a model obtained without filtering dynamic objects. The arrow indicates a polygon whose texture contains fractions of an image of a person which walked through the scene while the robot was scanning it. After applying the approach the corresponding beams and parts of the pictures were filtered out. The resulting model shown in the lower image of Figure 4.11 therefore only contains textures showing static objects.

4.3.5 Extracting Textured 3D Objects

Additionally to filtering dynamic objects and learning static aspects of environments this algorithm can also be used to separate dynamic objects from the environment. The key idea is to extract all measurements from the 3D data that lie within a bounding box around the beams whose probability that they are reflected by dynamic objects exceeds a threshold value which was set to 0.7 in our experiments. Figure 4.9 shows two views of a typical 3D data sets obtained with this approach. Whereas the data points belonging to a dynamic object are shown in black, the rest of the data is depicted in grey. Again we used the camera to map textures on the 3D data that were identified as belonging to a dynamic object. Figure 4.10 depicts three views of the resulting model. As can be seen from the figure, the approach can accurately extract realistic looking textured 3D models of dynamic objects.

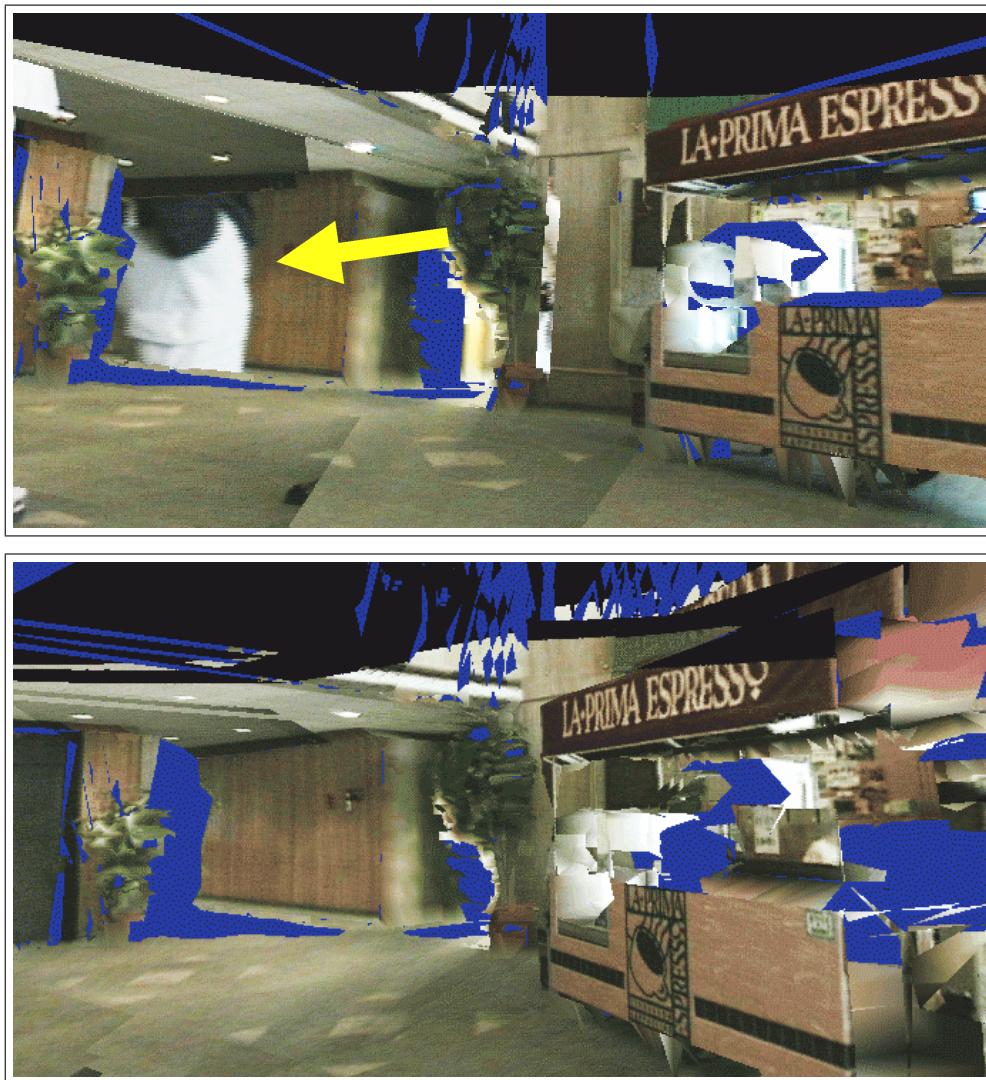


Figure 4.11: Textured 3D models obtained using Robin. The upper image shows the result without filtering. The lower image shows the resulting model obtained with our algorithm.

4.4 Related Work

For mobile robots, there exist several techniques to mapping in dynamic environments. The approaches mostly relevant to the work reported here are the methods developed by Wang *et al.* [Wang and Thorpe, 2002, Wang *et al.*, 2003] and the work described in the previous Chapter 3. Wang *et al.* use a heuristic and feature-based approach to identify dynamic objects in range scans. The corresponding

measurements are then filtered out during 2D scan registration. In Chapter 3 we described an approach to track persons in range scans and to remove the corresponding data during the registration and mapping process. Compared to these techniques, the algorithm presented in this chapter does not rely on any pre-defined features. Rather, it considers every measurement individually and estimates a posterior about whether or not this data item has been generated by a dynamic object.

Additionally, there has been work on updating maps or improving localization in populated environments. For example, in the system described in [Burgard *et al.*, 2000] we update a given static map using the most recent sensory input to deal with people in the environment during path planning. Montemerlo *et al.* [Montemerlo and Thrun, 2002] present an approach to simultaneous localization and people tracking. Siegwart *et al.* [Siegwart, R., 2002] present a team of tour-guide robots that operates in a populated exhibition. Their system uses line features for localization and has been reported to successfully filter range-measurements reflected by persons. Fox *et al.* [Fox *et al.*, 1999b] present a probabilistic technique to identify range measurements that do not correspond to the given model of the environment. These approaches, however, require a given and fixed map which is used for localization and for the extraction of the features corresponding to the people. The technique presented in this chapter, in contrast, does not require a given map. Rather it learns the map from scratch using the data acquired with the robot's sensors. The algorithm repeatedly interleaves the process of estimating which beams are caused by dynamic objects with a mapping and localization algorithm. Thereby it iteratively improves its estimates and generates more accurate models of the environment.

From a more general perspective, the problem of estimating dynamic elements in data can be regarded as an outlier detection problem, since the spurious measurements are data items that do not correspond to the static aspects of the environment which have to be estimated. The identification of outliers is an important subtask in various application areas such as data mining [John, 1995, Brodley and Friedl, 1996, Ramaswamy *et al.*, 2000], correspondence establishment [Cox and Hingorani, 1996, Besl and McKay, 1992], clustering [Duda *et al.*, 2001], or statistics [Barnett and Lewis, 1994]. In all these fields, errors in the data reduce the accuracy of the resulting models and thus can lead to a decreased performance whenever the learned models are used for prediction or robot navigation, for example. The problem considered in this chapter differs from these approaches in the fact that outliers cannot be detected solely based on their distance to the other data items. Rather, the measurements first have to be interpreted and transformed into a global representation (map) before individual measurements can be identified as outliers.

4.5 Summary

In this chapter we presented a probabilistic approach to mapping in dynamic environments. This approach uses the EM algorithm to interleave the identification of measurements that correspond to dynamic objects with a mapping and localization algorithm. This way it incrementally improves its estimate about spurious measurements and the quality of the map. The finally obtained maps contain less spurious objects and are also more accurate because of the improved range registration.

The technique has been implemented and tested on different platforms. In several experiments carried out in indoor and outdoor environments we demonstrated that this approach yields accurate maps even if used on a fast moving vehicle without odometry information. We also presented an application to learn textured 3D models of dynamic environments. Finally, we applied the algorithm to extract dynamic objects from 3D data. The results illustrate that the approach can reliably estimate which beams correspond to dynamic objects.

CHAPTER 5

Efficient Mapping with Rao-Blackwellized Particle Filters

5.1 Introduction

In the literature, the mobile robot mapping problem is often referred to as the *simultaneous localization and mapping problem (SLAM)* [Castellanos *et al.*, 1999, Dissanayake *et al.*, 2000, Leonard and Feder, 1999]. This is because mapping includes both, estimating the position of the robot relative to the map and generating a map using the sensory input and the estimates about the robot's pose.

One of the hardest problems in SLAM is that of closing a loop. As a robot traverses a large cycle in the environment, it faces the hard data association problem of correctly adding new observations to its own map under large position errors. This problem has long been acknowledged for its hardness, and a number of approaches have addressed it [Bosse *et al.*, 2003, Gutmann and Konolige, 1999, Thrun *et al.*, 2000]. Recently, Murphy and colleagues have presented Rao-Blackwellized particle filters [Doucet *et al.*, 2000, Murphy, 1999] as an effective way of representing alternative hypotheses on robot paths and associated maps. Montemerlo *et al.* [Montemerlo *et al.*, 2002, Montemerlo and Thrun, 2003] extended this idea to efficient landmark-based SLAM using Gaussian representations.

In this chapter we present a highly efficient approach to simultaneous localization and mapping with laser scans. As previously proposed by Murhpy [Murphy, 1999], our approach applies a Rao-Blackwellized particle filter to estimate a posterior of the path of the robot, in which each particle maintain an own entire map. This differs from work in [Thrun *et al.*, 2000], where only a single

map is retained. To scale to larger environments, we transform sequences of laser range-scans into odometry measurements using the range-scan registration technique described in Chapter 2. This way our system can deal with significantly larger environments than the original approach [Murphy, 1999], since the scan matching yields odometry estimates that are an order of magnitude more accurate than the raw wheel encoder data. Simultaneously, the transformation of sequences of scans into odometry measurements reduces the well-known particle depletion problem [van der Merwe *et al.*, 2000], since the number of resampling operations is significantly reduced. By using a learned model of the residual errors of the range registration our approach can correctly integrate the corrected odometry into the particle filtering process. As a result, we obtain a drastic reduction in the number of particles needed to build large-scale maps, or an improved ability to map large environments. This is demonstrated in our experimental results section, in which we compare our approach to previous techniques.

This chapter is organized as follows. In the following section, we will briefly discuss the Rao-Blackwellized particle filter techniques for probabilistic mapping and localization. In Section 5.3, we describe our approach to integrate scan matching with Rao-Blackwellized particle filters to achieve a robust approach for simultaneous mapping and localization. Section 5.4 presents several experiments illustrating that our approach can successfully learn accurate maps with range scanners in large-scale environments. Additionally, we present experiments illustrating that our technique outperforms existing approaches. Finally we will discuss the related work in Section 5.5.

5.2 Mapping with Rao-Blackwellized Particle Filter

In probabilistic terms the goal of map learning is to find the map and the robot positions which yield the best interpretation of the data d_t gathered by the robot [Thrun, 2001]. Here the data $d_t = \{u_{0:t-1}, z_{1:t}\}$ consists of a stream of odometry measurements $u_{0:t-1}$ and perceptions of the environment $z_{1:t}$. The mapping problem can be phrased as recursive Bayesian filtering for estimating the robot positions along with a map of the environment:

$$\begin{aligned} & p(x_{1:t}, m \mid z_{1:t}, u_{0:t-1}) \\ &= \alpha \cdot p(z_t \mid x_t, m) \cdot \\ & \quad \int p(x_t \mid x_{t-1}, u_{t-1}) p(x_{1:t-1}, m \mid z_{1:t-1}, u_{0:t-2}) dx_{1:t-1} \end{aligned} \quad (5.1)$$

In probabilistic mapping and localization it is typically assumed that the odometry measurements are governed by a so-called probabilistic motion model $p(x_t \mid$

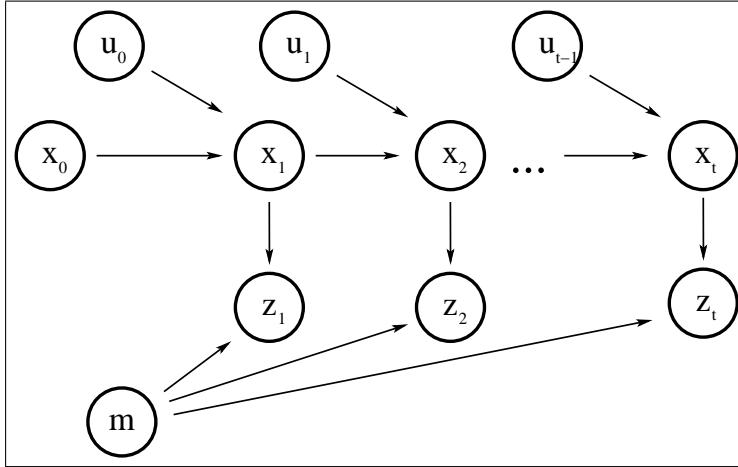


Figure 5.1: Graphical model of concurrent mapping and localization as filtering process.

$x_{t-1}, u_{t-1})$ which specifies the likelihood that the robot is at x_t given that it previously was at x_{t-1} and the motion u_{t-1} was measured. On the other hand, the observations follow the so-called observation model $p(z | x)$, which defines for every possible location x in the environment the likelihood of the observation z .

More recently, Murphy and colleagues [Murphy, 1999, Doucet *et al.*, 2000] have presented Rao-Blackwellized particle filters as an efficient way to represent the full posterior of the robot pose. Figure 5.1 depicts a graphical model of Rao-Blackwellized simultaneous mapping and localization. The key idea of this approach is to solve the recursive Bayes filter update by the following equation:

$$\begin{aligned} p(x_{1:t}, m | z_{1:t}, u_{0:t-1}) = \\ p(m | x_{1:t}, z_{1:t}, u_{0:t-1})p(x_{1:t} | z_{1:t}, u_{0:t-1}) \end{aligned} \quad (5.2)$$

Here, a particle filter is used to represent robot trajectories $x_{1:t}$ and an individual map is conditioned on each sample of the particle filter. The importance weights of the samples are computed according to the likelihood of the observation in the maximum likelihood map constructed using exactly the path this particular particle has taken. The key advantage of this approach is that the samples approximate at each point in time the full posterior over robot poses and maps.

However, particle filters are known to be subject to major approximation errors. One of these errors is known as the particle depletion problem [van der Merwe *et al.*, 2000]. This problem can lead to a divergence of the filter and can result in the lack of particles in the vicinity of the correct state. In the SLAM context this can prevent the robot from closing a given loop. There are two parameters that have a major influence on the approximation error. First, the number of particles

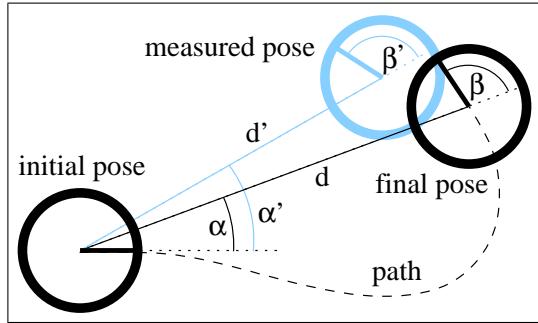


Figure 5.2: Parameters of the probabilistic motion model.

needs to be high enough to represent the posterior. However, too many samples can prevent the filtering process from being fast enough for online-processing. Furthermore, the number of resampling steps needs to be limited in order to avoid that the samples converge too quickly to the maximum likelihood state which is undesirable especially in ambiguous situations. On the other side, too few resampling steps could result in a divergence of the filter since many samples are wasted on unlikely states and the uncertainty typically introduced by robot motions would exceed the certainty gained by incorporating observations of the environment.

In the following section we will describe our solution to this problem. This approach transforms sequences of laser measurements into odometry measurements using a scan matching procedure and utilizes the remaining laser scans for map estimation.

5.3 Combination with Scan Matching

Our approach to deal with the previously described problems is to use a scan matching routine to correct the odometry and to use this corrected path information as input for the sampling step in the Rao-Blackwellized particle filter.

The 2D scan matching we apply, which is described in detail in Chapter 2, aligns a scan relative to the previous scans by computing an occupancy grid map [Moravec, 1988] from the previous measurements. To avoid the time consuming ray-tracing operation required to compute the likelihood of a measurement $p(z | x)$ we apply an approximation which considers only the endpoint of a beam (see Section 2.2.4). This way, $p(z | x)$ can be computed efficiently using fast look-up operations. To be able to incorporate also maximum range measurements, our system assumes that the cell 20cm in front of the one in which the maximum range measurement ends must be unoccupied.

A key question when combining a scan matching routine with a probabilis-

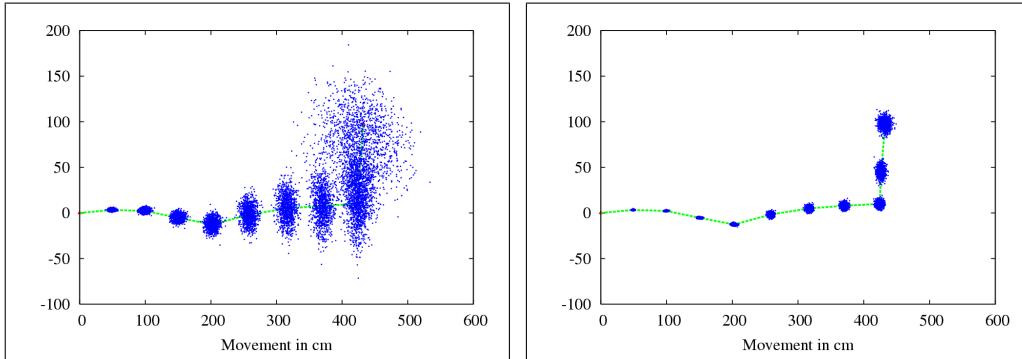


Figure 5.3: Sample densities obtained with the models for the raw odometry (left image) and for scan matching (right image) for ten incremental movements of a real robot.

tic technique is how to estimate the uncertainty of the scan matching process to correctly incorporate this uncertainty during the prediction step of the sampling procedure. In our current system we use a parametric model of the odometry error and learn the parameters of this model using data acquired during experiments. To learn the parameters of the model we performed an experiment in which we generated a statistics of alignment errors after convergence of scan matching. Using a data set recorded in the Intel Research Lab Seattle, we applied our system executed with a manually designed motion model. We then took the resulting map (see Figure 5.6 (right)) as ground truth and compared the raw odometry and the results of the scan matching with the positions corrected by the routine. The error model we use has three parameters as it assumes that in every single movement there are three errors involved (see also Figure 5.2). First, whenever the robot starts to move, it makes a small rotational error $\alpha' - \alpha$. Second, the robot introduces a certain error $d' - d$ to the distance between the final location and the starting position. Finally, the true final orientation differs by a certain amount from the measured orientation which is expressed by a non-zero difference $\beta - \beta'$. The means and the variances of the relative errors in these three parameters were learned by comparing the approximated displacement after convergence of the scan matching routine with the (estimated) ground truth information. Alternative models of odometry errors and corresponding techniques for parameter estimation have been proposed by Borenstein and Feng [Borenstein and Feng, 1996], Doh *et al.* [Doh *et al.*, 2003], as well as Bengtsson and Baerveldt [Bengtsson and Baerveldt, 2001].

Figure 5.3 plots the resulting sample densities obtained when relying on pure odometry (left image) and the densities obtain with the error model for the scan matching process (right image). As the figure shows, the samples are much more

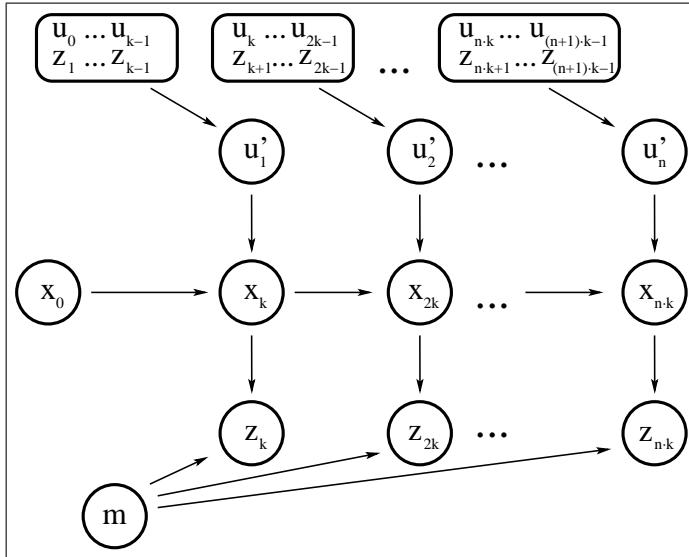


Figure 5.4: Graphical model of the integration of scan matching and probabilistic mapping.

focused if the scan matching routine is used. This leads to the desired effect that the variance of the posterior is reduced, that fewer samples can be used, and that larger loops can be closed.

A graphical model of our approach to integrate results from the scan matching process into the Rao-Blackwellized sampling routine is depicted in Figure 5.4 (compare with Figure 5.1). The key idea is to compute every k steps a new odometry measurement u'_j out of the $k - 1$ previous observations z and the k most recent odometry readings. The k -th observation is then used to compute the weights of the samples in the particle filter. Note that this clear separation between laser scans used for odometry and laser scans used for map estimation ensures that all information is used only once.

One important aspect when using Rao-Blackwellized particle filters for mapping is the efficient update of the maps of the individual particles. Montemerlo *et al.* [Montemerlo *et al.*, 2002] as well as Eliazar and Parr [Eliazar and Parr, 2003] proposed a tree-structure to efficiently update the map. In the system described here, we only use a limited number of scans gathered by the robot to update the map of a particle. We use a fixed number of scans in the history which intersect with the area visible according to the pose of the corresponding particle. This way, the update of the map of every particle can be achieved in constant time. Although this is an approximation only, we never found any evidence that the quality of the resulting maps was decreased significantly.

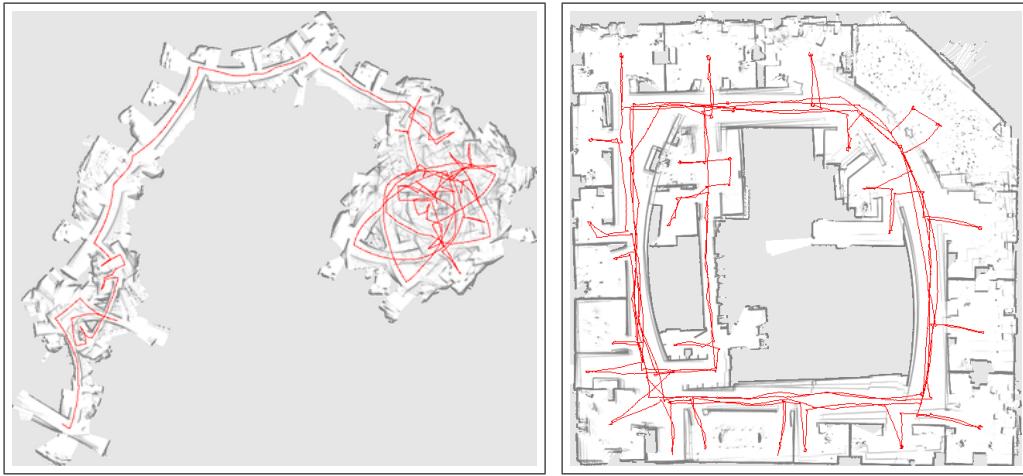


Figure 5.5: Mapping of the Intel Research Lab with the raw odometry data (left) and after scan matching (right).

5.4 Experimental Results

The approach described above has been implemented and tested using different robotic platforms and in different environments as well as in extensive simulation runs. In all experiments, we found out that the system can operate online and can also robustly close large and also nested loops.

The first experiment was carried out using a Pioneer 2 robot equipped with a SICK LMS laser range-finder in the Intel Research Lab, Seattle, WA. The size of this environment is $28m \times 28m$. The robot traveled 491m with an average speed of 0.19m/s. Figure 5.5 (left) shows the map generated based on the raw odometry data provided by the robot. As can be seen from the figure, the robot suffers from serious errors in odometry so that the resulting map is useless without any correction. Figure 5.5 (right) shows the map created with our scan matching technique. Although local structures of the map appear to be very accurate, the map is globally inconsistent. For example many structures like walls, doors etc. can be found twice and with a small offset between them. Finally, the left image of Figure 5.6 shows the resulting map obtained with our system. Although the sharpness of this map is not as high as that of the map created only with scan matching, they are globally consistent. The map was created in real-time, i.e. the computation time needed to process the data did not exceed the time to record them. We used 100 samples, a number we found to yield satisfactory results on almost all data sets. Figure 5.6 (right) shows a map created using 500 particles. Whereas this map is more accurate and has a similar crispness as the scan matching map, the time to compute this map was several hours.

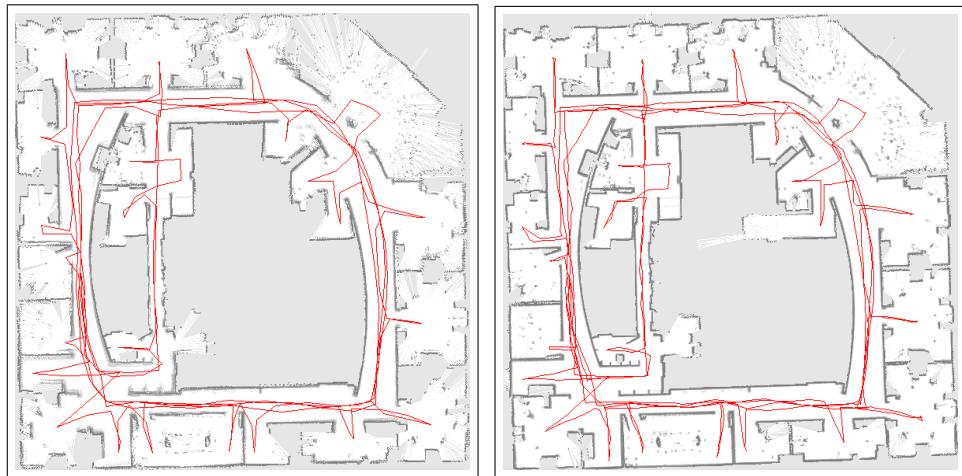


Figure 5.6: Map of the Intel Research Lab created with 100 samples (left) and offline with 500 samples (right).

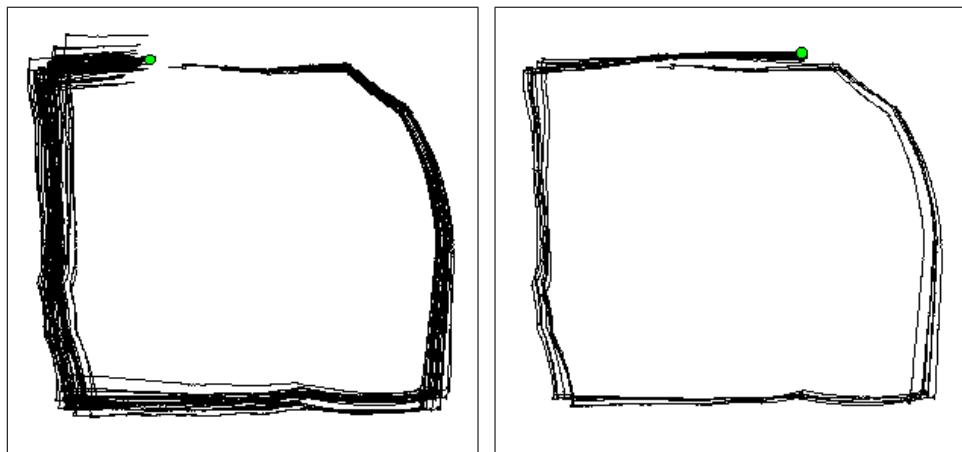


Figure 5.7: Trajectories of all 100 samples shortly before (left) and after (right) closing the loop.

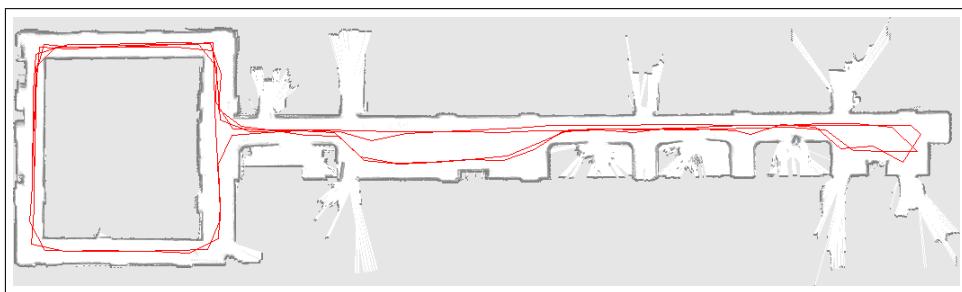


Figure 5.8: Map of the Sieg Hall at the University of Washington created in real-time.

Figure 5.7 visualizes the trajectories of all samples shortly before and after closing the major loop in this data set. As the left image illustrates, the robot is quite uncertain about its position relative to the starting position upon its return. However, after a few resampling steps the uncertainty has been reduced drastically (right image).

5.4.1 Mapping Large-Scale Environments with Multiple Cycles

A second example map obtained with our approach is depicted in Figure 5.8. The map shows the fourth floor of the $50\text{m} \times 12\text{m}$ large Sieg Hall of the University of Washington, WA, USA. As can be seen from the figure, the robot went several times around the circle and still successfully learned a consistent map. This map was generated in real-time using 100 samples. The grid resolution was 10cm. Figure 5.9 shows the last example map obtained with our approach. The size of this environment is $232\text{m} \times 198\text{m}$. The robot traveled approximately 2km with an average speed of 0.6m/s. The map was created offline using 500 samples.

5.4.2 Comparison to Previous Online Techniques

The second experiment is designed to show the advantage of our integrated technique over previous approaches that represent a posterior over poses in a single map only [Thrun, 2001, Gutmann and Konolige, 1999]. For this experiment we used a data set generated for the Wean Hall of the Carnegie Mellon University using our B21r simulator. The size of this environment is $32\text{m} \times 10\text{m}$. In the simulation the robot moved 251m with an average speed of 0.78m/s. To obtain realistic data, we added a serious amount of noise to the ground truth data provided by the simulation system. The resulting input trajectory is depicted in Figure 5.10 (top). Please note, that pure scan matching again failed to correctly close the loop using this data set. We implemented the particle filter strategy presented by Thrun *et al.* [Thrun *et al.*, 2000, Thrun, 2001]. In this approach the particle filter is used to estimate the correct pose relative to the previously seen area. The robot localizes itself in the current map and correct the path if an assignment to a previously build part of the map is found. In our system we achieve this by using only that sample with the highest likelihood during the resampling process whenever the robot closes a loop. After this, we continue with the normal resampling procedure described above. The middle image in Figure 5.10 shows the result of this procedure. The point in time when the system discovered that it closed the loop and the resulting inconsistencies are labeled. The inconsistencies are a consequence of the fact that only the particle with the highest importance factor survives at the time



Figure 5.9: The image shows the map of the $232\text{m} \times 198\text{m}$ large campus area of the Faculty of Applied Science, University of Freiburg.

the robot closes the loop. Since this particle does not always correspond to the correct position of the robot (as it is the case in this experiment) and since the motion model cannot compensate for this error, the resulting map is inconsistent. In contrast to that, our approach integrates scan matching with a Rao-Blackwellized particle filter and builds a consistent map of the environment (see bottom image of Figure 5.10) since it provides accurate predictions and simultaneously maintains the robot pose uncertainty in the posterior.

Finally, we analyzed whether the standard Rao-Blackwellized particle filter without odometry correction by scan matching provides the same performance as our approach. For this purpose we run the standard procedure using the input data for the Intel Research Lab. We used shorter resampling steps (three times more often than in the other runs to avoid a fast divergence) and 200 samples which was the maximum number of samples that allowed updates in real-time on our 1.8GHz Pentium IV PC. Since the standard procedure was not able to learn a consistent map, we repeated this experiment with increasing numbers of samples. It turned out that under 1000 samples, which was the maximum number our PC equipped with 768MB of main memory could handle, we could not observe a case in which the standard algorithm converged. An example map typically obtained using the standard algorithm is depicted in Figure 5.11. The fact that our algorithm reliably

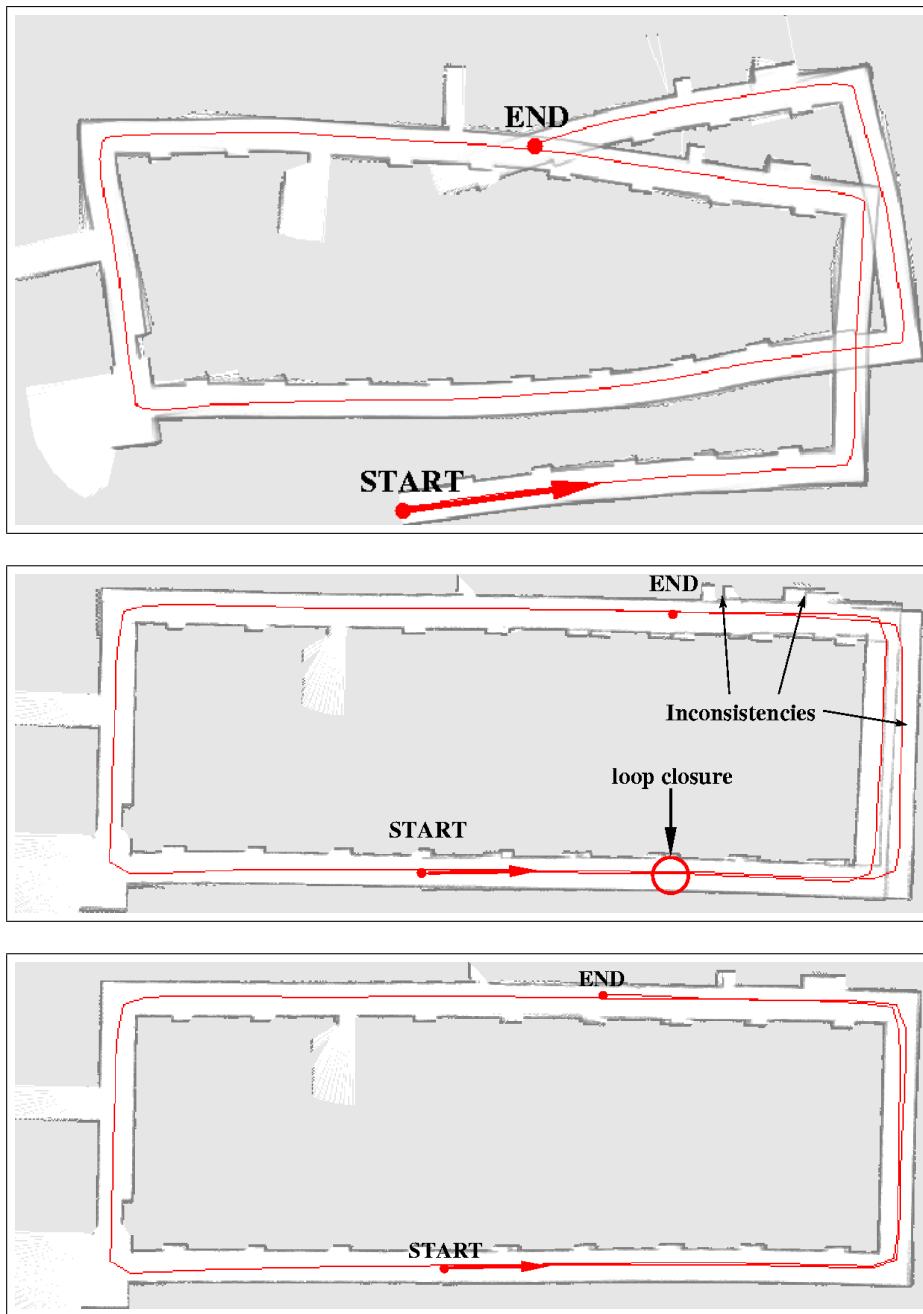


Figure 5.10: The top image shows the map obtained from the raw input data in the simulation experiment. In the middle, the resulting map is shown if the mapping process is continued with the maximum likelihood sample after closing the loop at the marked place. The inconsistencies in the right part of the map show that the loop is not correctly closed. The bottom map was built using our approach.



Figure 5.11: Map created with the standard Rao-Blackwellized particle filtering technique in real-time using 100 samples and based on the raw odometry data.

converges with 100 samples indicates that the integration of the scan matching routine yields an enormous improvement.

5.5 Related Work

Mapping is largely considered the most difficult perceptual problem in robotics. In probabilistic terms the goal of map learning is to find the map and the robot positions which yield the best interpretation of the data. Unfortunately, estimating the full posterior in Equation 5.1 is not tractable in general. One popular approach is to restrict observations to landmark detections, and represents robot positions by Gaussians [Dissanayake *et al.*, 2000]. In this context, the Bayes filter can be approximated efficiently by an EKF [Smith *et al.*, 1990, Leonard and Durrant-Whyte, 1991] for which the state consists of the robot positions along with positions of the landmarks.

Other researchers attempted to overcome the restrictions to landmark observations by using laser range-finders and incremental scan matching [Besl and McKay, 1992, Weiß *et al.*, 1994, Röfer, 2002] (see also Section 2.3.2). Whereas this approach has the advantage that it yields accurate results and can be implemented efficiently if the registration is performed with respect to a global map or with respect to a fixed number of scans only. Its major disadvantage lies in the

greedy maximization step. When the robot has to close larger loops, this approach suffers from registration errors during loop closures and therefore tends to fail in large environments. To overcome this problem, extensions of this approach have been developed which maintain a posterior about the position of the vehicle [Gutmann and Konolige, 1999, Thrun, 2001]. The key idea of these techniques is to delay the maximization until the robot detects that a loop has been closed. This is usually done by identifying that the robot enters an already known area from an unknown area and simultaneously observes a high likelihood of its observations for potential positions that are under consideration. If the registration of the vehicle in its map can be done with high likelihood, the previous poses are corrected backwards in time according to the pose correction that is necessary to properly close the loop. The posterior of the robot's poses is then replaced by a Dirac distribution which has its mode at the most likely position when the robot closes the loop. Furthermore, subsequent backwards corrections are stopped when the robot reaches this node. Whereas this approach can reliably close even large loops, it has the disadvantage that it under-estimates the uncertainty in the robot pose when closing loops.

As mentioned before, Murphy *et al.* have presented Rao-Blackwellized particle filters [Murphy, 1999] as an effective way of representing alternative hypotheses on robot paths and associated maps. We showed in the last section, that this work suffers from the number of particles needed to build a consistent map. Montemerlo *et al.* [Montemerlo *et al.*, 2002] extended this idea to efficient landmark-based SLAM using Gaussian representations to and successfully realized it on real robots. The drawback of this technique is the dependency on landmarks.

5.6 Summary

In this chapter we presented a highly efficient algorithm for simultaneous mapping and localization using laser scans that combines a scan matching procedure with Rao-Blackwellized particle filtering. The scan matching routine is used to transform sequences of laser measurements into odometry measurements. The corrected odometry and the remaining laser scans are then used for map estimation in the particle filter. The lower variance in the corrected odometry reduces the number of necessary resampling steps and this way decreases the particle depletion problem. In practical experiments we demonstrated that our approach allows to learn maps of large-scale environments in real-time with as few as 100 samples. Simultaneously, it outperforms previous approaches with respect to robustness and efficiency.

CHAPTER 6

Lazy Data Association in SLAM

6.1 Introduction

It is widely acknowledged that the SLAM problem consists of a continuous and a discrete component [Dissanayake *et al.*, 2001, Thrun, 2002]. The continuous estimation problem pertains to the location of individual features in the environment and the pose of the robot relative to these features. The discrete aspect of the SLAM problem is the *data association problem* [Bar-Shalom and Fortmann, 1988, Cox, 1993, Montemerlo and Thrun, 2003], which is the problem of determining whether or not two features observed at different points in time correspond to the same object in the physical world. Data association problems arise when matching two consecutive range scans [Lu and Milios, 1998] or when closing a large cycle in the environment [Bosse *et al.*, 2003, Gutmann and Konolige, 1999]. Unfortunately, the number of possible data associations may grow exponentially over time. With unknown data association, the SLAM posterior can in the worst case possess exponentially many modes whereas it commonly contains only a single mode for SLAM problems with known data associations.

The data association problem has been addressed extensively in the SLAM literature [Thrun *et al.*, 1998a, Neira and Tardós, 2001, Tardós *et al.*, 2001]. The problem here is to decide whether an assignment is a good choice or if we can find a better or correct one. This can be done in a *proactive* way by maintaining one or multiple hypotheses or in a *lazy* way by changing the assignments later on when we detect an error. In the previous chapters we have seen two different proactive approaches: Scan-matching (see Section 2.3.2) and Rao-Blackwellized Mapping (see Chapter 5). Although both described techniques do not deal with landmarks, the underlying data association problem is the same. We can assign

every grid cell as an own landmark, or we can try to figure out the association between poses. For the sake of simplicity, we do not explicitly separate these cases and use the formulation of the landmark based SLAM problem.

The most basic example of a proactive strategy is incremental maximum likelihood (ML) data association, which chooses the most likely hypothesis (see Section 2.3.2). In more sophisticated approaches, ML data association decisions are made for groups of features at-a-time [Tardós *et al.*, 2001]. Other proactive algorithms generate many data association hypotheses when a feature is observed, and later terminate all but one of them as more sensor data arrives. Two examples of such an approach are the multi hypothesis Kalman filter (MHT) [Bar-Shalom and Fortmann, 1988] and particle filter-based algorithms like Rao-Blackwellized Mapping (see Chapter 5). These algorithms are significantly more robust, but at the expense of a much higher computational overhead required for managing many hypothetical maps. The main problem with proactive techniques is computational: In ambiguous situations, multiple data association hypotheses must be generated to ensure that - with high likelihood- the correct association is among them.

This chapter seeks to establish a *lazy* data association technique that can “repair” past data associations arbitrarily far back into the past. Just as the ML data associator, our approach picks the most likely data association when a feature is observed. However, it differs from ML in that it monitors sensor data to detect whether a different set of data associations (past and present) can yield a map of higher likelihood. When such an opportunity is detected, past data association decisions are revised accordingly. We illustrate our approach in the context of several challenging mapping task, one involving a subterranean vehicle mapping a mine. A comparison with the popular Rao-Blackwellized mapping algorithm illustrates the advantages of lazy over proactive data association.

6.2 Preliminaries

6.2.1 SLAM with Known Data Association

We adopt the common probabilistic formulation of the feature-based SLAM problem. The vehicle pose at time t is denoted as x_t . We will use $x_{1:t}$ to denote the sequence of vehicle poses from time 1 to time t . The environment of the vehicle is composed of N features whose locations will be marked with $\Theta = \theta_1, \dots, \theta_N$. The vehicle’s estimate of Θ is the map. The goal of SLAM is to recover the map of Θ and the vehicle path $x_{1:t}$ from sensor measurements and robot controls. Measurements will be denoted by z_t , and controls by u_t . In SLAM algorithms we often try to achieve this goal by recovering the posterior distribution over the map

and the vehicle pose $p(\xi_{1:t}, \Theta | z_{1:t}, u_{1:t})$.

Under *known data association* and Gaussian noise, this posterior can be estimated using the extended Kalman filter (EKF) [Moutarlier and Chatila, 1989, Smith and Cheeseman, 1986, Smith *et al.*, 1990]. However, the EKF is inefficient since the time for the update step is quadratic in N . A flurry of recent research has led to a number of algorithms that can perform the update in constant time under the assumption of known data association [Leonard and Feder, 1999, Montemerlo *et al.*, 2003, Thrun *et al.*, 2002].

6.2.2 Incremental Data Association

Let $c_{1:t}$ be a vector of *correspondence variables*. The variables are discrete and their values are in $\{1, 2, \dots, N\}$. If $c_t = n$, the measurement z_t corresponds to the feature θ_n . If $c_t = c_s$ for two different points in time s and t , both measurements detected the same object in the physical world. We now seek to identify the sequence of correspondence variables $c_{1:t}$ that maximize the posterior:

$$\hat{c}_{1:t} = \underset{c_{1:t}}{\operatorname{argmax}} p(x_{1:t}, \Theta | z_{1:t}, u_{1:t}, c_{1:t}) \quad (6.1)$$

Unfortunately, the maximization is carried out over t variables; furthermore, these variables interact, and there is an exponential number of values that the combined vector $c_{1:t}$ can take. This exponential complexity makes finding the correct data association difficult.

Incremental ML approaches to SLAM bypass this problem by estimating the t -th data association \hat{c}_t at the time the t -th measurement arrives and freeze it forever for the future. This is equivalent to assuming that the optimal setting of the association variables can be found by optimizing one after the other:

$$\hat{c}_t = \underset{\hat{c}_t}{\operatorname{argmax}} p(x_{1:t}, \Theta | z_{1:t}, u_{1:t}, \hat{c}_t, \hat{c}_{1:t-1}) \quad (6.2)$$

where the variables $\hat{c}_{1:t-1}$ are held constant in the optimization on the right-hand side. Figure 6.1(a) illustrates this approach. Shown there is the tree of possible data association variables, with time 1 at the root and time t at the leaves. The incremental approach greedily follows what appears to be the most likely path at each step, from the root on down. This is illustrated by the path highlighted in gray in Figure 6.1(a). Unfortunately, once a wrong choice has been made, the incremental ML approach cannot recover such a situation. Moreover, wrong data association decisions introduce errors in the map which, subsequently, can induce more errors in the data association. For this reason, incremental ML data association is considered brittle in SLAM.

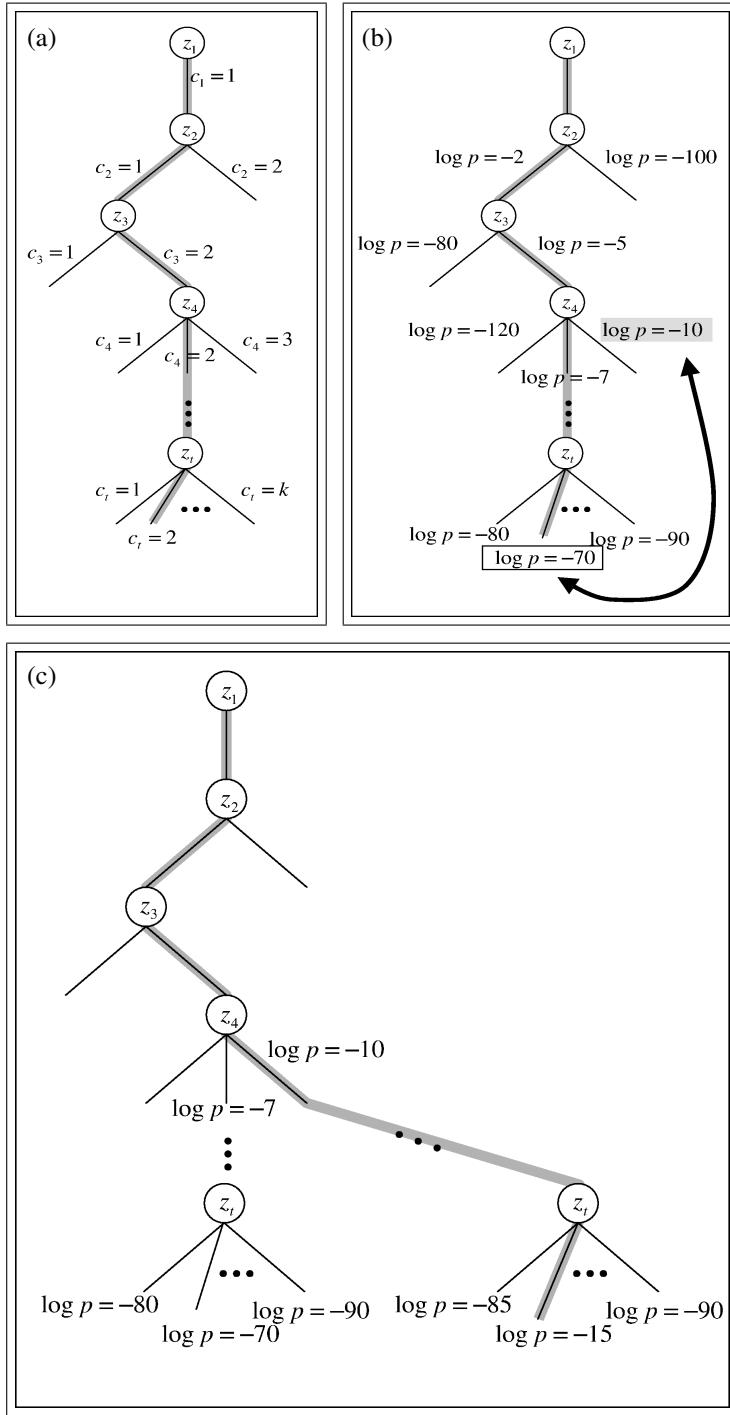


Figure 6.1: (a) The data association tree, whose branching factor grows with the number of landmarks in the map. (b) The proposed algorithm maintains the log-likelihood for the entire frontier of expanded nodes, enabling it to find alternative paths. (c) The improved path generated by the algorithm.

6.3 Lazy Data Association

6.3.1 Tree Search

Our approach uses a search procedure for considering alternative data association decisions not just at the current time step, but also for time steps in the past. A simple argument enables us to drastically reduce the number of nodes expended during this search. This argument is reminiscent of that underlying the correctness of the A* algorithm [Nilsson, 1982]. Figure 6.1(b) illustrates the basic idea: Our approach maintains not just a single path through the data association tree, but an entire frontier. Every time a node is expanded (e.g., through incremental ML), all alternative outcomes are also assessed and the corresponding likelihoods are memorized. This is illustrated in Figure 6.1(b), which depicts the log-likelihood for an entire frontier of the tree. Notice that we chose to represent the likelihood values as log-likelihoods, which is numerically more stable than probability values.

Finding the maximum in Equation (6.1) implies that the log-likelihood of the chosen leaf is greater or equal to that of any other leaf at the same depth. Since the log-likelihood decreases monotonically with the depth of the tree, we can guarantee that we indeed found the optimal data association values when the log-likelihood of the chosen leaf is greater or equal to the log-likelihood of any other node on the frontier. Put differently, when a frontier node assumes a log-likelihood greater than the one of the chosen leaf, there might be an opportunity to further increase the likelihood of the data by revising past data association decisions. Our approach then simply expands such frontier nodes. If an expansion reaches a leaf, this leaf is chosen as the new data association; otherwise the search is terminated when the entire frontier possesses values that are all smaller or equal to the one of the chosen leaf. This approach is guaranteed to always maintain the best set of values for the data association variables; however, occasionally it might require substantial search.

6.3.2 Equivalency Constraints

An important missing part is a SLAM representation that lets us efficiently modify data association variables. Our approach effectively implements the idea of global data association, but it does so via a set of auxiliary variables, called *equivalency variables*. Each such variable has the form $\gamma(t, s)$ where t and s are two different points in time. The equivalency relation $\gamma(t, s)$ holds if and only if c_t and c_s correspond to the same physical landmark, that is, $c_t = c_s$. Clearly, each assignment of the data association variables $c_{1:t}$ defines a set of such equivalency relationships. Conversely, each set of equivalency relationships constrains the space of all valid

data association values. In the limit as all equivalency relations are recovered, the data association can be determined up to a simple index permutation (which can never be recovered, since the index of a feature is arbitrary). The optimization problem defined in Equation (6.1), thus, becomes one of finding equivalency constraints.

Let Γ define an arbitrary set of such pairwise equivalency constraints. Each constraint $\gamma \in \Gamma$ has the form $\gamma(t, s)$. The goal then is to identify the optimal set of constraints Γ , which is the set of constraints that maximizes the posterior:

$$\underset{\Gamma}{\operatorname{argmax}} p(x_{1:t}, \Theta \mid z_{1:t}, u_{1:t}, \Gamma) \quad (6.3)$$

The notion of equivalency constraints makes it possible to relate two features to each other even though their absolute identity is unknown.

6.3.3 Recovering the Path Posterior under Equivalency Constraints

Our approach for recovering the path posterior is similar in spirit to the Lu/Milios algorithm [Lu and Milios, 1997]. The key insight is that equivalency constraints can be ‘translated’ into soft constraints that tie together two poses in the path posterior. More specifically, consider the constraint $\gamma(t, s)$. This constraint can be ‘softened’ into a constraint that ties together the location of the feature detected by z_t and the one detected by z_s . Such a ‘softened’ constraint can be expressed by a quadratic constraint of the form:

$$[f(z_t, x_t) - f(z_s, x_s)]^T R [f(z_t, x_t) - f(z_s, x_s)] \quad (6.4)$$

Here R is a quadratic penalty, and f is the function that projects the measurement z_t into three dimensional coordinates, based on the robot pose x_t . In general, f is a non-linear projection; however, by approximating it with a first order Taylor expansion we obtain a quadratic constraint of the form

$$\left[A \begin{pmatrix} x_t \\ x_s \end{pmatrix} - a \right]^T R \left[A \begin{pmatrix} x_t \\ x_s \end{pmatrix} - a \right] \quad (6.5)$$

with a Jacobean matrix A and a vector a (indices omitted for brevity). This quadratic constraint captures the information pertaining to the robot path, which arises from multiple observations of the same feature according to the data association constraint $\gamma(t, s)$.

Additional constraints for the robot path originate from the robot controls $u_{1:t}$, here written in negative log-form

$$-\log p(x_{1:t} \mid u_{1:t}) = -\log p(x_1) - \sum_{\tau=2}^t \log p(x_\tau \mid u_\tau, x_{\tau-1})$$

$$\begin{aligned}
&= \text{const.} + \frac{1}{2} \sum_{\tau=2}^t [x_\tau - g(u_\tau, x_{\tau-1})]^T Q_\tau [x_\tau - g(u_\tau, x_{\tau-1})] \\
&\approx \text{const.} + \frac{1}{2} \sum_{\tau=2}^t \left[B_\tau \begin{pmatrix} x_\tau \\ x_{\tau-1} \end{pmatrix} - b_\tau \right]^T Q_\tau \left[B_\tau \begin{pmatrix} x_\tau \\ x_{\tau-1} \end{pmatrix} - b_\tau \right]
\end{aligned} \tag{6.6}$$

Each B_τ is a Jacobean matrix, and b is a vector. The last step of this approximation involves again a Taylor expansion, in which the nonlinear motion model is linearized. Adding Equations (6.5) and (6.6) together leads to a system of quadratic equations in the path variables $x_{1:t}$ of the form

$$J := \text{const.} + [D \cdot x_{1:t} - d]^T P [D \cdot x_{1:t} - d] \tag{6.7}$$

Here D is a sparse matrix that links together elements in the path vector, and d is a vector. The matrix P defines the Mahalanobis distance composed of the terms R and Q_τ . This quadratic function is (up to a constant) the logarithm of a Gaussian approximation to the posterior over the robot path under the constraint set Γ . The mean of this Gaussian is recovered by setting the first derivative to zero:

$$\frac{\partial J}{\partial x_{1:t}} = D^T P [D \cdot x_{1:t} - d] \stackrel{!}{=} 0 \tag{6.8}$$

The solution to this equality is given by

$$x_{1:t} = (D^T P D)^{-1} D^T P d \tag{6.9}$$

where the matrix in the inversion is high-dimensional but extremely sparse. The covariance is simply the second derivative of J :

$$\frac{\partial^2 J}{\partial x_{1:t}^2} = D^T P D \tag{6.10}$$

The mean and covariance are the estimate of the SLAM posterior at time t . The classical solution to this problem involves the inversion of a sparse matrix, which is costly [Gupta *et al.*, 1997]. However, there exists a number of efficient approximations, such as loopy belief propagation [Murphy *et al.*, 1999] and tree-based approximation techniques [Wainwright, 2002] for approximating these quantities; all of those techniques can exploit an existing solution when modifying the set of equality constraints.

The key insight into this set of quadratic equations is that incrementally adding or removing a constraint can be done computationally very efficiently. This is the direct result of the matrix inversion lemma. Suppose we would like to add a

constraint of the form $\gamma(t, s)$. This leads to a local modification of the matrix D and the vector d in (6.7), which involve only x_t and x_s :

$$\begin{aligned} D' &\leftarrow D + S\Delta S^T \\ d' &\leftarrow d + S\delta \end{aligned} \quad (6.11)$$

Here Δ and δ are a low-dimensional matrix, and vector, respectively. The matrix S is a projection matrix for mapping low-dimensional vectors back into high-dimensional spaces. According to the inversion lemma gives us a new solution for the inverse term in (6.9):

$$\begin{aligned} (D'^T P D')^{-1} &= ([D + S\Delta S^T]^T P [D + S\Delta S^T])^{-1} \\ &= (D^T P D + 2S^T \Delta^T S P D + S^T \Delta^T S P S^T)^{-1} \end{aligned} \quad (6.12)$$

6.3.4 Incorporating Negative Measurement Information

The approach described so far only accounts for matching points in three dimensional space. Equality constraints bend the path of the robot relative to the path reconstructed from pure odometry. As a result, the likelihood maximizing set of equality constraints would be the empty set, and each feature would simply be declared a new one. This is because a bent path is less likely under the robot's control variables than a non-bent one.

The problem with this approach is that it does not account for “negative” information. Negative information pertains to situations where a robot fails to see a measurement. Range sensors, which are brought to bear in our implementation, return positive and negative information with regards to the presence of objects in the world. The positive information are object detections. The negative information applies to the space between the detection and the sensor. The fact that the robot failed to detect an object closer than its actual reading provides information about the *absence* of an object within the measurement range.

To evaluate the effect of a new constraint on the overall likelihood of the data, our approach evaluates both types of information: positive and negative. Both types are obtained by calculating the pairwise (mis)match of two scans under their pose estimate. In our implementation, the log-likelihood of each measurement is obtained in the same way as described for the scan-matching in Section 2.3. In doing so, it is straightforward to determine the probability of a measurement in a way that incorporates both the positive and the negative information.

6.4 Experimental Results

We have implemented our algorithm and tested it in realistic setting with large-scale data sets. Our first data set was acquired in an abandoned mine [Baker *et*

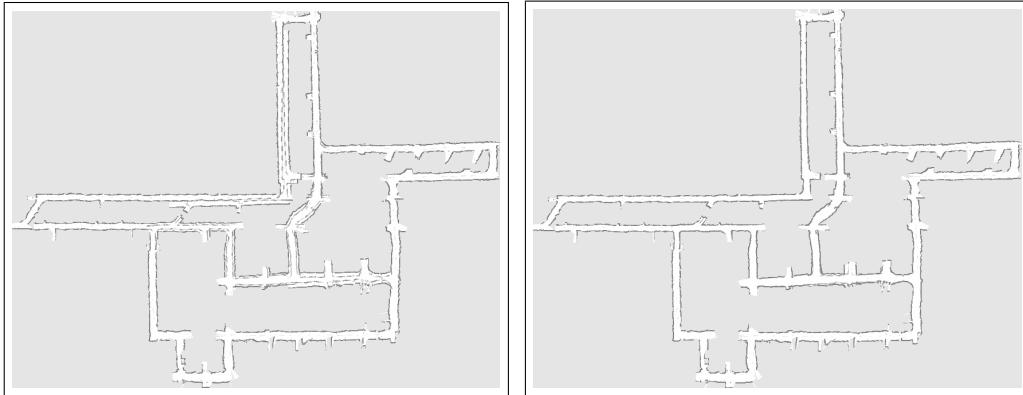


Figure 6.2: (left) Mine map with incremental ML scan matching and (right) using our lazy data association approach. The map is approximately 250 meters wide, and acquired without odometry information.

al., 2003, Thrun *et al.*, 2003], using a vehicle equipped with a laser range finder. The vehicle does not provide any odometry or controls information and no values for $u_{1:t}$ are available. A further difficulty arises from the absence of well-defined “landmarks” in the mine. When faced with raw 10^9 laser measurements, the number of possible data association variables is beyond what can be handled computationally. To make this problem tractable, we modified the basic algorithm in a number of ways. Instead of using all poses in the optimization, our approach acquires local occupancy maps of approximately five meters length, assuming that within a local map, the incremental ML data association techniques works reliable (which in practice it does as described in Chapter 2). As a result, we only have to align a few hundred local maps, making the problem computationally tractable. Further, we have not yet implemented the most efficient version of our algorithm (e.g., we are not using Equation (6.12)), which makes our implementation slower than real-time.

The left panel of Figure 6.2 depicts the result of incremental ML data association, which is equivalent in our case to regular incremental scan matching. Obviously, certain corridors are represented doubly in this map, illustrating the shortcomings of the ML approach. The right panel, in comparison, shows the result of our approach. Clearly, this map is more accurate than the one generated by the incremental ML approach. Its diameter is approximately 250 meters wide, and the floor of the mine is highly uneven.

Figure 6.3 (left) illustrates the log-likelihood of the most recent measurement (not the entire path), which drops significantly as the map becomes inconsistent. At this point, our approach engages in searching alternative data association values. It quickly finds the “correct” one and produces the map shown in Fig-

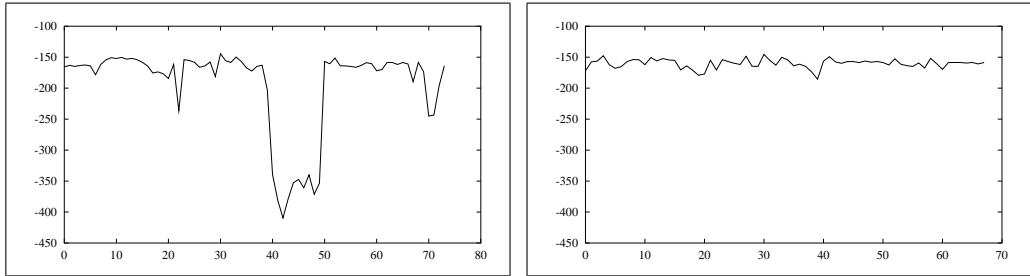


Figure 6.3: (left) Log-likelihood of the actual measurement, as a function of time. The lower likelihood is caused by the wrong assignment. (right) Log-likelihood using our approach, which recursively fixed false data association hypotheses. The success of our approach is manifested by the lack of a distinct dip.

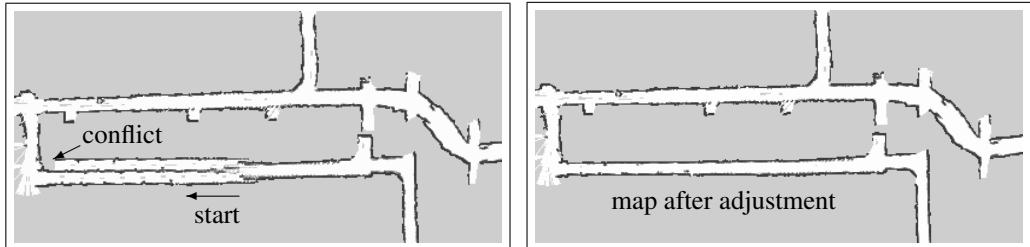


Figure 6.4: Example of our lazy data association technique: (a) When closing a large loop, the robot first erroneously assumes the existence of a second, parallel hallway. However, this model leads to a gross inconsistency as the robot encounters a corridor at a right angle. At this point, our approach recursively searches for improved data association decisions, arriving on the map shown in diagram (b).

ure 6.2 (right). The critical area is shown in Figure 6.4, illustrating the moment at which the likelihood takes its dip. The log-likelihood of the measurement for our approach is shown in Figure 6.3 (right).

We also compared lazy data association with the Rao-Blackwellized mapping algorithm. Our comparison is based on the implementation described in the previous chapter, which addresses mapping with laser range finders (instead of idealized point features). The data set was gathered in a large indoor environment, using a Pioneer 2 robot equipped with a laser range-finder. To make this problem difficult, the robot first traversed a small loop a number of times, before closing a larger loop. This is shown in Figure 6.5(a). Figure 6.5(b) shows the result for the incremental ML approach, which is implemented here as an incremental scan matching algorithm. The map built with Rao-Blackwellized mapping is shown

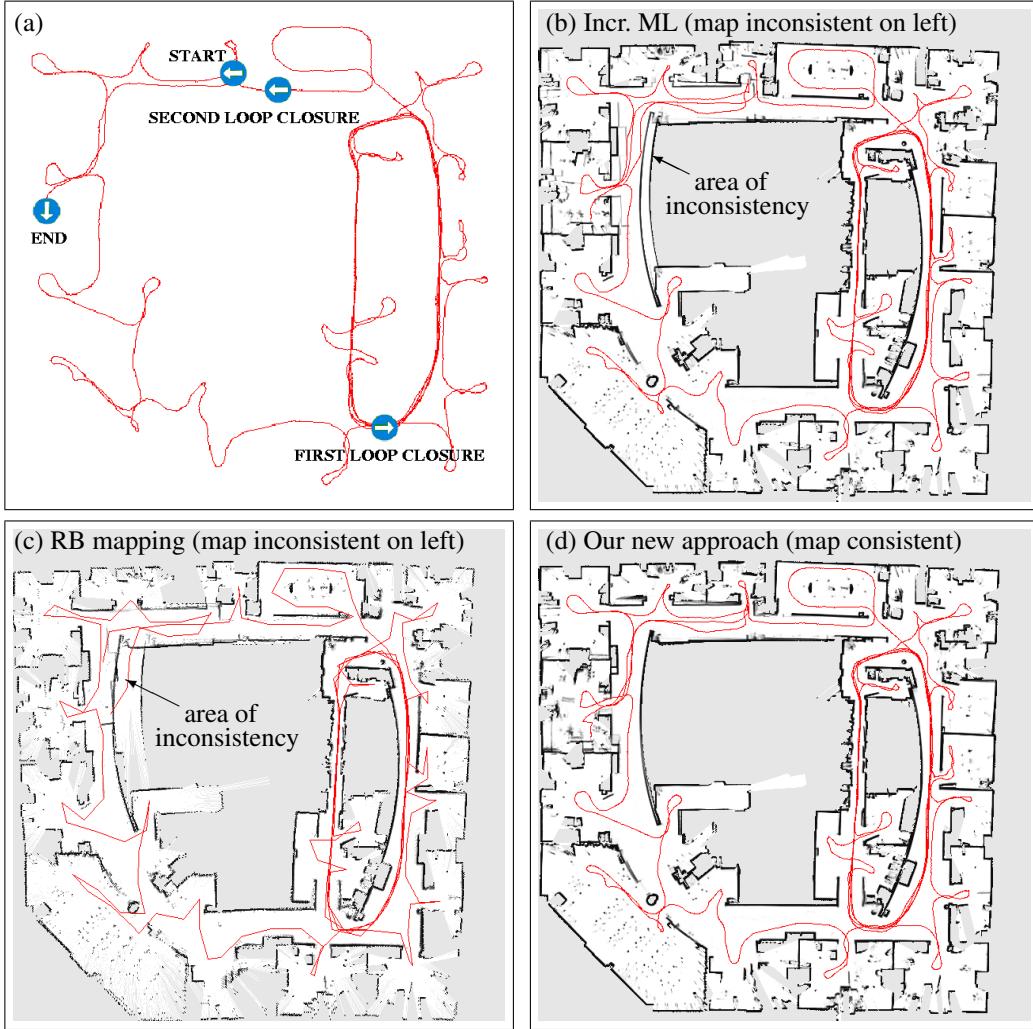


Figure 6.5: (a) Path of the robot. (b) Incremental ML (scan matching). (c) Rao-Blackwellized mapping. (d) Our approach.

in Figure 6.5(c), for 100 particles. Both of these maps show inconsistencies in the upper left corner. Our approach produces the map in Figure 6.5(d), which is significantly more accurate.

When mapping the small cycle, Rao-Blackwellized mapping runs out of particles. This is shown in Figure 6.6, which plots the particles (including their paths) before and after closing the small cycle on the right. While in principle, the problem can be reduced by using larger particle sets, eventually such a deprivation takes place, posing intrinsic limits on Rao-Blackwellized mapping's ability to map large environments with many cycles. This is a fundamental problem inherent in

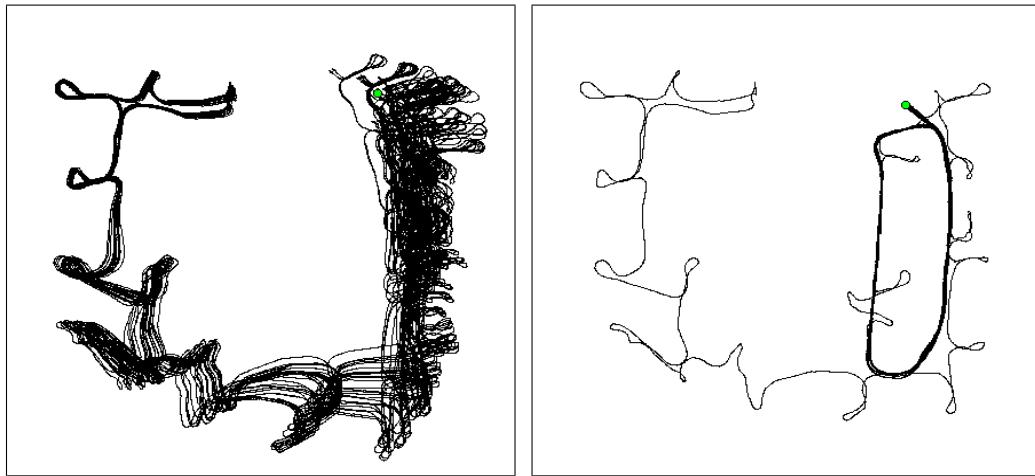


Figure 6.6: The problem with Rao-Blackwellized mapping is particle depletion: (left) Particle paths before closing the small loop on the right and (right) after closing it.

all proactive approaches; Rao-Blackwellized mapping is among the most robust proactive approaches in the present literature.

6.5 Summary

This chapter described a new algorithm for data association in SLAM. In essence, our approach searches the combinatorial tree of possible data association decisions. The search is lazy: only when an alternative assignment appears promising it will be evaluated. To achieve an efficient implementation, our approach condenses maps into graphical representation, and employs equality constraints for alleged data associations. Using linear algebra techniques, these constraints can be added or removed efficiently. We have evaluated our approach using some of the most challenging data sets in our possession, and have consistently found that it produces accurate maps, even if for maps with many large cycles.

CHAPTER 7

Mapping and Localization with RFID Technology

7.1 Introduction

In the previous chapters we presented approaches to learn a detailed geometric description of the environment. Nevertheless, we are not only interested in creating maps with a detailed geometric description of all objects in the scene. Sometimes one is only interested in locations of special objects. It would be straightforward if we could detect them in our regular data acquisition. The accuracy of the detection strongly depends on the deployed sensor, so often we cannot locate with a sufficient exactness.

In recent years automatic identification procedures have become very popular in many service industries, purchasing and distribution logistics, industry, manufacturing companies and material flow systems. Automatic identification systems do exist to provide information about people, animals, goods and products. Due to the technique used for the transfer of power and data, contact-less ID systems are called RFID systems (Radio Frequency Identification). RFID is a technology that is been around since at least the 1970s. Up to now, it has been too limited or too expensive to be practical for many applications. Within the context of RFID there are several methods for identification, but the most common one is to store a unique serial number to identify persons or objects. A RFID tag mostly consists of a chip and an antenna. The antenna enables the chip to transmit the identification information to a reader. The RFID tag reader can read the information from a certain distance. Typically we can assume that the place of the RFID tag is in the vicinity of the machine's antenna. Unfortunately this can be a region of several



Figure 7.1: Typical RFID tags used to label objects. The size of the tag depicted in the center is 11×5 cm.

meters and sometimes, in the case of reflections, the place of the RFID tag can be outside the usual expected vicinity of the antenna. This shows that building maps of special objects and their location based on RFID tags can be a difficult problem.

Recent advances in the field of radio frequency identification techniques have reached a state that will allow us within the next years to equip virtually every object in an environment with small, cheap Radio Frequency Identification (RFID) tags [Finkenzeller, 2000]. Figure 7.1 depicts three different RFID tags that were used to carry out the experiments described in this chapter. The detection range of these tags is approximately 6 m.

RFID tags open up a wide variety of applications. For example, an important problem in the health-care sector is the recognition of daily activities a home patient is engaged in. The Guide project [Philipose *et al.*, 2003] uses small RFID readers worn by a person to identify the objects the person touches. The sequence of contacted objects is used by a Bayesian reasoning system to estimate the activity of the person and to provide support if needed. Location context can provide important information for the interpretation of RFID readings. For example, grasping the toothpaste has very different meanings depending on whether it happens in the storage room or in the bathroom.

In this chapter, we investigate how RFID technology can be enhanced by location information. We use a mobile robot equipped with RFID antennas to determine the locations of RFID tags attached to objects in an indoor environment. Figure 7.2 (left) depicts the robot built to carry out this research. The robot consists of an off-the-shelf Pioneer 2 robot equipped with a laser range scanner and two RFID antennas. The antennas are mounted on top of the robot and point ap-



Figure 7.2: Pioneer 2 with Sick Laser Range Finder, RFID reader and two antennas (left). Experimental setup used for learning the likelihood function of tag detections (right).

proximately 45 degrees to the left and to the right with respect to the robot. To use these antennas for estimating the locations of objects, we first learn a sensor model that describes the likelihood of detecting an RFID tag given its location relative to one of the antennas. Since the noise of these sensors is highly non-Gaussian, we represent the measurement likelihood model by a piecewise constant approximation. Then we describe a technique to estimate the locations of RFID tags using a mobile robot. This process uses a map previously learned from laser range data. We then apply Monte Carlo localization [Dellaert *et al.*, 1999, Fox *et al.*, 1999a] to estimate the pose of the robot and even of moving persons in this environment. Experimental results suggest that it is possible to accurately localize moving objects based on this technology. Further experiments demonstrate that RFID tags greatly reduce the time required for a global localization of the mobile robot in its environment. Additionally, this technology can be used to drastically reduce the number of samples required for global localization.

This chapter is organized as follows. In the following section we will present the sensor model for RFID receivers. Then we describe in Section 7.2 how this model can be used in combination with the laser-based Rao-Blackwellized mapping approach, presented in Chapter 5, to effectively determine and map the locations of RFID tags. In Section 7.4 we describe how the resulting beliefs about the locations of the tags can be utilized to determine the position of the robot and of persons in the environment. We show experimental results illustrating the advantages of RFID tags for robot localization and person tracking in Section 7.5. Finally, we discuss the related work in Section 7.6.

7.2 Learning a Probabilistic Sensor Model for the RFID Antenna

To localize an RFID tag in a global reference frame, we estimate the posterior $p(x | z_{1:t}, r_{1:t})$, where x is the position of the tag, $z_{1:t}$ are the observations at time steps $1, \dots, t$, and $r_{1:t}$ are the possibly different locations of the RFID antenna. According to Bayes rule and under the assumption of independence of consecutive measurements given we know the location x of a tag we obtain the following recursive update rule:

$$p(x | z_{1:t}, r_{1:t}) \propto p(z_t | x, r_t) p(x | z_{1:t-1}, r_{1:t-1}) \quad (7.1)$$

According to this equation, the key term is the quantity $p(z_t | x, r_t)$ which specifies the likelihood of the observation z_t given the position x of the tag and the location r_t of the antenna. We make the simplifying assumption that this likelihood only depends on the *relative* offset between tag and antenna. This means that it only depends on the difference between x and r_t . The following aspects need to be considered when designing an observation model for RFID tags.

1. There are plenty of *false-negative readings*, i.e., situations in which the tag is not detected although it is in the vicinity of the antenna
2. Additionally, we obtain *false-positive readings*. In such cases the antenna detects a tag that is not in range specified by the manufacturer. This also includes detection of the RFID tag with the wrong antenna.

There are several reasons for these measurement errors. For example, the orientation of the tag with respect to the RFID receiver influences the energy absorbed by its own antenna. Depending on this angle, the energy will vary and might not be high enough to power the chip inside the tag. In such a case the tag will simply not respond. Furthermore, the shape and size of the detection range largely depends on the environment. For example, metal typically absorbs the energy sent by the RFID reader and therefore tags attached to metallic objects will be detected only in a short range. But even other objects influence the detectability of tags. For example, if a tag is attached to a concrete wall its detection statistics typically changes drastically. Furthermore, the radio frequency waves emitted by the antenna can be reflected by objects such that the antenna even detects objects outside the specified detection range. Note that the observation model for the RFID antennas should be able to cover this wide range of situations and should not make the robot overly confident in the location of a particular tag or even in its own location during localization.

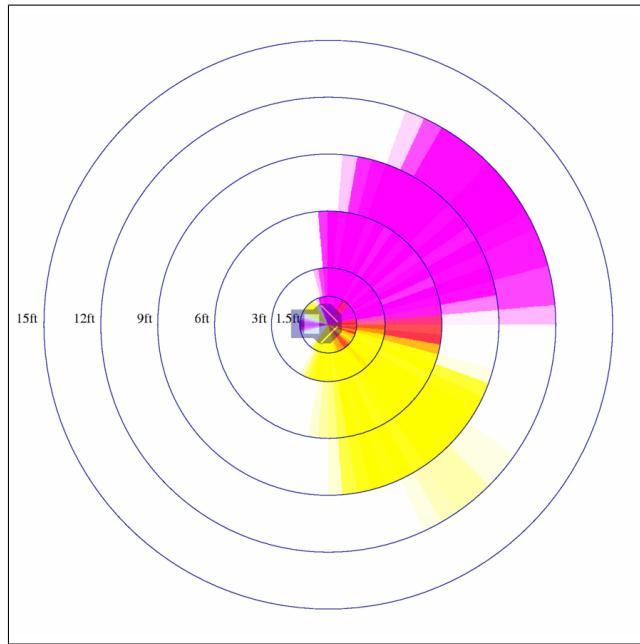


Figure 7.3: Detection field for the left (upper/green histogram) and right (lower/red histogram) antenna. The middle/blue histogram shows the area where the tag can be seen by both antennas.

To determine the observation model for the RFID antennas we generated a statistics by counting frequencies. We proceeded in the following way. We attached an RFID tag to a box and rotated the robot in front of the box (see right image of Figure 7.2). We repeated this for different distances and counted for every point in a discrete grid the frequency of detections of the antenna given the tag was placed at a position covered by this grid cell relative to the robot.

The resulting histogram is shown in Figure 7.3. This figure contains the detection statistics for both sensors. The histograms were built from 12,822 measurements. As can be seen from the figure, both antennas show quite different behaviors although they were measuring the same RFID tag.

The resulting sensor model used to conservatively approximate the histograms is shown in Figure 7.4. This model consists of three components. The major detection range for each antenna consists of an arc with an opening angle of 95 degrees in the direction of the antenna. Additionally, an antenna always detects RFID tags that are close to it even if they are behind the antenna. This is modeled by a circular region around the center of the receiver. The corresponding likelihood for the two detection ranges are also depicted in Figure 7.4. For locations outside these areas we assume a constant likelihood of 0.5.

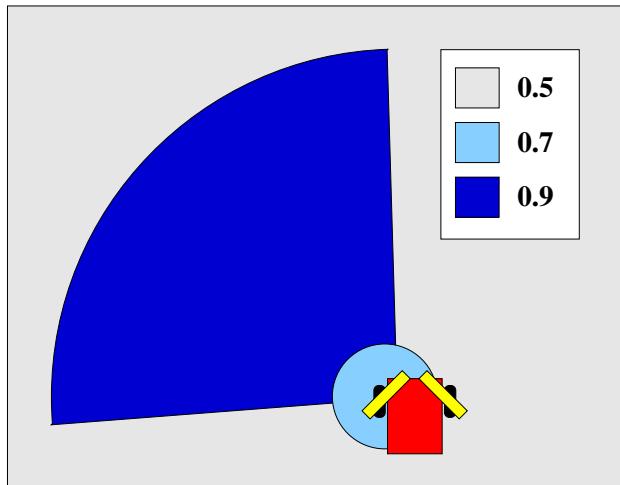


Figure 7.4: Simplified sensor model for the left antenna.

7.3 Mapping RFID Tags

The first application of the sensor model described in the previous section is estimating the location of RFID tags in the environment using a mobile robot. To learn the positions of the tags our system proceeds in two steps. First it learns the geometric structure of the environment using a laser range sensor. Afterwards it estimates the positions of the tags based on the path of the robot.

Since our robot is equipped with a laser range scanner, we apply the Rao-Blackwellized mapping algorithm (see Chapter 5) to learn the geometrical structure of the environment. The map used for the experimental results is depicted in Figure 7.5. Given this map and the maximum likelihood path of the robot computed by the Rao-Blackwellized mapping algorithm we can now estimate the locations of the RFID tags. Here we apply the recursive Bayesian filtering scheme given in Equation 7.1, with $r_{1:t}$ representing the path of the robot.

To represent the belief about the location of an RFID tag we use a set of 1000 randomly chosen positions uniformly distributed in a 25 square meter wide area around the current pose of the robot. This area is independent of the antenna that detected the tag in order to avoid that a detection failure of an antenna results in a suboptimal placement of the sampled positions. It is initialized at the first detection of the RFID tag by the robot.

To each of the randomly chosen potential positions we assign a numerical value storing the posterior probability $p(x | z_{1:t}, r_{1:t})$ that this position corresponds to the true position of the tag. Whenever the robot detects a tag, the posterior is updated according to Equation 7.1 and using the sensor model described in the previous section.

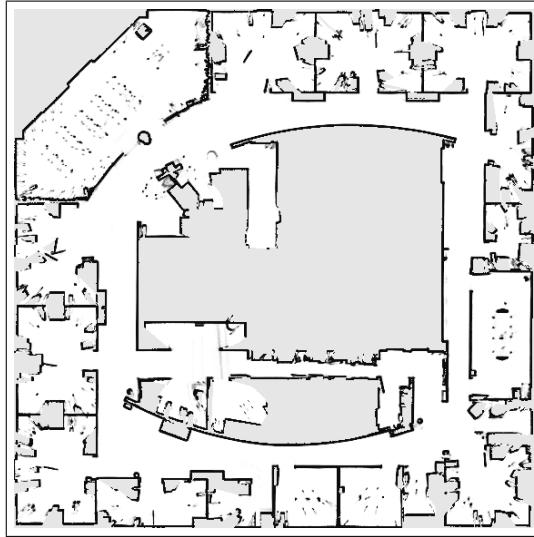


Figure 7.5: Map of the Intel Research Lab Seattle generated by our Rao-Blackwellized mapping routine.

7.4 Localization with RFID Tags

Given the posterior distribution $p(x | z_{1:t}, r_{1:t})$ over potential positions of an RFID tag we are now ready to compute the likelihood of an observation z^l during localization, given the robot or a person is placed at a location l . According to the law of total probability we obtain

$$p(z^l | l) = \sum_x p(z^l | x, l) p(x | z_{1:t}, r_{1:t}). \quad (7.2)$$

In this equation the term $p(z | x, l)$ corresponds to the relative sensor model described in Section 7.2. The relative offset of the sensor is computed from the global coordinates of the detected RFID tag, x , and the robot pose, l . Thus, to determine the likelihood of a tag detection given the robot is at location l , we have to integrate over the posterior probability of the tag's location given the data obtained during the mapping process.

To estimate the pose l of the robot or of persons in the environment, we apply the well-known recursive Bayesian filtering scheme:

$$\begin{aligned} p(l_t | z_{1:t}^l, u_{0:t-1}) &= \alpha \cdot p(z_t^l | l_t) \\ &\cdot \int_{l'_{t-1}} p(l_t | u_{t-1}, l'_{t-1}) \cdot p(l'_{t-1} | z_{1:t-1}^l, u_{0:t-2}) d l'_{t-1} \end{aligned} \quad (7.3)$$

Here α is a normalization constant ensuring that $p(l_t | z_{1:t}, u_{0:t-1})$ sums up to one over all l_t . The term $p(l_t | u_{t-1}, l'_{t-1})$ describes the probability that the object is

at position l_t given it executed the movement u_{t-1} at position l'_{t-1} . This quantity is computed depending on the object we are tracking. In case of a mobile robot we compute this quantity based on the odometry measurements [Fox *et al.*, 1999a]. If we are tracking persons, we simply represent this density by a Gaussian centered around l_t . Furthermore, the quantity $p(z_t | l_t)$ denotes the likelihood of the observation z_t according to our observation model, which is computed using Equation 7.2. To represent the posterior about the pose of the object being tracked we apply a Monte-Carlo localization [Dellaert *et al.*, 1999, Fox *et al.*, 1999a]. In Monte-Carlo localization, the belief of the robot is represented by a set of random samples [Arulampalam *et al.*, 2002]. Each sample consists of a state vector of the underlying system, which is the pose l of the robot in our case, and a weighing factor ω . The latter is used to store the importance of the corresponding particle. The posterior is represented by the distribution of the samples and their importance factors. The particle filter algorithm used by our system is also known as *sequential importance sampling with resampling* [Arulampalam *et al.*, 2002]. It updates the belief about the pose of the robot according to the following two alternating steps:

1. In the prediction step, we draw for each sample a new sample according to the weight of the sample and according to the model $p(l_t | u_{t-1}, l'_{t-1})$ of the robot's dynamics given the movement u_{t-1} executed since the previous update. In the case of localizing a person, this model is simply a Gaussian centered at l_{t-1} .
2. In the correction step, the new observation z_t is integrated into the sample set. This is done by bootstrap resampling, where each sample is weighted according to the observation likelihood $p(z_t | l_t)$.

To globally localize the object, we initialize the particle set with a uniform distribution. In the case of RFID sensors, we fortunately can efficiently sample potential locations of the object. We simply place samples only in the potential detection range of the RFID sensor. Such an approach has been applied successfully in the past, for example by Lenser and Veloso [Lenser and Veloso, 2000].

7.5 Experimental Results

Our approach described above has been implemented and tested using a Pioneer 2 robot equipped with a SICK LMS laser range-finder and an Alien Technology's 915 MHz RFID reader with two circularly polarized antennas (see left image of Figure 7.2). The experiments described here were carried out in the Intel Research Lab, Seattle, WA. Figure 7.5 shows a two-dimensional occupancy grid map generated with our Rao-Blackwellized mapping routine. The size of the environment

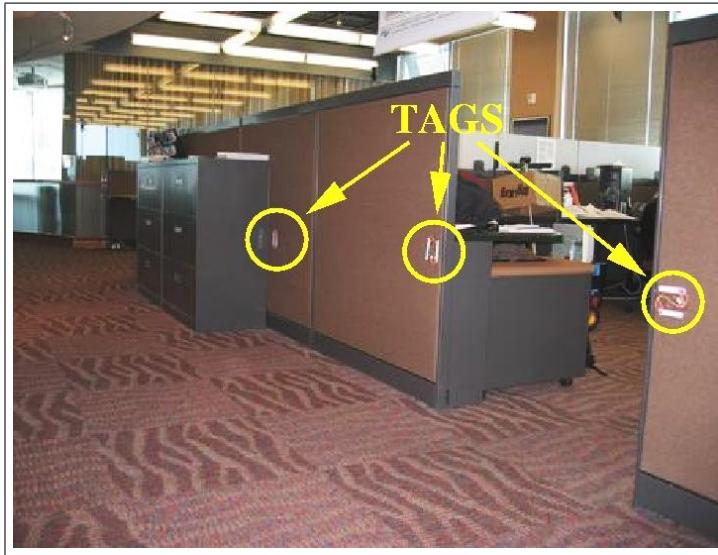


Figure 7.6: RFID tags attached to walls.

is 28m by 28m. We installed 100 tags in this environment (see Figure 7.6). The tags were of the types depicted in Figure 7.1 and all of them were able to communicate with the robot. Most of them were installed along the circular corridor of the environment.

7.5.1 Mapping RFID tags

As already mentioned above, we use the trajectory estimated by our Rao-Blackwellized mapping routine to determine the posterior about the locations of the tags. When a tag is detected for the first time, we initialize a discrete set of randomly chosen points around the robot and use a uniform distribution to initialize the belief. Whenever a tag is detected, the posterior probability of each sample in that set is multiplied with the likelihood of the observation given the tag is at the position corresponding to that sample. Afterwards we normalize the belief over all samples.

Figure 7.7 shows a typical example for the evolution of the belief of an RFID tag. The leftmost image shows the initial sample set after the first detection of an RFID tag. The remaining images illustrate how the belief focuses on the true position of the tag as more measurements are obtained. They show the corresponding beliefs after 6, 17, 65, 110, and 200 measurements. Note that the diameter of each circle representing a particle corresponds to its importance weight. As can be seen from the figure, the belief quickly converges to a unimodal distribution. Note that this is not necessarily the case. In principal, our representation can also handle

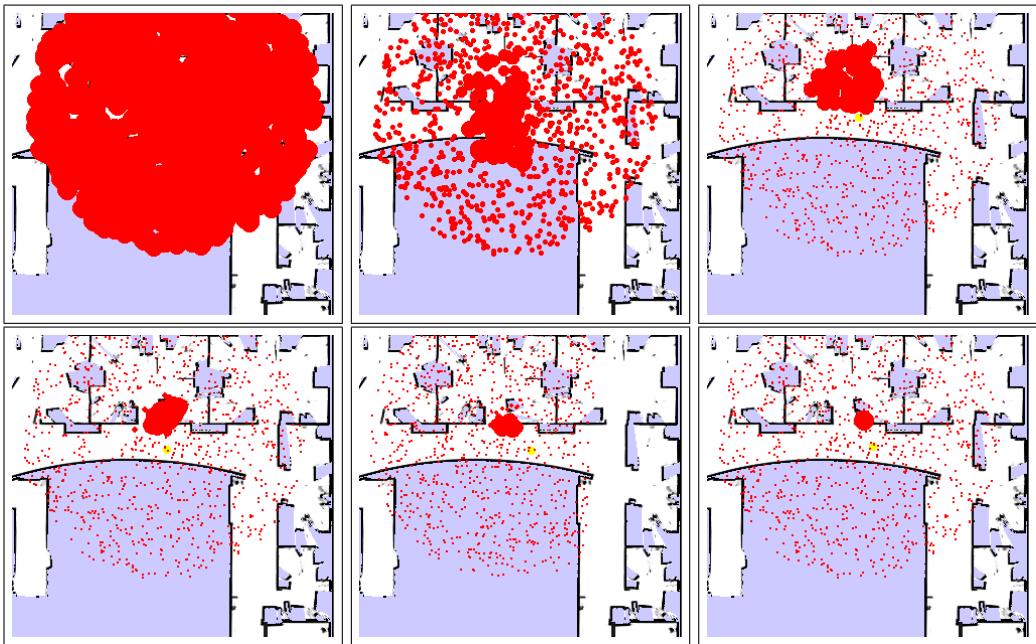


Figure 7.7: Evolution of the posterior about the localization of an RFID tag over time. The width of the circles represents the importance weight of the corresponding particle. It is drawn proportional to the ratio between the importance weights of the particular sample and the maximum likelihood sample.

ambiguities in which the location of an RFID tag cannot be determined uniquely, for example, because the robot cannot reach locations which are required to resolve the ambiguity.

Figure 7.8 depicts the positions of the robot when it detected the tag, for which the beliefs are plotted in Figure 7.7. Detections of the right antenna are displayed by filled circles and for each detection of the left antenna we draw an unfilled circle. As can be seen from the figure, the measurement noise is quite high and there are several false detections. Nevertheless, our algorithm is able to accurately localize the tag at the wall close to the entrance.

After traveling 791.93m with an average speed of 0.225m/s the robot had processed 50,933 detections of RFID tags. The resulting map of the tags (at their maximum likelihood position) is shown in Figure 7.11 (left). Thus, our sensor model allows to learn the positions multiple tags in a standard office environment.

7.5.2 Localization with RFID Tags

The next set of experiments is designed to illustrate that the RFID map generated in the previous step can be used to localize the robot and even persons equipped

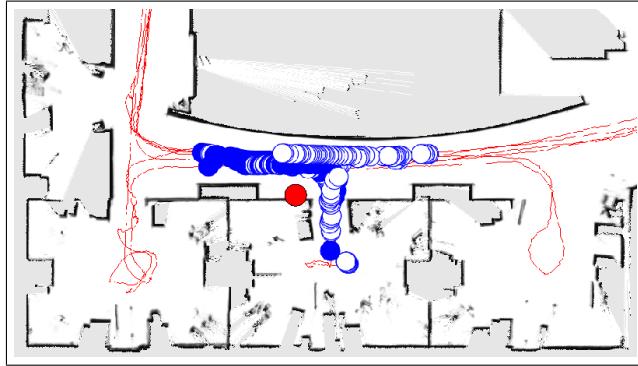


Figure 7.8: Places where the robot has detected the RFID tag with the left (unfilled circle) or right antenna (filled circle)

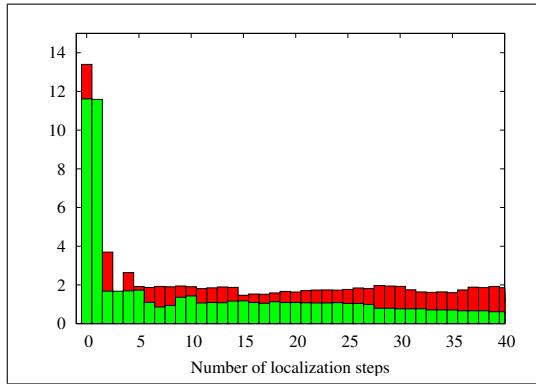


Figure 7.9: Error (in m) during global localization with (green) and without (red) odometry using RFID tags only.

with RFID antennas.

In the first experiment we steered the robot through the environment and applied Monte-Carlo localization to globally estimate the position of the vehicle. To simulate the situation in which we localize a person instead of the robot we simple ignored the odometry information and changed the motion model in the Monte Carlo localization procedure. As already mentioned above we used a standard motion model [Fox *et al.*, 1999a] to estimate the pose of the robot. In order to localize and keep track of a person we simply replaced this motion model by a Gaussian distribution centered around the current pose. Note that this is only a rough approximation of the motions of a person. Better models therefore can be expected to result in more accurate estimates.

Figure 7.9 shows the localization error during a global localization run using RFID tags only. The two plots show the localization error for global localization

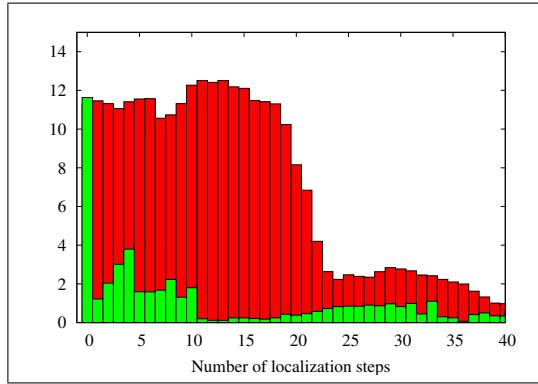


Figure 7.10: Positioning error of the laser based global localization (in m) without (red) and with (green) RFID data.

without odometry (red) and with odometry (green).

The center image of Figure 7.11 shows the trajectory for the object being tracked when no odometry information is used. The corresponding ground-truth obtained by laser-based localization is depicted in the right image of the same figure. As can be seen, even with such noisy sensors the estimated trajectory is quite close to the ground truth.

7.5.3 Improving Global Localization with RFID Tags

The final experiment is designed to illustrate that the RFID technology can be used to drastically improve the global localization performance even in the case where highly accurate sensors such as laser range finders are used. To analyze this we used a pre-recorded data set to figure out how efficiently the robot can determine its global position in this map. Since the RFID tags are only placed close to the corridor we generated samples only in the corridor of the environment. We compared the time required for global localization using laser data with the time needed when laser and RFID tags were used simultaneously. Figure 7.10 shows the average localization error for a typical run for both cases. As the figure illustrates, global localization can be achieved much faster when laser and RFID data are combined (green) compared to a situation in which only laser data is used (red).

Additionally, the use of RFID sensors can greatly reduce the number of samples required for global localization. Figure 7.12 shows the localization error depending on the number of particles for the case in which only laser data is used as well as for the situation in which the laser data is combined with RFID information. It turns out that laser-based global localization is efficient when at least 10.000 particles are used. On the other hand, if we fuse the laser data with the

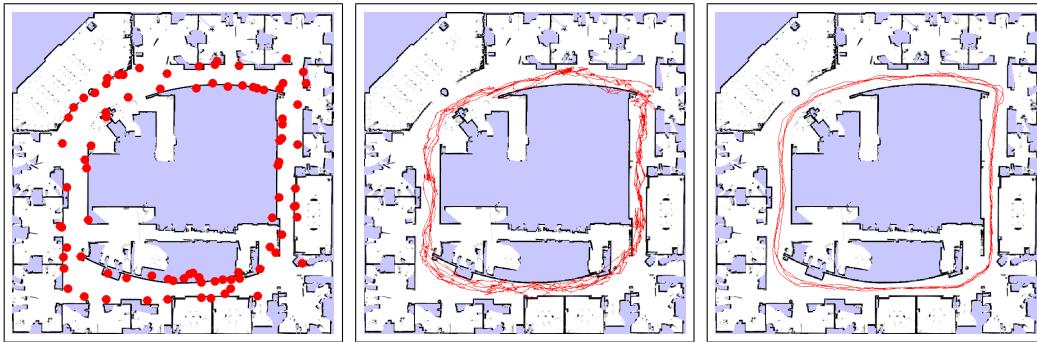


Figure 7.11: Map of Intel Lab with most likely positions of the RFID tags (left), estimated trajectory without odometry (center), and the corresponding ground truth (right).

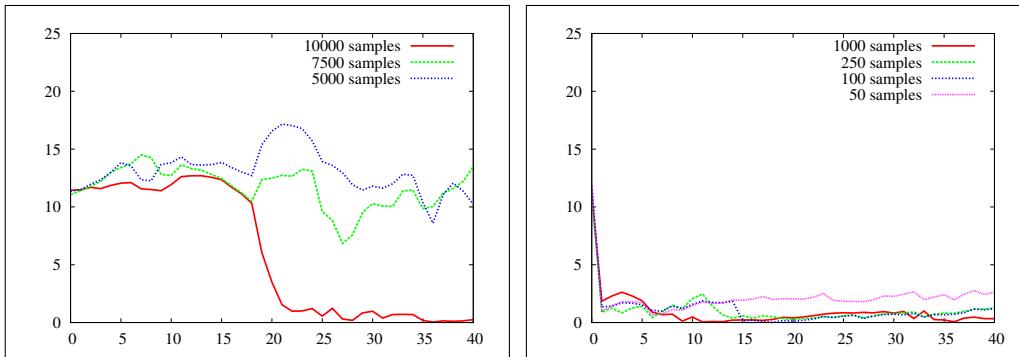


Figure 7.12: Localization error (in m) during global localization for different numbers of particles and depending on whether only laser data is used (left image) or whether the combination of laser data and RFID measurements is used (right image).

information about the RFID tags, we can globally localize the object with as few as 50 samples.

7.6 Related Work

In the last years RFID sensors [Finkenzeller, 2000] have started to enter the field of mobile robotics. Nowadays RFID readers can detect low-cost passive tags in the range of several meters. These improvements in the detection range of passive tags make this technology more and more attractive for robotics applications since the information provided by tags can be used to support various tasks like navigation, localization, mapping, and even service applications such as people tracking.

Most of the applications of RFID technology, however, assume that the readers are stationary and only the tags that are attached to objects or persons move. The main focus is to trigger events if a tag is detected by a reader or entering the field of range (for example, to keep track of the contents of storage places [Brusey *et al.*, 2003]). Recently Kantor and Singh used RFID tags for mapping. Their system relies on active beacons which provide distance information based on the time required to receive the response of a tag. Additionally, the positions of the tags have to be known more or less accurately [Singh *et al.*, 2002, Kantor and Singh, 2002]. Tsukiyama [Tsukiyama, 2003] also requires given RFID tag positions. Their system assumes perfect measurements and does not include techniques to deal with the uncertainty of the sensor.

The problem considered here is closely related to the simultaneous localization and mapping (SLAM) problem, in which a robot has to generate a map while simultaneously estimating its pose relative to this map. However, since RFID sensors do not provide distance or angle information, SLAM-techniques for range-only [Singh *et al.*, 2002, Kantor and Singh, 2002], bearing-only [Deans and Herbert, 2000] or range and bearing [Dissanayake *et al.*, 2000, Leonard and Feder, 1999, Thrun *et al.*, 1998b] cannot be applied directly to the data provided by the RFID system. Our algorithm instead uses a variant of Rao-Blackwellized mapping to learn the geometric structure of the environment using laser data (see Chapter 5) and then estimates the positions of the tags based on the trajectory.

7.7 Summary

In this chapter we presented an approach to generate maps of RFID tags with mobile robots. We presented a sensor model that allows us to compute the likelihood of tag detections given the relative position of the tag with respect to the robot. Additionally we described how to compute a posterior about the position of a tag after the trajectory and the map has been generated with a highly accurate Rao-Blackwellized mapping algorithm for laser range scans. We furthermore presented how the posterior can be used to localize a robot as well as persons in the environment.

The system has been implemented on a Pioneer 2 robot that was augmented by two RFID antennas. In practical experiments we demonstrated that the system can build accurate maps of RFID tags. We furthermore illustrated that the resulting maps can be used for accurate localization of the robot and moving objects even without odometry information. Finally we presented an experiment demonstrating that the combination of a laser-range scanner and RFID technology can greatly reduce the computational demands for the global localization of a moving mobile robot.

CHAPTER 8

Modeling Nonrigid Objects with Mobile Robots

8.1 Introduction

In recent years, there has been a flurry of work on acquiring 3D models from range data. The classical setting involves a range sensor (e.g., a 3D range camera or a stereo vision system) used to acquire range images of the target object from multiple vantage points. The problem of integrating multiple range scans into a 3D model is commonly known as *scan registration* (see Section 2.4). Most state-of-the-art implementations are based on the popular *iterative closest point* algorithm (see Section 2.4.1). The topic has received significant attention in fields as diverse as computer vision [Rusinkiewicz and Levoy, 2001, Huot *et al.*, 2000] and medical imaging [Feldmar *et al.*, 1997], large-scale urban modeling [Teller *et al.*, 2001], and mobile robotics [Lu and Milios, 1997, Gutmann and Konolige, 1999, Thrun, 2001].

ICP aligns range scans by alternating a step in which closest points are identified, and a step by which the optimal translation and rotation of scans relative to each other is computed. In doing so, most of them are making a rigid object assumption: range scans, if aligned correctly, must be spatially consistent with each other.

Many objects are deformable. For example, people change shape, as do trees, pillows, and so on. A natural research goal is therefore to extend ICP to accommodate local object transformations. Following recent work primarily found in the medical imaging literature [Chui and Rangarajan, 2000, Wang and Staib, 2000, Chung *et al.*, 2003, Bakircioglu *et al.*, 1999], we propose an approach

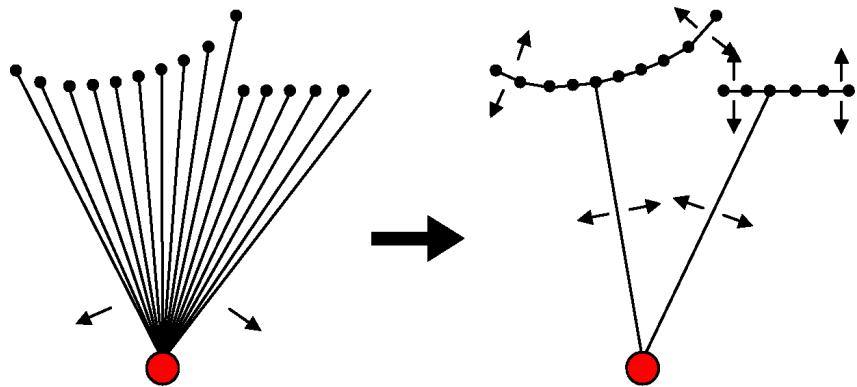


Figure 8.1: The essential idea: Rather assuming that the relation of measurement coordinates are fixed relative to the location from which the measurement was taken, the approach proposed here constrains the relation of measurement coordinates in a soft way. The exact configuration of a scan is calculated while scans are registered to each other.

suited for scan registration and 3D modeling of nonrigid objects that can efficiently deal with hundreds of thousands of variables. To accommodate local deformations, this approach transforms scans into Markov random fields, where nearby measurements are linked by a (nonlinear) potential function. All links are soft, which means that they can be bent, but bending them incurs a penalty. Figure 8.1 illustrates this transformation: rigid links between the measurement coordinates and the robot sensor are replaced by soft links between adjacent measurement points. The resulting problem of scan registration under these soft constraints becomes a high-dimensional optimization problem with orders of magnitude more variables involved than in regular ICP. We show how to solve this problem via linearization by Taylor series expansion, and then propose a coarse-to-fine hierarchical optimization technique for carrying out the optimization efficiently.

This new algorithm is applied to the problem of learning 3D models of non-stationary objects with a mobile robot. We describe an implemented robot system that utilizes a model differencing technique similar to the one described in [Anguelov *et al.*, 2002] to segment scans. By acquiring views of the target objects from multiple sides, the approach enables a robot to acquire a 3D model of a non-stationary object.

8.2 ICP Variant for 3D Scan Registration

This section describes a variant of the previously described *iterative closest point* algorithm (see Section 2.4.1) for *rigid* objects. Like the original ICP, the algo-

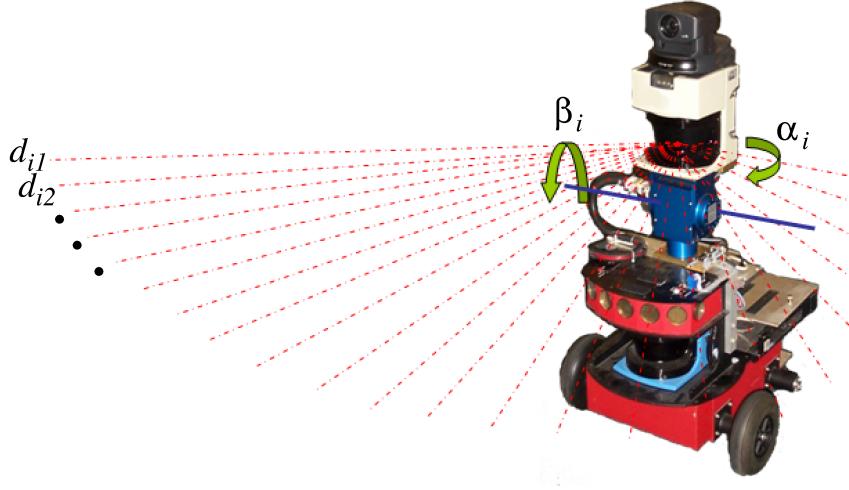


Figure 8.2: Mobile robot with two laser range scanners. The robot can acquire 3D data by tilting the upper scanner

rithm alternates two phases, one in which nearest points is identified, and one in which the distance between all pairs of nearest points is minimized. This variant describes the ICP algorithm in slightly different probabilistic framework and focuses on operating conditions for mobile robots. This helps us to modify the variant to extend the technique to register *non-rigid* scans.

8.2.1 Three-dimensional Scans taken from a Mobile Robot

The input to the ICP algorithm is a set of 3D scans denoted

$$\mathbf{d} = \mathbf{d}_1, \mathbf{d}_2, \dots \quad (8.1)$$

Each such scan \mathbf{d}_k consists of a collection of 1D range measurements, arranged as a 3D “range image”:

$$\mathbf{d}_k = d_{1k}, d_{2k}, \dots \quad (8.2)$$

Figure 8.3b shows a typical scan acquired by a robot. Further below, we will denote the horizontal angle of the k -th range measurement by α_k and the vertical one by β_k . In our experimental setup, scans are obtained by a SICK laser range finder mounted on a tilt unit shown in Figure 8.2.

The problem of scan registration can be formulated as the problem of recovering the vantage points from which the scans were taken. In our approach, scans

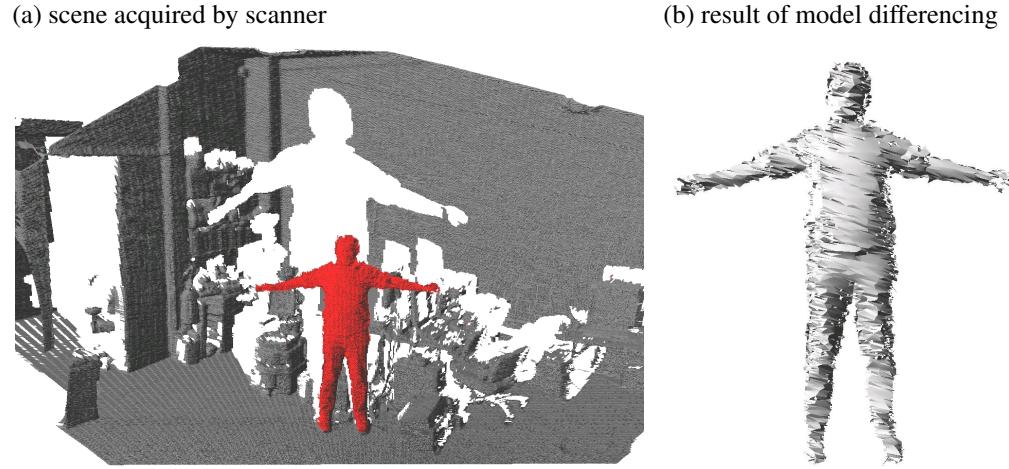


Figure 8.3: (a) 3D range scan acquired by mobile robot shown in Figure 8.2. (b) Scene from which the scan in (a) is extracted through background differencing.

are taken from a mobile robot; hence each vantage point is described by three variables: its x - y location in Cartesian coordinates and its orientation γ :

$$\mathbf{x}_k = (x_k, y_k, \gamma_k)^T \quad (8.3)$$

Here \mathbf{x}_k denotes the vantage point from which scan \mathbf{d}_k was acquired. The set of all vantage points will be denoted \mathbf{x} . Recovering the vantage points \mathbf{x} is equivalent to registering the scans if one vantage point is (arbitrarily) defined to be $\mathbf{x}_1 = (0, 0, 0)^T$.

8.2.2 Measurement Model

Scans are registered in world coordinates. To do so, measurements d_{ik} must be mapped into 3D world coordinates. This is achieved by a projective function π , which takes as an argument a range measurement and a vantage point and returns as its output the corresponding coordinate in 3D world coordinates:

$$\pi(d_{ik}, \mathbf{x}_k) = \begin{pmatrix} x_k + d_{ik} \cos(\gamma_k + \alpha_i) \sin \beta_i \\ y_k + d_{ik} \sin(\gamma_k + \alpha_i) \sin \beta_i \\ z + d_{ik} \cos \beta_i \end{pmatrix} \quad (8.4)$$

Here α_i and β_i are the orientation of the range measurement d_{ik} relative to the sensor. The variable z is the generic height of the sensor which in our robot system is fixed, but is easily generalized to variable heights. For brevity, we will sometimes write π_{ik} instead of $\pi(d_{ik}, \mathbf{x}_k)$.

We define the quality of a pairwise scan registration by the probability representing the likelihood $p(d_{ik})$ of a range measurement under a fixed (hypothetical) registration:

$$p(d_{ik} | \mathbf{d}_l, \mathbf{x}_k, \mathbf{x}_l) = \begin{cases} \max_j |2\pi\Sigma|^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2}(\pi_{ik} - \pi_{jl})^T \Sigma^{-1} (\pi_{ik} - \pi_{jl}) \right\} & \text{if } \pi_{ik} \in \mathcal{F}_l \\ \text{const.} & \text{if } \pi_{ik} \notin \mathcal{F}_l \end{cases} \quad (8.5)$$

This likelihood distinguishes two cases. The first case models the noise in range perception by a Gaussian with a sensor-specific covariance Σ (usually a diagonal matrix). The measurement error under this Gaussian is given by the distance between the point π_{ik} under consideration, and the point j in scan \mathbf{d}_l that maximizes this Gaussian. As is easily to be seen, this point π_{jl} minimizes the Mahalanobis distance to π_{ik} . Thus, π_{jl} is simply the point “nearest” to π_{ik} in the scan \mathbf{d}_l under Σ^{-1} .

However, finding the nearest point only makes sense if π_{ik} falls within the perceptual range of scan \mathbf{d}_l . If π_{ik} is occluded, it might be perfectly well-explained by an object not detectable from the vantage point \mathbf{x}_l . This is captured by the second case in (8.6), which applies when π_{ik} lies outside the free space of scan l . The free space of scan l is denoted \mathcal{F}_l . It is defined as the region between the robot and the detected objects. If π_{ik} lies outside this region, the measurement probability is assumed to be uniform. The value of this uniform depends on the range of occluded space, but it plays no role in the optimization to come; hence we leave it unspecified.

8.2.3 Registration as Likelihood Maximization

The goal of scan registration is to determine the vantage points \mathbf{x} that maximize the joint likelihood of the scans \mathbf{d} and \mathbf{x} :

$$p(\mathbf{d}, \mathbf{x}) = p(\mathbf{d} | \mathbf{x}) p(\mathbf{x}) \quad (8.6)$$

The first term of this product is obtained by calculating the product over all individual measurement likelihoods, assuming noise independence in each measurement:

$$p(\mathbf{d} | \mathbf{x}) = \prod_k \prod_i \prod_{l \neq k} p(d_{ik} | \mathbf{d}_l, \mathbf{x}_k, \mathbf{x}_l) \quad (8.7)$$

If there are more than two scans, this product over counts the evidence (measurement noise is penalized more than once); however, since our goal lies in the optimization of the likelihood, not in calculating its actual value, this shall be of no

concern. The term $p(\mathbf{x})$ in (8.6) is the *prior* on the individual vantage points (and hence on the registration). The prior of the point \mathbf{x}_k is expressed as a Gaussian with mean and covariance matrix Ψ : In our robot system, this prior is obtained from the robot's 2D localization routines, supplied by the techniques presented in the previous chapters.

The negative logarithm of the joint likelihood (8.6) is given by the following sum:

$$\begin{aligned} G &= -\log p(\mathbf{d}, \mathbf{x}) \\ &= \text{const.} + \frac{1}{2} \sum_k [(\mathbf{x}_k - \bar{\mathbf{x}}_k)^T \Psi^{-1} (\mathbf{x}_k - \bar{\mathbf{x}}_k) \\ &\quad + \sum_i \sum_{l \neq k} \min_j (\pi_{ik} - \pi_{jl})^T \Sigma^{-1} (\pi_{ik} - \pi_{jl})] \end{aligned} \quad (8.8)$$

Scans are registered by minimizing the sum, which is equivalent to maximizing the likelihood function.

Unfortunately, minimizing G is not possible *in closed form* for the following three reasons: (1) the exact terms in the sum depend on the occlusion constraint which are a function of the vantage points \mathbf{x} ; (2) the result of the minimization over j involves a discrete search for a nearest neighbor, whose outcome is again a function of \mathbf{x} ; (3) the projection functions π are nonlinear in the points \mathbf{x} , and the resulting nonlinear optimization problem is hard and has no closed form solution.

8.2.4 Optimization Procedure

ICP-style algorithms minimize the negative log-likelihood by calculating a sequence of vantage points (registrations):

$$\mathbf{x}_k^{[0]}, \mathbf{x}_k^{[1]}, \mathbf{x}_k^{[2]}, \dots \quad (8.9)$$

The first set of points are obtained from the prior (e.g., the robot odometry): The $(n+1)$ -th vantage points are computed from the n -th ones by the following five step algorithm:

Step 1. The set of occluded points is determined for the n -th points $\mathbf{x}_k^{[n]}$. This step involves the calculation of a relative orientation of a point π_{ik} to scan \mathbf{d}_l via \mathbf{x}_l . It furthermore involves a range comparison with the corresponding measurement in scan \mathbf{d}_l , to determine if π_{ik} is occluded relative to \mathbf{d}_l .

Step 2. The minimization in (8.8) is carried out by determining the closest measurement j in each scan relative to each point π_{ik} :

$$l_{ikl}^{[n]} = \min_j (\pi_{ik} - \pi_{jl})^T \Sigma^{-1} (\pi_{ik} - \pi_{jl}) \quad (8.10)$$



Figure 8.4: Example for registration with ICP of five *rigid* scans. The rightmost image shows the resulting model.

For $\Sigma = \text{const.} \cdot I$, this calculation is equivalent to finding the closest points (hence the name of the algorithm).

Step 3. The projection function π is linearized to obtain a quadratic objective function. This is achieved by the Taylor series expansion:

$$\pi(d_{ik}, \mathbf{x}_k) \approx \pi_{ik}^{[n]} + \mathbf{J}_{ik}^{[n]}(\mathbf{x}_k - \mathbf{x}_k^{[n]}) \quad (8.11)$$

Here $\mathbf{J}_{ik}^{[n]}$ is the Jacobian (gradient) of π at $\mathbf{x}_k^{[n]}$, which is directly obtained from the definition of π in (8.4):

$$\mathbf{J}_{ik}^{[n]} = \begin{pmatrix} 1 & 0 & -d_{ik} \sin(\gamma_k + \alpha_i) \sin \beta_i \\ 0 & 1 & d_{ik} \cos(\gamma_k + \alpha_i) \sin \beta_i \\ 0 & 0 & 0 \end{pmatrix} \quad (8.12)$$

With this approximation, $G^{[n]}$ (the negative log-likelihood function G in the n -th iteration) becomes quadratic in \mathbf{x}_k and is of the form

$$\begin{aligned} G^{[n]} = & \text{const.} + \frac{1}{2}(\mathbf{x} - \bar{\mathbf{x}})^T \Psi^{-1} (\mathbf{x} - \bar{\mathbf{x}}) \\ & + \frac{1}{2}(\mathbf{A}^{[n]} \mathbf{x} - \mathbf{c}^{[n]})^T \tilde{\Gamma}^{-1} (\mathbf{A}^{[n]} \mathbf{x} - \mathbf{c}^{[n]}) \end{aligned} \quad (8.13)$$

Here $\mathbf{A}^{[n]}$ is a matrix, $\mathbf{c}^{[n]}$ a vector, and \mathbf{x} the vector of all vantage points (the unknowns). The various sums in (8.8) are simply integrated into $\mathbf{A}^{[n]}$ and $\mathbf{c}^{[n]}$,

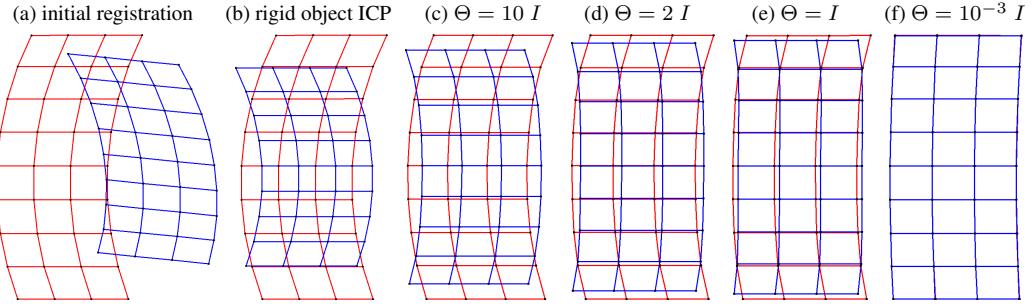


Figure 8.5: Illustration in 2D: (a) initial configurations of two scans in red and blue, (b) result of optimal registration, (c-f) result of recovering the object configuration by our new adjustable shape technique.

and $\tilde{\Sigma}$ generalizes Σ to the full space of all points \mathbf{x} :

$$\tilde{\Sigma}^{-1} = \begin{pmatrix} \Sigma^{-1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \Sigma^{-1} \end{pmatrix}^{-1} \quad (8.14)$$

Step 4. The x - y -values of the vantage points $\mathbf{x}^{[n+1]}$ are now obtained through a closed-form solution of minimizing $G^{[n]}$. This solution is obtained by setting the first derivative of $G^{[n]}$ with respect to \mathbf{x} to zero (derivation omitted, see [Strang, 1998]):

$$\begin{aligned} \mathbf{x}^{[n+1]} \\ = (\Psi^{-1} + \mathbf{A}^{[n]T} \Sigma^{-1} \mathbf{A}^{[n]})^{-1} (\Psi^{-1} \bar{\mathbf{x}} + \mathbf{A}^{[n]T} \Sigma^{-1} \mathbf{c}^{[n]}) \end{aligned} \quad (8.15)$$

In the original ICP algorithm [Besl and McKay, 1992], this step is implemented by calculating the “center of mass” of each scan and shifting the scans accordingly (which is computationally simpler); our approach also accommodates a prior.

Step 5. Because of the linearization, only Cartesian coordinates are actually updated in Step 4; the orientation coordinates remain unmodified. For this reason, the original ICP literature introduced a separate step for calculating orientations. This is achieved by a singular value decomposition (SVD) or an Eigenwert analysis step, in which the correlation between residual errors in the scan is determined, and each scan is rotated so as to minimize these error correlations. Details on this step can be found in Section 2.4.1 or in the ICP literature [Horn, 1987, Horn *et al.*, 1988, Chen and Medioni, 1991, Besl and McKay, 1992].

All steps are iterated until a convergence criterion is reached. Satisfactory registrations are usually obtained within the first three to four iterations. If the total

number of scans is small (see Figure 8.4), the computationally most expensive step is to determinate of the closest points in Step 2. This step is usually implemented efficiently by representing scans through kd-trees [Breiman *et al.*, 1984].

Figure 8.5a-b illustrate the result of scan registration in 2D. The initial configuration in Figure 8.5a is transformed into the one shown in Figure 8.5b, which is the one that minimizes the squared distance (maximizes the likelihood). Clearly, both scans are incompatible in shape. Pure registration techniques are unable to handle such shape deformations, but the technique presented in the next section is.

8.3 Recovering the Surface Configuration of Non-rigid Objects

The key idea for extending ICP to nonrigid objects was already discussed in the introduction to this chapter, and is highlighted in Figure 8.1. Technically, it involves two modifications: First, the static relationship between points π_{ik} and the corresponding vantage points \mathbf{x}_k is replaced by nonrigid links between adjacent points. These links can be bent (at a probabilistic penalty), to accommodate nonrigid surfaces. Second, and as consequence of this, the optimization now involves the determination of the location of all points π_{ik} , in addition to the robot poses \mathbf{x}_k . This optimization problem is high dimensional, and we will discuss a hierarchical optimization technique for tackling it efficiently. A key characteristic of the approach proposed here is that it fits neatly into the ICP methodology above: Again, under appropriate linearization the target function is quadratic, and estimates are obtained just as in (8.15).

8.3.1 Links

The definition of links between pairs of adjacent points makes it necessary to augment the notion of a measurement point. In particular, this approach associates an (imaginary) coordinate system with each node. The origin of each coordinate system is the familiar coordinate π_{ik} , and its orientation is specified by three Euler angles (an alternative formulation may use quaternions as in Section 2.4.1):

$$\mathbf{r}_{ik} = \begin{pmatrix} \phi_{ik} \\ \theta_{ik} \\ \psi_{ik} \end{pmatrix} \quad (8.16)$$

The orientation is initialized arbitrarily; e.g. $\mathbf{r}_{ik} = (0, 0, 0)^T$. (The result of the optimization is invariant with respect to this initialization.) A link is now

given by the affine coordinate transformations among the coordinate systems of adjacent measurements. Each link possesses six parameters, three for rotation (denoted $\Delta\mathbf{r}_{i \rightarrow j,k}$) and three for translation (denoted $\Delta\pi_{i \rightarrow j,k}$). They are calculated as follows:

$$\begin{aligned}\Delta\mathbf{r}_{i \rightarrow j,k} &= \mathbf{r}_{jk} - \mathbf{r}_{ik} \\ \Delta\pi_{i \rightarrow j,k} &= R_z(-\psi_{ik}) \cdot R_y(-\theta_{ik}) \cdot R_x(-\phi_{ik}) \begin{pmatrix} x_{jk} - x_{ik} \\ y_{jk} - y_{ik} \\ z_{jk} - z_{ik} \end{pmatrix}\end{aligned}\quad (8.17)$$

Here the R 's are the (obvious) rotation matrices around the three coordinate axes:

$$\begin{aligned}R_y(\phi) &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{pmatrix} \\ R_y(\theta) &= \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix} \\ R_z(\psi) &= \begin{pmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix}\end{aligned}$$

Links enable us to recover a node's coordinates from any of its neighbors:

$$\begin{aligned}\pi_{jk} &= \underbrace{\pi_{ik} + R_x(\phi_{ik}) \cdot R_y(\theta_{ik}) \cdot R_z(\psi_{ik}) \Delta\pi_{i \rightarrow j,k}}_{=: \hat{\pi}_{i \rightarrow j,k}} \\ \mathbf{r}_{jk} &= \underbrace{\mathbf{r}_{ik} + \Delta\mathbf{r}_{i \rightarrow j,k}}_{=: \hat{\mathbf{r}}_{i \rightarrow j,k}}\end{aligned}\quad (8.18)$$

To model nonrigid surfaces, our approach allows violations of these link constraints. This is obtained by introducing the following Gaussian potentials for each link

$$h_{i \rightarrow j,k} = |2\pi\Theta|^{-\frac{1}{2}} \exp \left\{ -\frac{1}{2} \left(\begin{array}{c} \pi_{jk} - \hat{\pi}_{i \rightarrow j,k} \\ \mathbf{r}_{jk} - \hat{\mathbf{r}}_{i \rightarrow j,k} \end{array} \right)^T \Theta^{-1} \left(\begin{array}{c} \pi_{jk} - \hat{\pi}_{i \rightarrow j,k} \\ \mathbf{r}_{jk} - \hat{\mathbf{r}}_{i \rightarrow j,k} \end{array} \right) \right\}$$

Here Θ defines the strength of the link (the resulting structure is a Markov random field [Winkler, 1995]).

8.3.2 Target Function

The negative logarithm of these potentials, summed over all links, is given by the following function H (constant omitted):

$$H = \frac{1}{2} \sum_{i \rightarrow j, k} \left(\begin{array}{c} \pi_{jk} - \hat{\pi}_{i \rightarrow j, k} \\ \mathbf{r}_{jk} - \hat{\mathbf{r}}_{i \rightarrow j, k} \end{array} \right)^T \Theta^{-1} \left(\begin{array}{c} \pi_{jk} - \hat{\pi}_{i \rightarrow j, k} \\ \mathbf{r}_{jk} - \hat{\mathbf{r}}_{i \rightarrow j, k} \end{array} \right) \quad (8.19)$$

As in the scan registration problem, all these terms are nonlinear in the node coordinates π_{ik} and the orientations \mathbf{r}_{ik} . To obtain a closed-form solution for the resulting equation system, the link function is linearized via a Taylor series expansion:

$$\left(\begin{array}{c} \pi_{jk} \\ \mathbf{r}_{jk} \end{array} \right) \approx \left(\begin{array}{c} \pi_{ik}^{[n]} \\ \mathbf{r}_{ik}^{[n]} \end{array} \right) + K_{ik, ik}^{[n]} \left(\begin{array}{c} \pi_{ik} - \pi_{ik}^{[n]} \\ \mathbf{r}_{ik} - \mathbf{r}_{ik}^{[n]} \end{array} \right) \quad (8.20)$$

where $\pi_{ik}^{[n]}$ and $\mathbf{r}_{ik}^{[n]}$ specify the coordinate system for the node ik in the n -th iteration of the optimization. The matrix $K_{ik, ik}^{[n]}$ is a Jacobian matrix of dimension six by six, which is obtained as the derivative of the functions (8.18).

By the same logic by which G can be approximated by a quadratic function in the scan registration problem, substituting our approximation back into the definition of H gives us a quadratic function in all variables π_{ik} and \mathbf{r}_{ik} . This function can be written in the form

$$H = \frac{1}{2} \left[\mathbf{B}^{[n]} \left(\begin{array}{c} \pi \\ \mathbf{r} \end{array} \right) - \mathbf{f}^{[n]} \right]^T \tilde{\Theta}^{-1} \left[\mathbf{B}^{[n]} \left(\begin{array}{c} \pi \\ \mathbf{r} \end{array} \right) - \mathbf{f}^{[n]} \right] \quad (8.21)$$

where π is the vector of all coordinate system origins, \mathbf{r} the vector of all Euler angles, and $\mathbf{B}^{[n]}$ is a matrix and $\mathbf{f}^{[n]}$ a vector. Calculating $\mathbf{B}^{[n]}$ and $\mathbf{f}^{[n]}$ is involved but mathematically straightforward.

8.3.3 Optimization Procedure

This new version of ICP now minimizes the combined target function $G + H$, which is again quadratic in all parameters (\mathbf{x} , π , and \mathbf{r}). By doing so, it simultaneously recovers the scan registration and the surface configuration of the object. The solutions for \mathbf{x} and π are completely analogous to the one in (8.15):

$$\left(\begin{array}{c} \pi^{[n+1]} \\ \mathbf{r}^{[n+1]} \end{array} \right) = (\mathbf{B}^{[n]T} \mathbf{B}^{[n]})^{-1} \mathbf{B}^{[n]T} \mathbf{f}^{[n]} \quad (8.22)$$

The global orientation is optimized by a single global SVD. Our new augmented optimization leads to relative adjustments between measurement points, in which the links play the role of soft constraints.

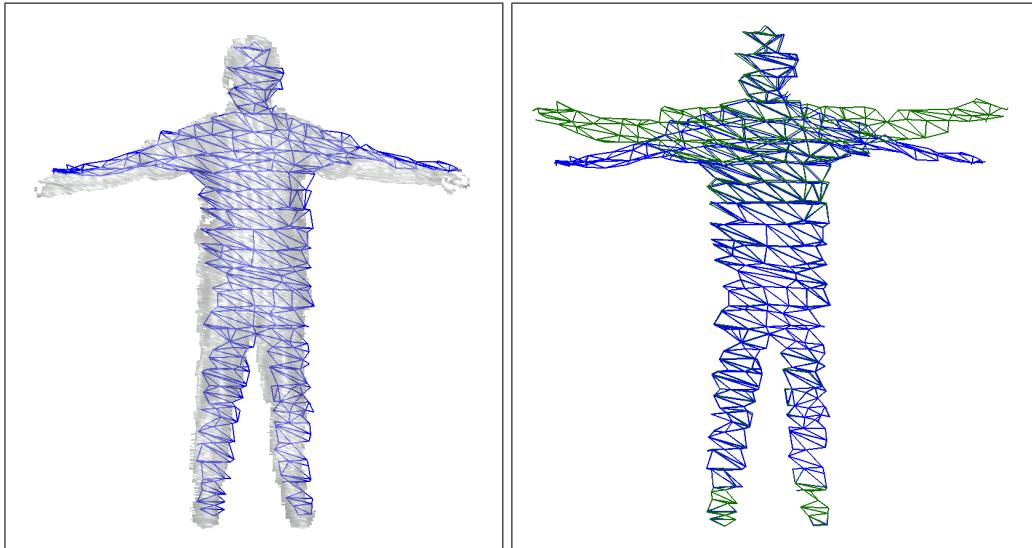


Figure 8.6: Example of a thinned graph superimposed to the original scan (left) and before and after adjustment (right). Thinning is necessary to perform the optimization efficiently.

This is illustrated in Figure 8.5(c-f), for different values for Θ . As the various diagrams illustrate, scans are deformed to improve their match. The degree of the deformation depends on the value of Θ , which defines the rigidity of the surface. Figure 8.5(c-f) illustrates that our approach succeeds in locally rotating and even rescaling the model.

8.3.4 Efficient Variable Resolution Optimization

The main problem with the approach presented so far is its enormous complexity. The number of variables involved in the optimization is orders of magnitude larger compared to scan registration. This is because the target function H is a function of all measurement points π and orientations r , whereas G has only the vantage points x as its arguments. The matrix $B^{[n]}$ in (8.22) is, thus, a (sparse) matrix with many thousand dimensions.

To tackle such problems efficiently the optimization is reduced to a sequence of nested optimization problems. In a first step, scans are analyzed for connected components (regions without large disparities); links exist only between connected components in each scan; hence H factors naturally into different subproblems for different connected components. Next, the resulting scan patches are thinned. Thinning proceeds by identifying a small number of representative

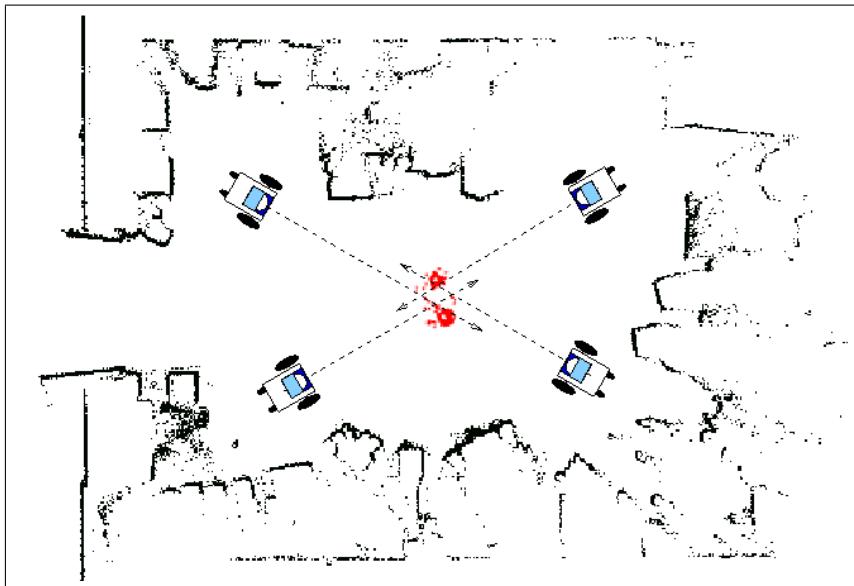


Figure 8.7: 2D map, object (center) and four different vantage points.

landmark measurements that are approximately equally spaced. This computation is performed by stipulating a grid over the scan (in workspace coordinates), and selecting measurements closest to the center points of each grid cell. An all point shortest path search then associates remaining measurements with landmark measurements. The optimization is first performed for the thinned scan; after the landmark scans are localized (and the corresponding coordinate transformation are computed), the remaining measurements are optimized locally, in groups corresponding to individual landmark measurements. Smoothness is attained by using multiple landmark measurements as boundary conditions in this optimization. Figure 8.6 shows an examples of a thinned graph, for which the optimization can be carried out in seconds.

8.4 Setup and Experimental Results

The approach was implemented to acquire 3D models of non-stationary objects using a mobile robot. We first acquired a map of the environment (see Figure 8.7) using the incremental scan matching technique for 2D map building. Non-stationary objects were detected through differencing of scans, using the robot's localization routines to get a rough estimate of this pose. Figure 8.3 illustrates the segmentation process. Red scans are retained while the black scans are assumed to correspond to the background and are henceforth discarded.

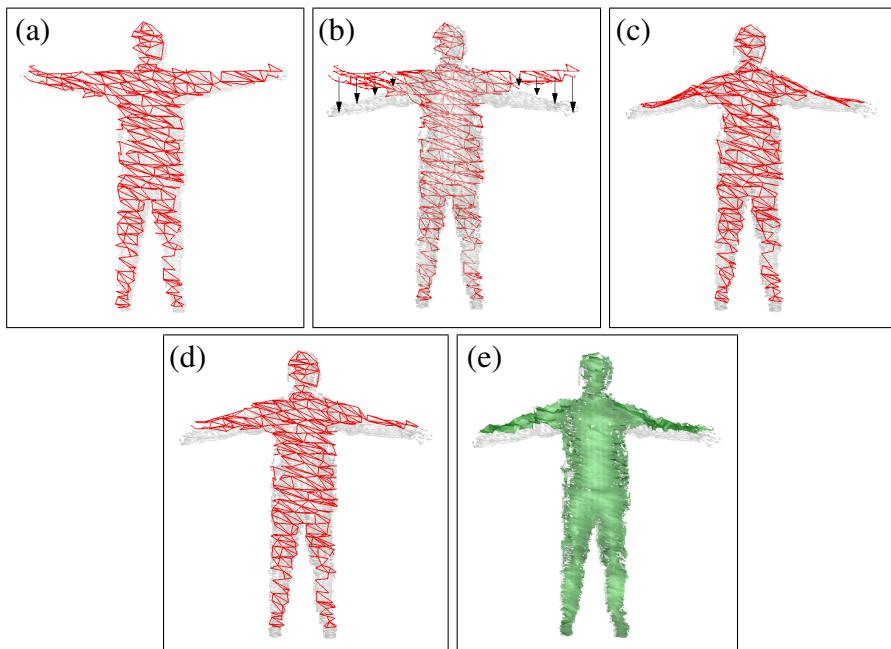


Figure 8.8: (a) Thinning, (b) nearest neighbor search, (c) optimization (first iteration), (d) optimization (second iteration), (e) optimization of remaining nodes.

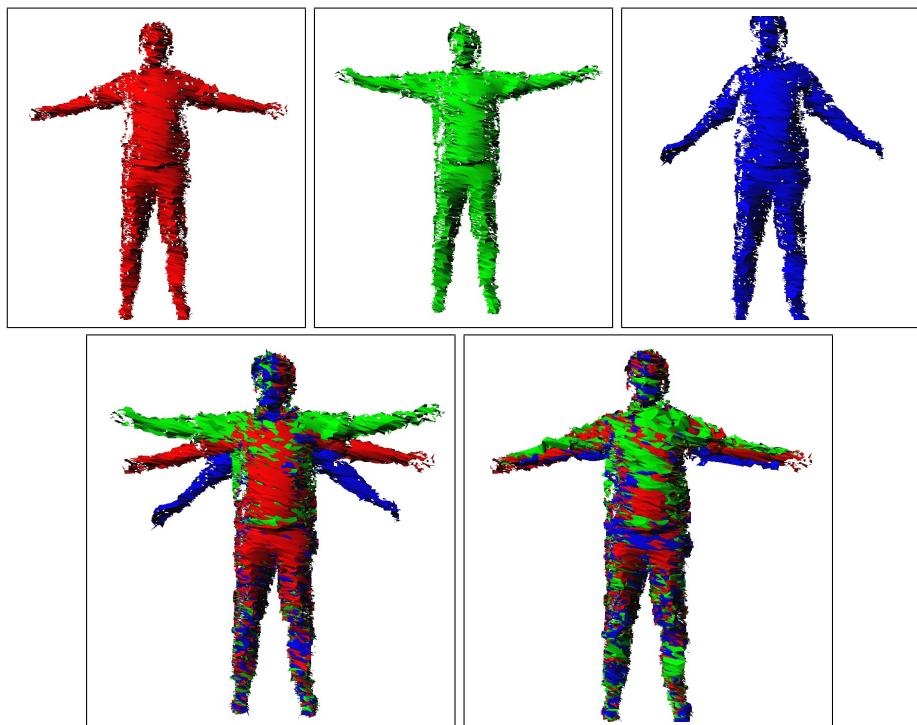


Figure 8.9: Optimal scan registration with rigid object assumption (lower left), corresponding result for our approach (lower right).

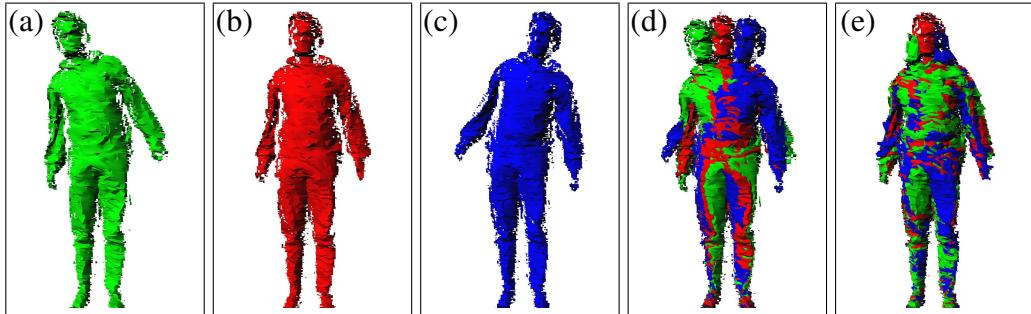


Figure 8.10: (a-c) three scans, (d) results of optimal scan registration and (e) our approach.

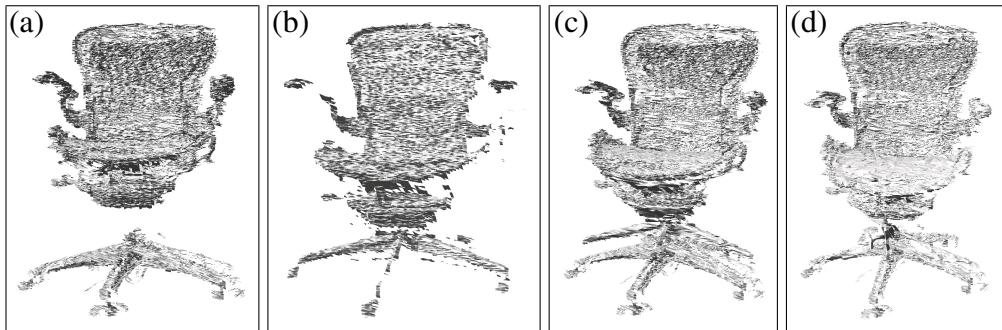


Figure 8.11: Adjustable chair: Scans of the chair in high position (a), scan from the chair in low position (b), the resulting model of the chair with multiple feet (c), and the resulting model of the chair with the transformed scan (b).

Figure 8.8 illustrates one iteration of the algorithm in all essential steps, using data acquired by the robot. Results for matching three scans with different postures are shown in Figure 8.9. While the standard registration procedure leads to a model with six arms, our approach correctly deforms the scan to arrive at an improved model, with two arms. A similar result is shown in Figure 8.10(a-c), which shows three raw scans on the left, followed by the result of (rigid) scan registration and the result of this approach. Another example is the chair in Figure 8.11(a-d) scanned in different heights. The standard registration will lead to multiple feet, this approach correctly aligns them. Table 8.1 shows the cumulative distance between points in the nearest neighbor calculation. The value marked as “start” is the result of an initial registration phase, reflecting the remaining distances under the rigid body assumption. All other columns correspond to further iterations of our algorithm as it adjusts the shape of the scans. This result illustrates numerically the integrity of the result is indeed improved by iterate the process.

	start	loop #1	loop #2	loop #3
<i>moving arms</i>				
scan 1	2.3266	0.8993	0.7986	-
scan 2	2.5320	0.8704	0.8001	-
<i>stretched body</i>				
scan 1	1.9369	1.2915	1.2008	1.1975
scan 2	2.5087	1.2964	1.2220	1.2120

Table 8.1: Average distance to the closest points to the matched model *after* scan registration. The decrease of this distance measures the improvement of the model through local surface deformations.

8.5 Summary

This chapter proposed a technique for simultaneous scan registration and scan deformation for modeling nonrigid objects. The deformation was made possible through the definition of (soft) links between neighboring scan points, whose configuration was calculated during registration. To tackle the resulting optimization problem efficiently, we described a hierarchical optimization techniques that operated on thinned graphs. Experimental results obtained using a mobile robot illustrated the viability of this approach.

There are many problems in object modeling that this chapter does not address, but whose inclusion shall be the subject of future research. For example, the present segmentation approach is somewhat simplistic: It will fail if more than one non-stationary object appears in the scene. The approach requires deformations to be small, and the target object may not move very far during acquisition. Objects are not sub-segmented. This will cause difficulties when components of objects are adjacent to different other components, or missing entirely, which can happen if components are combined and cannot be separated to find the corresponding part. A final direction of future research involves the integration into advanced techniques for between-object data association when modeling multiple objects in non-stationary environments [Anguelov *et al.*, 2002].

CHAPTER 9

Mapping Applications with Autonomous Robots

In this chapter we present two applications of systems for mapping with mobile robots. The first application is the robotic system, designed to autonomously explore and acquire 3D maps of abandoned mines, is a 1,500 pound vehicle, nicknamed “Groundhog”. Groundhog acquires large-scale consistent maps of the voids it explores. The second application shows the feasibility of acquiring high-resolution ground models using active laser range sensors on a low-flying helicopter platform. Low-flying air vehicles, such as helicopters, promise to overcome limitations which are inherent to ground-based vehicles: they are much less constraint than ground vehicles with regards to their navigational capabilities and can fly low enough to acquire data of vertical structures at high resolution.

9.1 Mine Mapping

9.1.1 Introduction

In recent years, the quest to find and explore new, unexplored terrain has led to the deployment of more and more sophisticated robotic systems, designed to traverse increasingly remote locations. Robotic systems have successfully explored volcanoes [Bares and Wettergreen, 1999], searched meteorites in Antarctica [Apostolopoulos *et al.*, 2001, Urmson *et al.*, 2001], traversed deserts [Bapna *et al.*, 1998], explored and mapped the sea bed [Durrant-Whyte *et al.*, 2001], even explored other planets [Matthies *et al.*, 1995]. This section presents a robot system designed to explore spaces much closer to us: abandoned underground mines.



Figure 9.1: The Groundhog robot is a 1,500 pound custom-built vehicle equipped with on-board computing, laser range sensing, gas and sinkage sensors, and video recording equipment.

According to a recent survey [Belwood and Waugh, 1991], “tens of thousands, perhaps even hundreds of thousands, of abandoned mines exist today in the United States. Not even the U.S. Bureau of Mines knows the exact number, because federal recording of mining claims was not required until 1976”. We are unaware of the location of many mines - despite the fact that most mines were built just a few generations ago. A recent near-fatal accident in Somerset, PA, speaks to this end: When miners in their routine work accidentally breached a nearby abandoned mine, fifty million gallons of water poured upon them, cutting off nine miners and almost burying them alive. The cause of this accident was officially determined to be a lack of accurate mine maps; the breached and flooded mine had been suspected to be several hundred feet away [Pauley *et al.*, 2002].

Even if accurate mine maps exist, those are usually just idealized 2D drawings. Little can be inferred from such sketches with regards to critical measures, such as the volume and the structural soundness of an abandoned mine. Accurate models of such abandoned mines would be of great relevance to a number of problems that directly affect the people who live or work near them. One is subsidence: structural shifts can cause collapse on the surface above. Ground water contamination is another problem of great importance, and knowing the location, volume,



Figure 9.2: In this configuration Groundhog is equipped with a tiltable SICK LMS laser range finders on either end. Groundhog is essentially symmetrical, enabling it to retract without having to turn around.

and condition of an abandoned mine can be highly informative in planning and performing interventions. Accurate volumetric maps are also of great commercial interest. Knowing the volume of the material already removed from a mine is of critical interest when assessing the profitability of re-mining a previously mined mine.

Abandoned mines are usually not accessible to people. Lack of structural soundness is one reason; another is the harshness of the environment (e.g., low oxygen levels, flooding) and the danger of explosion of methane, a gas that frequently accumulates when mines are no longer ventilated. This makes mine a superb target domain for autonomous robots. However, mapping a mine with a robotic vehicle is a challenge. The vehicle must be rugged enough to survive the harsh environmental conditions inside the mine. It must be able to perceive and negotiate major obstacles. It must navigate truly autonomously as there is no communication to an outside operator.

9.1.2 Robotic System for Mine Mapping

The robotic system designed to autonomously explore and acquire 3D maps of abandoned mines is a 1,500 pound vehicle, nicknamed “Groundhog” and shown in Figure 9.1; a detailed description of the hardware can be found in [Baker *et al.*, 2003]. Groundhog is essentially built out of the front halves of two ATVs, endowing it with identical steering mechanisms on either end. While the exact configuration of the robot varied from experiment to experiment, in its final configuration Groundhog was essentially symmetrical, enabling it to retract without

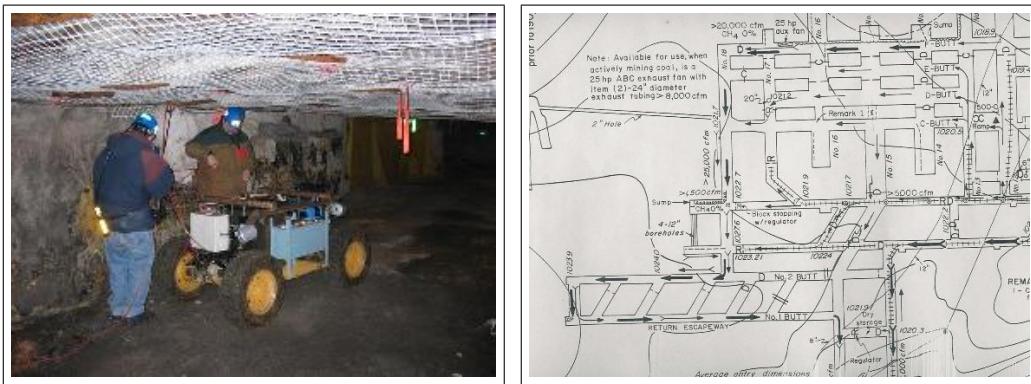


Figure 9.3: Groundhog in the Bruceton Research Mine. Since 1940 the mine had been maintained in a state safe for people to enter. Currently it will be used mainly for mine research and rescue. On the right side you can see the part of the available mine map.



Figure 9.4: Groundhog in the Florence Mine near Pittsburgh, PA. This abandoned mine is inaccessible to people and was mostly drained, leaving behind acidic mud. In this experiment the robot was tethered and under remote control.

having to turn around.

For acquiring 3D maps, Groundhog is equipped with tiltable SICK laser range finders on either end (see Figure 9.2). It also carries two mine-certified portable gas detectors to enable it to detect methane and other combustible gases. To navigate, Groundhog analyzes local 3D scans with regards to traversability. A fast C-space planner determines whether the terrain ahead can be negotiated, and if so, identifies suitable paths (more details about this topic can be found in [Ferguson *et al.*, 2003]). Those are then executed via PD control, using fast 2D scan matching to keep the vehicle localized. Failure to find a suitable path leads Groundhog to retract in reverse motion. Groundhog also acquires large-scale consistent maps of the voids it explores.

Groundhog's development began in the Fall of 2002. Approximately a dozen



Figure 9.5: Both entrances of the Mathies Mine, Pittsburgh PA. The Mathies mine was considered to be a "wet"mine. Some areas along the haulage track were always damp or wet. Mathies Mine was closed in April 2002.

test runs were carried out in a well-maintained inactive coal mine accessible to people: the Bruceton Research Mine located near Pittsburgh, PA. Mining was discontinued in the early 1940s, but since that time the mine had been maintained in a state safe for people to enter (see Figure 9.3). The mine features hallways several hundred meters long, putting to a test the vehicle's physical endurance and its ability to manage large amounts of data. However, this mine is technically not abandoned and therefore not subject to collapse and deterioration. On October 27, 2002, Groundhog descended for the first time into an abandoned mine inaccessible to people. This mine, the Florence Mine near Pittsburgh, PA, had been abandoned and flooded for many decades. Before the robot's entry, the mine was mostly drained, leaving behind acidic mud that miners refer to as "yellow boy". Figure 9.4 depicts the vehicle after descending approximately 30 meters into the mine, here operating on a tether and under remote control. On May 30, 2003, after a long series of test runs carried out in the Bruceton Research Mine, Groundhog finally entered an inaccessible abandoned mine in fully autonomous mode. The mine is known as the Mathies mine and is located in the same geographic area as the other mines. The core of this surface-accessible mine consists of two 1.5-kilometer long corridors which branch into numerous side corridors, and which are accessible at both ends . This was an important feature of this mine, as it provided natural ventilation and thereby reduced the chances of encountering combustible gases inside the mine. Figure 9.5 depicts both ends of the mine. A map of the mine, provided to us by the Mine Safety and Health Administration and the mine owner, is shown in Figure 9.6; apparently this is the most accurate map on record for this mine. To acquire a more accurate map of one of the main corridors, the robot was programmed to autonomously navigate through the corridor. 308 meters into the mine, the robot encountered a broken ceiling bar draping diagonally across its path. The robot made the correct decision to retract.

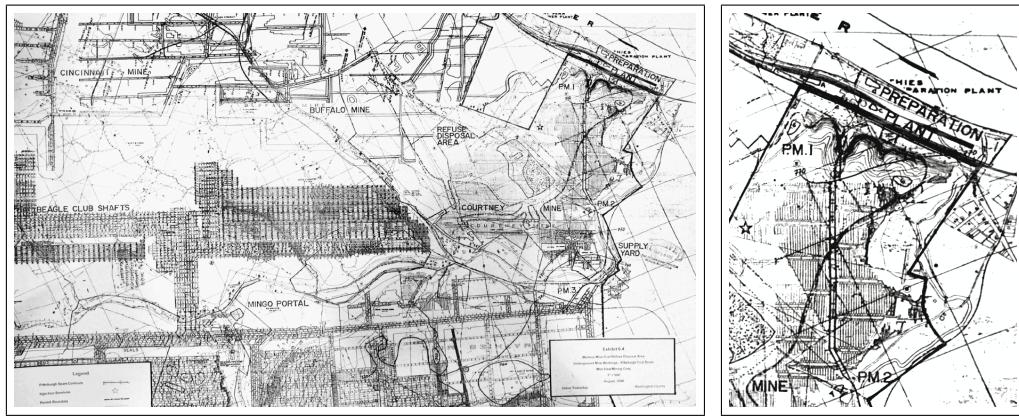


Figure 9.6: Available map of the Mathies Mine. On the right side you can see the magnified section with the two long corridors and entrances on both sides. In the upper part of the corridors you can find an inconsistency in the map (the two corridors are shifted to the left), which is an error in the map.

9.1.3 SLAM Technique in Mines

The core of the Groundhog navigation system is comprised of a software package that solves the SLAM problem by acquiring 2D maps. On the surface, mobile robots can often utilize GPS to acquire absolute position information. Underground, we do not have the luxury of GPS localization. At the lowest level of processing, Groundhog's mapping system utilizes a real-time scan matching technique for registering consecutive scans (as described in Chapter 2). Scans are acquired using laser range finder pointed forward. In our implementation, all calculations are carried out in real-time. The scan matcher enables Groundhog to recover two quantities: locally consistent maps and an estimate of the robot's own motion.

9.1.4 Mine Mapping Experiments

Groundhog was tested in a number of experiments, some taking place in laboratory settings, others in actual mines. As described in the introduction to this section, Groundhog navigated and mapped three different mines, all with vastly different characteristics. The Bruceton research mine enabled us to perform large-scale experiments, testing the vehicle's endurance and ability to acquire large mine maps with many cycles. However, our tests in this mine focused on the ability to acquire large-scale maps, not on autonomous navigation. The Florence mine enabled us, for the first time, to acquire a 3D map of an environment truly inaccessible to people. The fact that it was partially flooded limited the operational range



Figure 9.7: 2D map of the Florence mine (left image) and 3D structure made by the upwards pointed laser (right image).



Figure 9.8: Camera images recorded while autonomously exploring the Mathies mine. The imagery helps to detect the condition of the mine and shows the degree of deterioration.

of the robot. So far, our only autonomous run was performed in the Mathies mine, where the robot operated partially outside the range of our wireless communication link. Figure 9.7 (left) depicts the map of the Florence mine (see Figure 9.4). Here the robot's configuration involved a forward-pointed laser for 2D mapping, and an upwards pointed laser to map the ceiling structure; no sensor was available to map the ground. While this configuration is insufficient for autonomous navigation, it has the nice advantage that the 3D map (see the right image in Figure 9.7) can be constructed easily from the 2D map as the robot moves [Thrun *et al.*, 2000].

Groundhog entered the Mathies mine autonomously on May 30, 2003 at 10:55 AM, EST. Shortly thereafter, it lost radio contact with the ground station. One hour and 308m into the mine (see red line in the uppermost image of Figure 9.9), the robot encountered a roof-fall, including a steel support beam that draped diagonally across the corridor and blocked further progress. The machine made the appropriate decision to retract and begun to back out of the mine at approximately 12:00 PM (blue dashed line in the uppermost image of Figure 9.9), but encountered software difficulties starting at approximately 12:20 PM. After an-

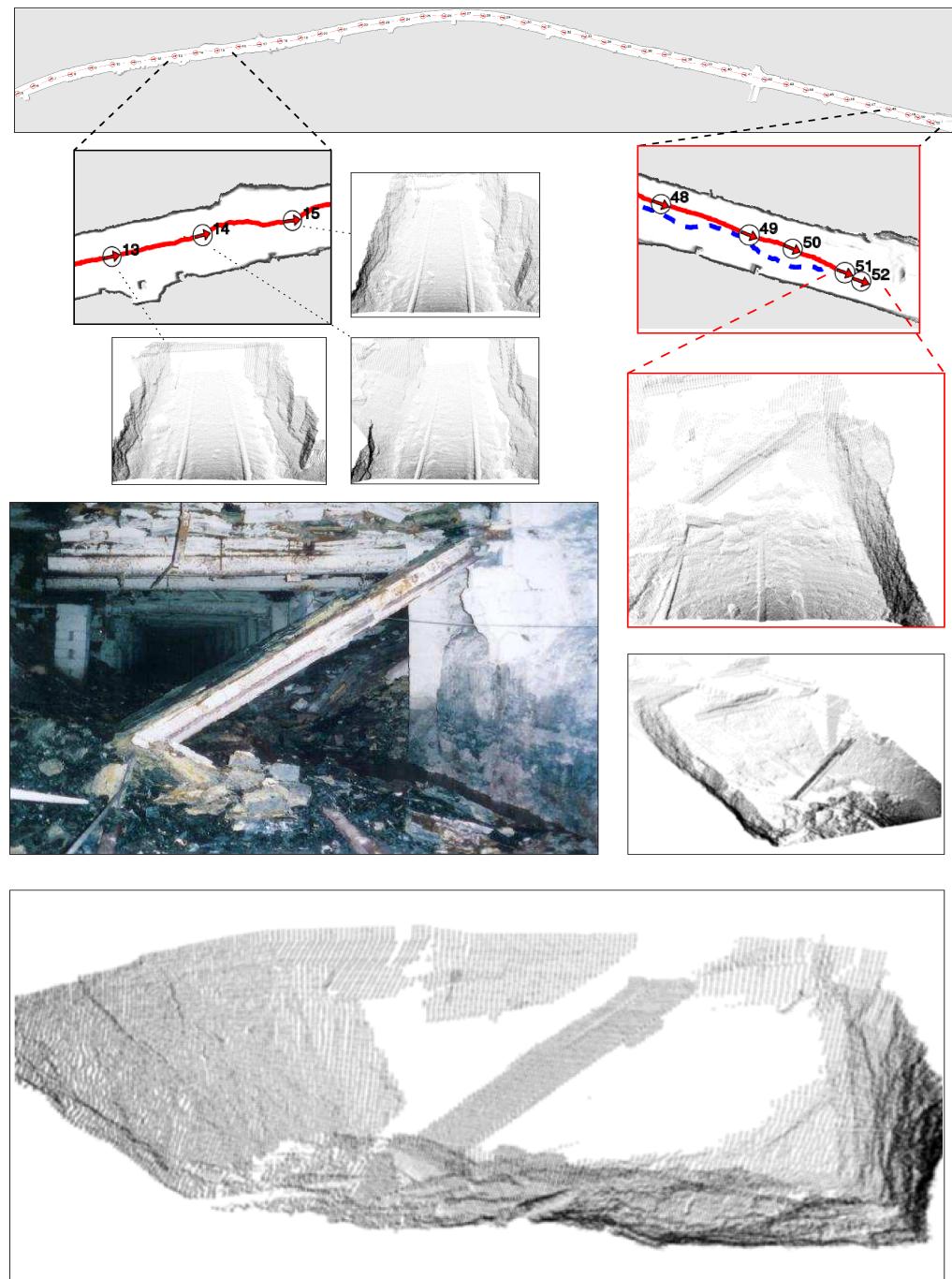


Figure 9.9: The uppermost image shows the resulting 2D map of the Mathies mine. The vantage points at which the robot chose to acquire a 3D scan are marked with small circles. The left upper images show three local 3D scans. On the right side you can see the 3D scans with the un-navigable obstacle and the left middle image shows the camera picture of the corresponding part of the mine.

other 30 minutes, the system had not resolved its problems, and it was decided to try to intervene over the weak wireless link at 12:56 PM. Under the strain of tele-operation, the wireless link locked up shortly thereafter, stranding the robot an estimated 200 meters inside the mine at 1:04 PM (yellow circle in the uppermost image of Figure 9.9). Subsequent efforts to re-establish the link failed, and at 3:30 PM, two mine safety inspectors received permission to suit up and proceed into the mine to try to manually reset Groundhog's wireless link. The link was successfully reestablished at 3:50 PM and the robot exited the mine under manual control at 4:02 PM. Figure 9.8 depicts imagery acquired inside the Mathies mine. These images were recorded with a low-light camera, using the robot's active IR light source for illumination. The 2D map of the Mathies mine is shown in Figure 9.9. This 308 meter-long map shows the obstruction on its right end. In 2D, the obstruction appears to be small and navigable. In 3D, however, it becomes apparent that the obstacle is not navigable. The robot's 3D map of the situation is shown in the left and lower part of Figure 9.9, along with the image. The resulting 2D map and the corresponding 3D maps were found to provided an unprecedented glimpse into the interior of this quickly deteriorating environment. A subsequent debrief with members of mine safety and environmental protection agencies confirmed that the level of detail provided by these models opens up unprecedented opportunities to understand the situation inside an abandoned mine, and to target corrective actions.

9.1.5 Summary

The system has been tested under extreme conditions, and generated accurate maps of abandoned mines that are inaccessible to people. Our research demonstrates that the autonomous acquisition of maps of abandoned mines is indeed feasible with autonomous robotic systems. Our extensive experimentation with the Groundhog system suggests a number of opportunities for further research. Chief among them is to develop systems that can autonomously map entire mines, not just fractions thereof. Difficulties in this task arise from the fact that side corridors were frequently closed before miners abandoned them, to stop the flow of gases from inactive into the active parts of a mine. Such closures pose insurmountable obstacles to our present system, but might be surmountable given appropriate means of environment modification. A second limitation of the present system is its inability to negotiate water and heavy mud. A good fraction of mines in the U.S. is flooded. This creates an opportunity to build submersible mine mapping robots, which would have the advantage of not being forced to the ground of a mine. Another possibility would be an amphibious vehicle for exploring partially flooded mines. Finally, being able to communicate with a robot while inside a mine would have great operational benefits, both with regards to trouble shoot-



Figure 9.10: Photo of the “Groundhog”-team, the mine owner of the Mathies mine, and the mine safety inspectors after the experiment.

ing and for assisting the robot in its exploration decisions. At present, there are only low-bandwidth technologies for communicating directly through solid matter. Establishing a network of wireless repeater stations, as proposed in [Nguyen *et al.*, 2002, Nguyen *et al.*, 2003], would be a viable extension to mine mapping robots, which could critically enhance the operational capabilities of future mine-exploring robots. Despite these limitations, Groundhog’s success in exploring and mapping abandoned mines opens a world of opportunities for subterranean robotic exploration. While the surface of the Planet has been mapped with great detail, most underground voids lack accurate maps, often to the detriment of the people who live or work nearby. This applies not just to man-made voids, such as mines. It equally applies to natural voids such as caves.

9.2 Helicopter Mapping



Figure 9.11: Instrumented helicopter platform: The system is based on the Bergen Industrial Twin, with a modified SICK LMS laser range finder, a Crossbow IMU, a Honeywell 3D compass, a Garmin GPS, and a Nikon D100 digital camera. The system is equipped with on-board data collection and processing capabilities and a wireless digital link to the ground station.

9.2.1 Introduction

Since sensing is usually confined to the immediate vicinity of the vehicle, active sensors such as sonars and laser range finders have become the technology of choice [Thrun, 2002] – albeit some notable exception using passive cameras [Murray and Little, 2001]. For the problem of acquiring accurate maps of outdoor terrain, ground vehicles are limited in two aspects: First, the ground has to be traversable by the vehicle itself. Many environments are cluttered with obstacles that are difficult to negotiate [Casper, 2002]. Second, all important features in the environment have to be perceivable from relatively low vantage points. Moreover, the set of vantage points that can be attained usually lie on an approximate 2D manifold parallel to the ground surface, since most ground vehicle cannot vary the height of their sensors. This is a severe limitation that is particularly troublesome in complex, natural terrain.



Figure 9.12: Some of the electronics on-board the helicopter: An Intel Stayton board with a 400Mhz XScale processor interfaces to the SICK LMS laser via a high speed RS422 serial link, and to all other devices (compass, GPS, IMU) via RS232. The communication to the ground is established via a 802.11b wireless link.

In complimentary research, there exists a huge body of literature on high aerial and satellite-based mapping (see e.g., [Becker and Bove, 1995, Debevec *et al.*, 1996, Konecny, 2002]. At high altitude, it is usually impossible to deploy active range sensors; instead, these techniques are commonly based on passive computer vision systems. While traversability is not an issue for high aerial vehicles, the relatively high vantage points makes it difficult to map vertical structures, and it limits the resolution at which maps can be acquired. Furthermore, clouds can cause obstruction or cast shadows in the imagery. And while air vehicles can change altitude and are therefore not subject to the 2D manifold constraint characteristic of ground vehicles, such changes have next-to-zero effect on the visual appearance of the surface structure.

Low-flying air vehicles, such as helicopters, promise to overcome these limitations: they are much less constraint than ground vehicles with regards to their navigational capabilities, yet they can fly low enough to acquire data of vertical structures at high resolution. A seminal system by Miller *et al.* [Miller and Amidi, 1998, Miller, 2002] has demonstrated the feasibility of acquiring high-resolution ground models using active laser range sensors on a low-flying helicopter platform.

This section describes a similar system for acquiring high-resolution 3D models of urban and ground structures. The system, shown in Figure 9.11 and Figure 9.13, is based on a Bergen Industrial Twin helicopter, equipped with a 2D

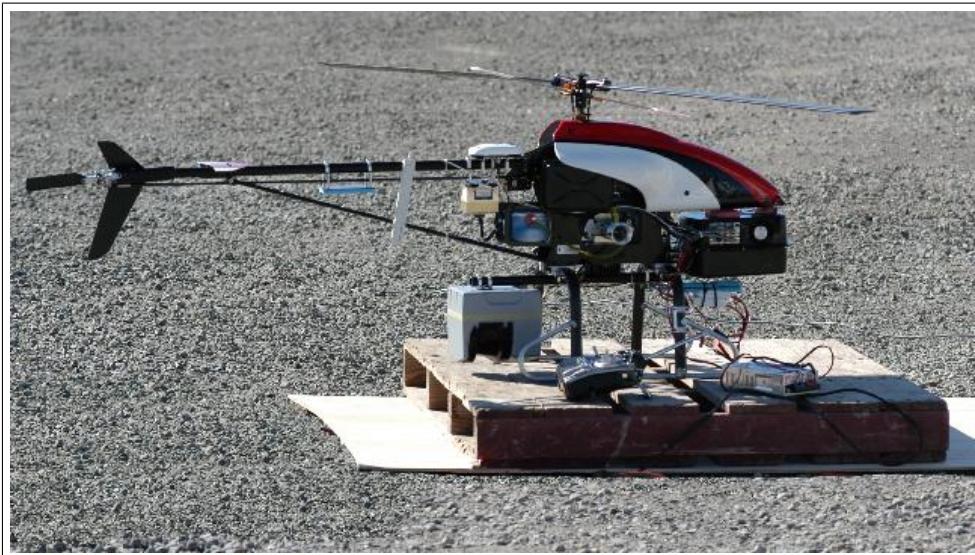


Figure 9.13: Improved version of the helicopter platform for the autonomous flight: lighter version of the modified SICK LMS laser range finder, differential GPS, and a PC/104 computer as a faster replacement for the Intel Stayton board.

SICK range finder and a suite of other sensors for position estimation. Figure 9.12 shows some of the computer equipment mounted on the vehicle of Figure 9.11. The 2D range finder provides the vehicle with 2D range slices at a rate of 75Hz, oriented roughly perpendicular to the robot's flight direction. The helicopter is flown under manual control (see Figure 9.14) as well as completely autonomous (see Figure 9.17). We will not address the problem of autonomous helicopter flying and describe only the used mapping technique. The system, necessary for the autonomous flight is developed by Andrew Y. Ng [Ng *et al.*, 2003]) and was successfully used in some of the experiments.

Building 3D maps with centimeter resolution is difficult primarily of two reasons:

1. Using GPS and other proprioceptive sensors, the location and of the sensor platform can only be determined up to several centimeters accuracy. Similar limitations apply to the estimation of its angular orientation. The ground position error induced by those angular errors are often in the order of several meters. This is illustrated by the ground model shown in Figure 9.15. Shown there is a 3D model reconstructed using the vehicle's best estimates of its position and orientation. These limitations mandate the use of the range finder as a means for further reducing the errors in the vehicle pose estimate, and to jointly solve the environment mapping the vehicle localization problem.



Figure 9.14: Helicopter flying by a building under manual remote control. The image also shows the pilot.

2. The rich literature on simultaneous localization and mapping, or SLAM (as we have described in Chapter 5 or Chapter 6), has developed extensive techniques for simultaneously estimating the structure of the environment (3D model, or map) and the pose of the vehicle. Range scan alignment techniques (applied in 2D in robotics and in 3D in object modeling) overcome this problem by cross-registering multiple scans. To do so, these techniques rely on multiple sightings of the same environmental feature. This is not the case for a 2D sensor that is moved through the environment in a direction perpendicular to its own perceptive field: Here consecutive measurements always correspond to different things in the world.

We have developed a probabilistic SLAM algorithm that addresses both of these problems. Our approach acquires 3D models from 2D scan data, GPS, IMU, and compass measurements. The algorithm exploits a local smoothness

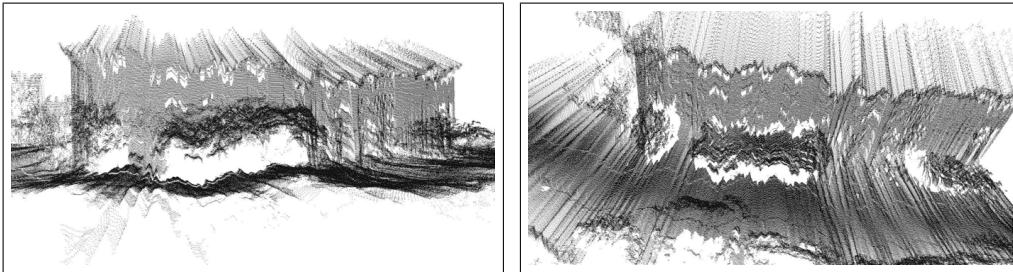


Figure 9.15: Raw data of a multi-storey building

assumption for the surface that is being modeled, but simultaneously allows for the possibility of large discontinuities of the mapped structure. By doing so, it can utilize range scans for vehicle localization, and thereby simultaneously improve both the pose estimate of the helicopter and the accuracy of the resulting 3D environment model. We believe that the maps acquired by our system are significantly more accurate and spatially consistent than previous maps acquired by helicopter systems. A key reason of this increase in accuracy comes from the fact that scans are used for the pose estimate of the vehicle’s sensor platform.

9.2.2 3D Modeling Approach

This section describes a variant of the previously described probabilistic matching for scan registration (see Chapter 2). The problem of scan registration can be formulated as the problem of recovering the vantage points from which the scans were taken. In our approach, scans are taken from a helicopter system; hence each vantage point is described by six variables: its x_t *pose* of the sensor’s local coordinate system at time t , relative to a global coordinate system of the 3D model, in three Cartesian coordinates and their three Euler angles ϕ_t (roll), θ_t (pitch), and ψ_t (yaw). In irregular intervals, we receive GPS and compass measurements for the pose, denoted by y_t .

Similar to Equation 2.19 in Section 2.3.1, we calculate a sequence of poses $\hat{x}_1, \hat{x}_2, \dots$ and corresponding maps by maximizing the marginal likelihood of the t -th pose and map relative to the $(t - 1)$ -th pose and map:

$$\hat{x}_t = \underset{x_t}{\operatorname{argmax}} \{ p(z_t | x_t, \hat{m}(\hat{x}_{1:t-1}, z_{1:t-1})) \cdot p(x_t | y_t, \hat{x}_{t-1}) \} \quad (9.1)$$

In this equation the term $p(z_t | x_t, \hat{m}(\hat{x}_{1:t-1}, z_{1:t-1}))$ is the probability of the most recent measurement z_t given the pose x_t and the map $\hat{m}(\hat{x}_{1:t-1}, z_{1:t-1})$ constructed so far. The term $p(x_t | u_t, \hat{x}_{t-1})$ represents the probability that the robot is at location x_t given the robot previously was at position \hat{x}_{t-1} and has measured the GPS and compass position y_t .

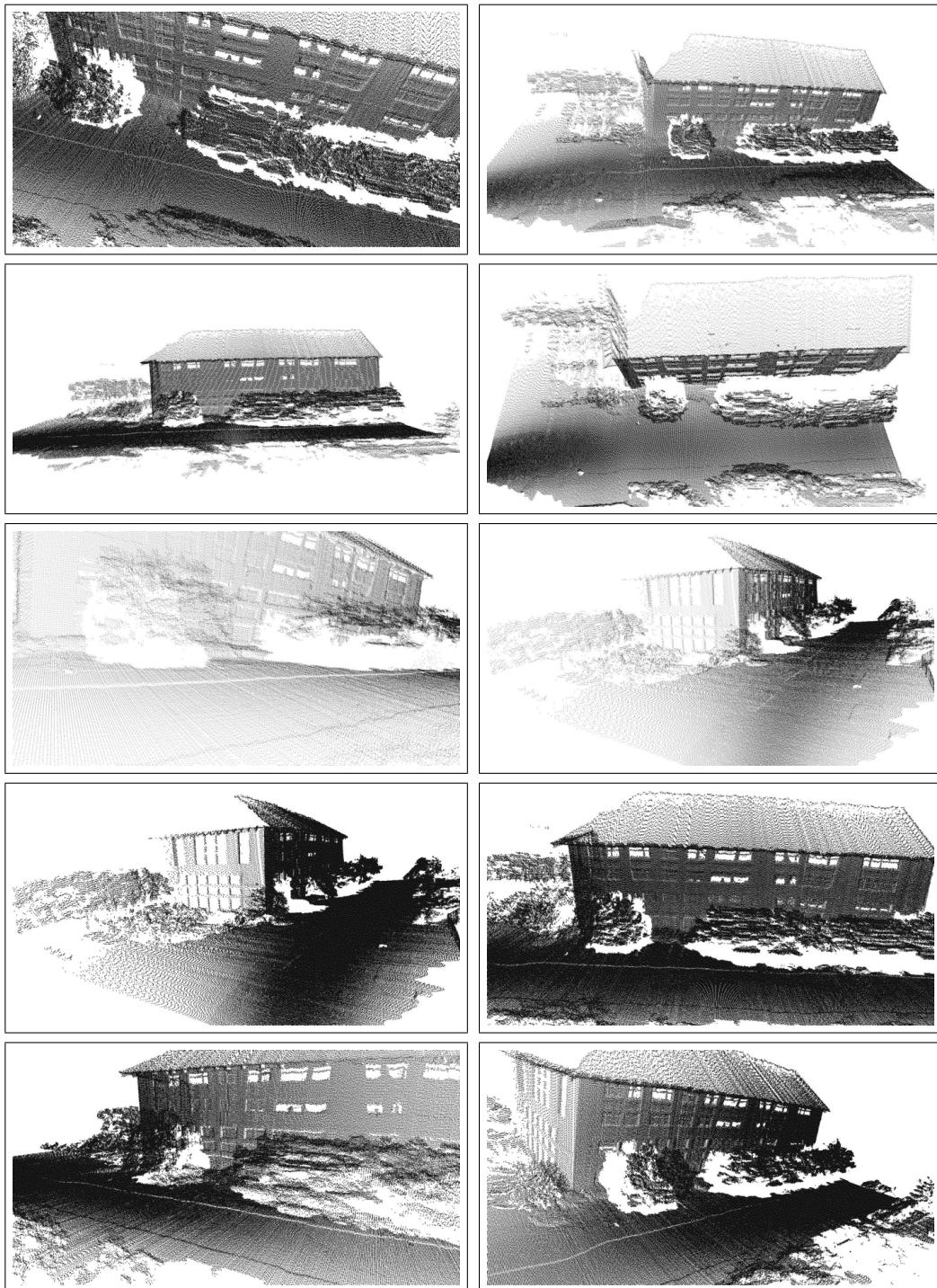


Figure 9.16: Snapshots of the 3D Map of the building shown in Figure 9.14. The map is represented as a VRML file and can be displayed using standards visualization tools.



Figure 9.17: Helicopter flying autonomous over a building at NASA DART.

The resulting pose \hat{x}_t is then used to generate a new map \hat{m} :

$$\hat{m}(\hat{x}_{1:t}, z_{1:t}) = \underset{m}{\operatorname{argmax}} p(m \mid \hat{x}_{1:t}, z_{1:t}) \quad (9.2)$$

Such a generative model is the most general approach to robotic mapping; however, it involves as many variables as there are features in the map m ; thus the resulting likelihood functions would be difficult to optimize in real-time.

For a forward-flying helicopter which never traverses the same location twice, it is sufficient to model a relative probability of subsequent measurements conditioned on the pose. Therefore, measurements obtained at previous places provide no information to maximize (9.1). In our system, we therefore use a map

$$\hat{m}(\hat{x}_{t-\Delta t:t-1}, z_{t-\Delta t:t-1}) = \hat{m}(\hat{x}_{t-\Delta t}, \dots, \hat{x}_{t-1}, z_{t-\Delta t}, \dots, z_{t-1}) \quad (9.3)$$

that is constructed based of the Δt most recent measurements only.

The overall approach can be summarized as follows: At any point $t - 1$ in time the robot is given an estimate of its pose \hat{x}_{t-1} and a map $\hat{m}(\hat{x}_{t-\Delta t:t-1}, z_{t-\Delta t:t-1})$. After the robot moved further on and after taking a new measurement z_t , the robot determines the most likely new pose \hat{x}_t . It does this by trading off the consistency of the measurement with the map and the consistency of the new pose with the GPS / compass position and the previous pose.

If we reduce the optimization space by one dimension (using a fixed value for the pitch of the sensor), both steps can be carried out highly efficiently. The

system is then similar to our previously used scan matching technique. Theoretical we can use all dimensions for the optimization, but due to the real-time limitation we have to simplify the problem here.

The result is an algorithm that implements the optimization in an incremental fashion, in which the pose at time t is calculated from the pose at time $t - 1$ under incorporation of all scan measurements. While such an implementation is subject to local minima, it can be performed in real-time and works well, as long as the helicopter never traverses the same area twice. The 3D model is also obtained in real-time, by using the corrected pose estimates to project measurements into 3D space. The model may simply be represented by a collection of scan points in 3D, or a list of polygons defined through sets of nearby scan points. Both are computed in real-time.

9.2.3 Experimental Results

We have tested our approach in a number of different environments, all of which involved significant vertical structure that cannot easily be mapped by high-aerial vehicles. The first experiment was taken under manual control; Figure 9.14 depicts the helicopter flying by a multi-storey building; it also depicts the pilot walking behind the vehicle. The raw data acquired in this flight is shown in Figure 9.15; this plot uses the helicopter's best estimate of its own pose for generating the map. These plots clearly show significant error, caused by a lack of accurate pose estimation.

Figure 9.18 depicts a sequence of maps as they are generated in real-time. The latency between the data acquisition and the display on the ground is less than half a second. Snapshots of the final 3D map are shown in Figure 9.16.

In the second experiment the helicopter was flying completely autonomous over the training facility of the Disaster Assistance and Rescue Team (DART) site at the NASA's Ames Research Center at Moffett Field, CA, USA (see Figure 9.17 and 9.19). The flight had a checkerboard-pattern, so that the sensor covered some regions multiple times. You can see the trajectory in the lower images of Figure 9.18. Although this, the used optimization technique considered only a short history of taken scans. This was necessary as the required time would prevent the system to work in real-time. Therefore the previous seen parts of the building was not taken into account for generating the complete model. The resulting model can be seen in the bottom image of Figure 9.19.

Unfortunately, we do not possess ground truth information for the mapped structures. This makes it impossible to assess the accuracy of the resulting models. However, the models appear to be visually accurate, locally consistent. The spatial resolution of these models in the centimeter range.

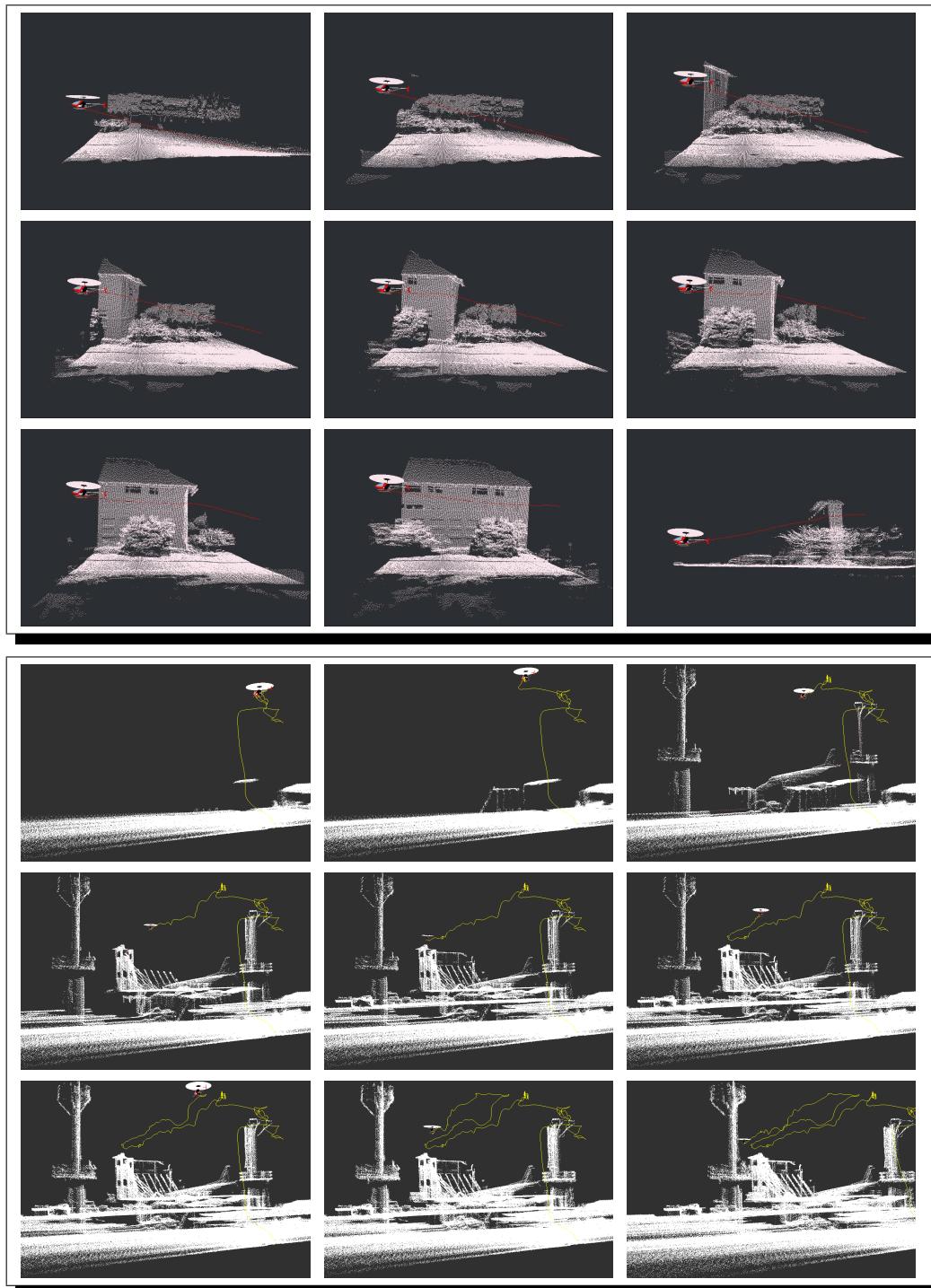


Figure 9.18: Visualization of the mapping process, carried out in real-time. This figure shows a sequence of snapshots taken from the interactive ground display, which displays the most recent scans with less than 0.5 seconds latency.



Figure 9.19: Photos of NASA’s Disaster Assistance and Rescue Team training facility (DART). The lower image shows the resulting model build from the collected helicopter data.

9.2.4 Summary

This section described initial results for an instrumented helicopter platform for 3D ground modeling. A real-time algorithm was developed that integrates pose estimates from multiple sensors with range data, acquired by a 2D laser range finders oriented perpendicular to the vehicle's flight direction. The algorithm uses a fast optimization technique to generate maps in real-time. Relative to prior work in [Miller and Amidi, 1998, Miller, 2002] our approach utilizes the range data for vehicle localization, which results in maps that are spatially significantly more consistent and – we suspect – more accurate. Experimental results suggest that the maps acquired by our approach are of unprecedented detail and accuracy; however, the exact accuracy is presently not known. Nevertheless, we believe that the findings in this work are highly promising.

It is important to notice that we does not address the topic of autonomous helicopter control, see [Bagnell and Schneider, 2001, Coleman *et al.*, 2002, Dittrich and Johnson, 2002, Saripalli *et al.*, 2002] for recent work in this area. However, an integration of accurate mapping and autonomous flight would make it possible to operate autonomous helicopters in rugged terrain, such as mountainous areas or caves. It would also open the door to the important problem of selecting safe landing pads in uneven terrain.

A second extension concerns the limitation that surfaces are only sensed once. This is also the case in the experiment carried out at the NASA DART site as the mapping process used always a small part of the data. Although the helicopter has seen parts of building multiple times, the previous data was not integrated. To enable the vehicle to integrate data from multiple fly-overs would require a mechanism for establishing correspondence to previously mapped terrain. While a number of techniques exist that offer this capability [Thrun, 2002], it is unclear whether they can be executed in real-time on a helicopter platform.

CHAPTER 10

Conclusions

The mapping problem is generally regarded as one of the most important problems in the pursuit of building truly autonomous mobile robots and also as one of the most difficult perceptual problem in robotics. This thesis presented approaches to solve several fundamental sub-problems in map building with mobile robots.

Most existing approaches assume that the environment is static during data acquisition. In practice this is often not the case. Public spaces, for example, cannot be closed for all people to avoid measurements caused by dynamic elements. A lot of state-of-the-art approaches can deal with a certain amount of spurious measurements, but they typically do not distinguish between static and dynamic objects of the environment. In this thesis we proposed two different approaches to deal with dynamic elements. Our first algorithm combines a people tracking technique based on a sample-based variant of Probabilistic Data Association Filters with an incremental scan matching algorithm. Practical experiments demonstrated in experiments that this approach can improve the accuracy of the maps in two ways. First, the alignment errors are significantly reduced and second, the amount of spurious objects in the resulting map is seriously reduced. Our method has successfully been applied to two-dimensional and three-dimensional maps built from laser range finder data. We demonstrated that this technique can be carried out online. To relax the assumption to predefined features and corresponding motion models we proposed a second approach which applies the popular EM-algorithm to determine for each laser range beam whether or not it is reflected by a dynamic object. In the expectation step we compute a probabilistic estimate about which measurements might correspond to static objects. In the maximization step we use these estimates to determine the position of the robot and the map. This process is iterated until no further improvement can be achieved. In practical experiments

we demonstrated that this technique is able to reliably filter out dynamic objects.

One of the most popular technique in robotic mapping is incremental scan matching. Due to the fact that this approach accumulates errors over time, it can easily happen that these accumulated errors lead to a global inconsistency. Accumulated errors caused by inaccurate or wrong registrations cannot be corrected due to the incremental nature of this approach. In the robotics community this problem is well-known as the ‘‘loop closing problem.’’ Whenever a robot maps a large cyclic environment, the accumulated error prevents the mapping algorithm from correctly combining the new measurements with the map build so far, when the robot returns to a previously seen region. This becomes even harder under real-time constraints, since they require efficient approaches to maintain the huge hypothesis spaces. There is large community in robotics which applies extended Kalman filters to efficiently solve the SLAM (simultaneous localization and mapping) problem. These approaches, however, assume that we can extract landmarks and that the correspondence (or data association) problem for these landmarks is solved.

In this thesis we presented a highly efficient approach to simultaneous localization and mapping with laser scans, which learns occupancy grid maps and therefore does not require predefined landmarks. As previously proposed by Murphy [Murphy, 1999], our approach applies a Rao-Blackwellized particle filter to estimate a posterior of the path of the robot, in which each particle has associated to it an entire map. To scale to larger environments, we transform sequences of laser range-scans into odometry measurements using our range-scan registration technique. This way our system can deal with significantly larger environments than the original approach, since the scan matching yields odometry estimates that are by one order of magnitude more accurate than the raw wheel encoder data. By using a learned model of the residual errors of the range registration our approach can correctly integrate the improved odometry into the particle filtering process. As a result, we obtain a drastic reduction in the number of particles needed to build large-scale maps, or, put differently, an improved ability to map large environments. This has been validated in a series of experiments carried out in simulation and with real robots. Simultaneously, our approach reduces the particle depletion problem, since the posterior distribution is more peaked so that more particles are close to the correct solution. Particle depletion can seriously affect a filter especially in the context of large loops, since the likelihood of having a particle at the right location reduces with the length of the loop. Without our optimization an intractably large number of particles would be required to avoid the particle depletion.

The EKF solution of the SLAM problem consists of a continuous and a discrete component. The continuous estimation problem pertains to the location of individual features or objects in the environment and the pose of the robot relative

to these objects. The discrete aspect of the SLAM problem is the data association problem, which is the problem of determining whether or not two measurements observed at different points in time correspond to the same object in the physical world. This can be done, for example, by maintaining multiple hypotheses like in the Rao-Blackwellized particle filter approach mentioned before. This solution, however, is proactive since it maintains a huge amount of hypotheses to always have “the right particle.” In this thesis we also presented a novel, non-proactive way to deal with potential data associations in SLAM. In essence, our approach searches the combinatorial tree of possible data association decisions. The search is lazy, which means that only when an alternative assignment indicates promise it will be evaluated. To achieve an efficient implementation, our approach condenses maps into a graphical representation and employs equality constraints for alleged data associations. Using linear algebra techniques, these constraints can be added or removed efficiently. We evaluated our approach using some of the most challenging data sets in our possession, and consistently found that it produces accurate maps, even if the maps contain many large cycles.

In addition to the problem of learning maps of the environment we also considered the task of detecting the locations or shapes of objects. In recent years, automatic identification systems have become very popular in many service industries, including warehouses, logistics, manufacturing, and many others. Advances in the field of radio frequency identification techniques have reached a state that will allow us within the next years to equip every object in an environment with small, cheap Radio Frequency Identification (RFID) tags. An RFID reader can read the information contained in a tag within a small range around the tag. What makes the localization of tagged objects especially hard is the fact that the detection range can be several square meters. In this thesis we presented a probabilistic approach to generate maps of RFID tags with mobile robots. We proposed a sensor model that allows us to compute the likelihood of tag detections given the relative position of the tag with respect to the robot. Additionally we described how to compute a posterior about the position of a tag after the trajectory and the map has been generated with our highly accurate Rao-Blackwellized mapping algorithm mentioned above. We also presented a way to localize a robot or even persons in the environment based on the resulting model.

Furthermore, we proposed a technique for simultaneous scan registration and scan deformation for modeling nonrigid objects. We extended the Iterative Closest Point (ICP) algorithm to accommodate local object transformations. The deformation is made possible through the definition of (soft) links between neighboring scan points, whose configuration is calculated during registration. First, the static relationship between points and the corresponding vantage points is replaced by nonrigid links between these adjacent points. These links can be bent (but at a probabilistic penalty), to accommodate nonrigid surfaces. Second, and as a con-

sequence of this, the optimization involves the determination of the location of all points, in addition to the robot poses. This optimization problem is much higher dimensional than in the original ICP algorithm. To tackle the resulting optimization problem efficiently, we described a hierarchical optimization techniques that operated on thinned graphs. Experimental results obtained using a mobile robot illustrated the viability of this approach.

Finally, we presented two examples of complex robotics systems, which are able to autonomously map their environment. The first system is a 1,500 pound vehicle, nicknamed “Groundhog,” which is designed to autonomously explore and acquire 3D maps of abandoned coal mines. The core of the Groundhog navigation system is comprised of a software package that solves the SLAM problem by acquiring 2D maps. At the lowest level of processing, Groundhog’s mapping system utilizes our real-time scan matching technique for registering consecutive scans. The scan matcher enables Groundhog to recover two quantities: locally consistent maps and an estimate of the robot’s own motion. The system has been tested under extreme conditions, and generated accurate maps of abandoned mines that are inaccessible to people. This demonstrates that the autonomous acquisition of maps of abandoned mines is indeed feasible with autonomous robotic systems.

The second system is a low-flying helicopter platform designed to acquiring high-resolution ground models using active laser range sensors. Our system integrates a pose estimations from multiple sensors with range data, acquired by a 2D laser range finder oriented perpendicular to the vehicle’s flight direction. The algorithm uses a fast optimization technique to generate maps in real-time. It exploits a local smoothness assumption for the surface that is being modeled, but simultaneously allows for the possibility of large discontinuities of the mapped structure. By doing so, it can utilize range scans for vehicle localization, and thereby simultaneously improve both the pose estimate of the helicopter and the accuracy of the resulting 3D environment model. We have tested our approach in a number of different environments, all of which involved significant vertical structure that cannot easily be mapped by high-aerial vehicles.

Despite these encouraging results there is a variety of open issues for future research:

- None of the approaches addresses the issue of robot control. The ultimate goal of robotics is to make robots “do the right thing.” During map acquisition, this might be an autonomous exploration system which controls the movements of the robot and select appropriate goal points. In a broader context, this issue involves the question of what elements of the environment have to be modeled for successfully enabling a robot to perform its task therein. While these issues have been addressed for decades in ad hoc ways, little is known about the general interplay between mapping and con-

trol under uncertainty.

- Most of the work and the experimental results were carried out using laser range finders. For the development of low-cost robots it will be important to transfer our techniques to other sensor technologies like cameras.
- Whereas our algorithms have been proven robust in large scale environments, it is unclear whether they can readily be applied to map complete cities. Although hierarchical extensions or data reduction can reduce memory requirements, the complexity of the loop closing problem correlates with the length of the loop.
- Many of the state-of-the-art mapping systems are confined to flat or two-dimensional environments. The real world, however, is three-dimensional. Robotic environments can include multiple levels, like floors or bridges, or non-flat surfaces (hills, cliffs, stairways, often streets, ramps etc). Several problems arise from this fact, since the dimensionality increases and data acquisition as well as data registration and data representation become seriously more complex. Using robots in outdoor environments is quite hard: on one hand because of the limitations of the current hardware and on the other hand because of limited capabilities of sensors and the enormously complex state estimation problems.

Despite these important open questions we hope that this thesis realizes a small step towards truly autonomous mobile robots.

BIBLIOGRAPHY

- [Anguelov *et al.*, 2002] D. Anguelov, R. Biswas, D. Koller, B. Limketkai, S. Sanner, and S. Thrun. Learning hierarchical object maps of non-stationary environments with mobile robots. In *Proceedings of the 17th Annual Conference on Uncertainty in AI (UAI)*, 2002.
- [Apostolopoulos *et al.*, 2001] D. Apostolopoulos, L. Pedersen, B. Shamah, K. Shillcutt, M. D. Wagner, and W. R. Whittaker. Robotic antarctic meteorite search: Outcomes. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4174–4179, 2001.
- [Arras *et al.*, 2002] K.O. Arras, R. Philippsen, M. de Battista, M. Schilt, and R. Siegwart. A navigation framework for multiple mobile robots and its applications at the Expo.02 exhibition. In *Proc. of the IROS-2002 Workshop on Robots in Exhibitions*, 2002.
- [Arulampalam *et al.*, 2002] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002.
- [Bagnell and Schneider, 2001] J. Bagnell and J. Schneider. Autonomous helicopter control using reinforcement learning policy search methods. In *Proceedings of the International Conference on Robotics and Automation 2001*. IEEE, May 2001.
- [Baker *et al.*, 2003] C. Baker, Z. Omohundro, S. Thayer, W. Whittaker, M. Montemerlo, and S. Thrun. A case study in robotic mapping of abandoned mines.

- In *Proceedings of the International Conference on Field and Service Robotics*, Lake Yamanaka, Japan, 2003.
- [Bakircioglu *et al.*, 1999] M. Bakircioglu, S. C. Joshi, and M. I. Miller. Landmark matching on brain surfaces via large deformation diffeomorphisms on the sphere. In *SPIE Medical Imaging*, 1999.
- [Bapna *et al.*, 1998] D. Bapna, E. Rollins, J. Murphy, M. Maimone, W. R. Whitaker, and D. Wettergreen. The atacama desert trek: Outcomes. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 597–604, 1998.
- [Bar-Shalom and Fortmann, 1988] Y. Bar-Shalom and T. E. Fortmann. *Tracking and Data Association*. Academic Press, 1988.
- [Bares and Wettergreen, 1999] J. Bares and D. Wettergreen. Dante ii: Technical description, results and lessons learned. *International Journal of Robotics Research (IJRR)*, 18(7):621–649, 1999.
- [Barnett and Lewis, 1994] V. Barnett and T. Lewis. *Outliers in Statistical Data*. Wiley, New York, 1994.
- [Becker and Bove, 1995] S. Becker and M. Bove. Semiautomatic 3-D model extraction from uncalibrated 2-D camera views. In *Proc. of the SPIE Symposium on Electronic Imaging, San Jose*, 1995.
- [Belwood and Waugh, 1991] J. J. Belwood and R. J. Waugh. Bats and mines: Abandoned does not always mean empty. 9(3):13–16, 1991.
- [Bengtsson and Baerveldt, 2001] O. Bengtsson and A. Baerveldt. Localization in changing environment - estimation of a covariance matrix for the idc algorithm. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1931–1937, 2001.
- [Bennewitz *et al.*, 2002] M. Bennewitz, W. Burgard, and S. Thrun. Using EM to learn motion behaviors of persons with mobile robots. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2002.
- [Besl and McKay, 1992] P. Besl and N. McKay. A method for registration of 3D shapes. *Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.
- [Beymer and K., 2001] D. Beymer and Konolige K. Tracking people from a mobile platform. In *IJCAI-2001 Workshop on Reasoning with Uncertainty in Robotics*, 2001.

- [Borenstein and Feng, 1996] J. Borenstein and L. Feng. Measurement and correction of systematic odometry errors in mobile robots. *IEEE Journal of Robotics and Automation*, 12(6):869–880, 1996.
- [Bosse *et al.*, 2003] M. Bosse, P. Newman, M. Soika, W. Feiten, J. Leonard, and S. Teller. An atlas framework for scalable mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2003.
- [Breiman *et al.*, 1984] L. Breiman, J. H. Friedman, R. A. Ohlsen, and C. J. Stone. Classification and regression trees. 1984.
- [Brodley and Friedl, 1996] C. E. Brodley and M. A. Friedl. Identifying and eliminating mislabeled training instances. In *Proc. of the Fourteenth National Conference on Artificial Intelligence (AAAI-96)*, pages 799–805, 1996.
- [Brusey *et al.*, 2003] j. Brusey, M. Harrison, Ch. Floerkemeier, and M. Fletcher. Resoning about uncertainty in location identification with rfid. In *IJCAI-2003 Workshop on Reasoning with Uncertainty in Robotics*, 2003.
- [Bui *et al.*, 2001] H. Bui, S. Venkatesh, and G. West. Tracking and surveillance in wide-area spatial environments using the Abstract Hidden Markov Model. *Intl. J. of Pattern Rec. and AI*, 2001.
- [Burgard *et al.*, 2000] W. Burgard, A.B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. Experiences with an interactive museum tour-guide robot. *Artificial Intelligence*, 114(1-2), 2000.
- [Casper, 2002] J. Casper. Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center. Master’s thesis, Computer Science and Engineering, University of South Florida, Tampa, FL, 2002.
- [Castellanos *et al.*, 1999] J.A. Castellanos, J.M.M. Montiel, J. Neira, and J.D. Tardós. The SPmap: A probabilistic framework for simultaneous localization and map building. *IEEE Transactions on Robotics and Automation*, 15(5):948–953, 1999.
- [Chatila and Laumond, 1985] R. Chatila and J.-P. Laumond. Position referencing and consistent world modeling for mobile robots. In *Proceedings of the 1985 IEEE International Conference on Robotics and Automation*, 1985.
- [Chen and Medioni, 1991] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 1991.

- [Chui and Rangarajan, 2000] H. Chui and A. Rangarajan. A new point matching algorithm for non-rigid registration. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2000.
- [Chung *et al.*, 2003] M. K. Chung, K. J. Worsley, S. Robbins, P. Paus, J. Taylor, J. N. Giedd, J. L. Rapoport, and A. C. Evans. Deformation-based surface morphometry with an application to gray matter deformation. *NeuroImage*, 18(2):198–213, 2003.
- [Coleman *et al.*, 2002] D. Coleman, D. Creel, D. Drinka, N. Holifield, W. Mantzel, A. Saidi, and M. Zuffoletti. Implementation of an autonomous aerial reconnaissance system. GeogaTech Entry into the 2002 International Aerial Robotics Competition, 2002.
- [Cox and Hingorani, 1996] I.J. Cox and S.L. Hingorani. An efficient implementation of reids multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE Transactions on PAMI*, 18(2):138–150, February 1996.
- [Cox, 1993] I.J. Cox. A review of statistical data association techniques for motion correspondence. *International Journal of Computer Vision*, 10(1):53–66, 1993.
- [Deans and Herbert, 2000] M. Deans and M. Herbert. Experimental comparison of techniques for localization and mapping using bearing-only sensor. In *Seventh Int. Symp. on Experimental Robotics*, 2000.
- [Debevec *et al.*, 1996] P.E. Debevec, C.J. Taylor, and J. Malik. Modeling and rendering architecture from photographs. In *Proc. of the 23rd International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, 1996.
- [Dellaert *et al.*, 1999] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte Carlo localization for mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 1999.
- [Dissanayake *et al.*, 2000] G. Dissanayake, H. Durrant-Whyte, and T. Bailey. A computationally efficient solution to the simultaneous localisation and map building (SLAM) problem. In *ICRA'2000 Workshop on Mobile Robot Navigation and Mapping*, 2000.
- [Dissanayake *et al.*, 2001] G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localisation and map

- building (SLAM) problem. *IEEE Transactions of Robotics and Automation*, 2001.
- [Dittrich and Johnson, 2002] J. Dittrich and E.N. Johnson. Multi-sensor navigation system for an autonomous helicopter. In *Proceedings of the 21st Digital Avionics Systems Conference*, 2002.
- [Doh *et al.*, 2003] N. Doh, H. Choset, and W. K. Chung. Accurate relative localization using odometry. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2003.
- [Doucet *et al.*, 2000] A. Doucet, J.F.G. de Freitas, K. Murphy, and S. Russell. Rao-Blackwellised particle filtering for dynamic Bayesian networks. In *Proc. of the Conference on Uncertainty in Artificial Intelligence (UAI)*, 2000.
- [Duda *et al.*, 2001] R. Duda, P. Hart, and D. Stork. *Pattern Classification*. Wiley-Interscience, 2001.
- [Durrant-Whyte *et al.*, 2001] H. Durrant-Whyte, S. Majumder, S. Thrun, M. de Battista, and S. Scheding. A Bayesian algorithm for simultaneous localization and map building. In *Proceedings of the 10th International Symposium of Robotics Research (ISRR'01)*, Lorne, Australia, 2001.
- [Elfes, 1992] A. Elfes. Multi-source spatial data fusion using bayesian reasoning. In M. A. Abidi and R. C. Gonzalez, editors, *Data Fusion in Robotics and Machine Intelligence*, pages 137–163. Academic Press, 1992.
- [Eliazar and Parr, 2003] A. Eliazar and R. Parr. DP-SLAM: Fast, robust simultaneous localization and mapping without predetermined landmarks. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*, Acapulco, Mexico, 2003. IJCAI.
- [Feldmar *et al.*, 1997] J. Feldmar, N. Ayache, and F. Betting. 3D-2D projective registration of free-form curves and surfaces. *Computer Vision and Image Understanding: CVIU*, 65(3):403–424, 1997.
- [Ferguson *et al.*, 2003] D. Ferguson, A. Morris, D. Hähnel, C. Baker, Z. Omondi, C. Reverte, S. Thayer, W. Whittaker, W. Whittaker, W. Burgard, and S. Thrun. An autonomous robotic system for mapping abandoned mines. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Proceedings of Conference on Neural Information Processing Systems (NIPS)*. MIT Press, 2003.
- [Finkenzeller, 2000] Klaus Finkenzeller. *RFID Handboook: Radio-Frequency Identification Fundamentals and Applications*. Wiley, New York, 2000.

- [Fox *et al.*, 1999a] D. Fox, W. Burgard, F. Dellaert, and S. Thrun. Monte Carlo localization: Efficient position estimation for mobile robots. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 1999.
- [Fox *et al.*, 1999b] D. Fox, W. Burgard, and S. Thrun. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research (JAIR)*, 11:391–427, 1999.
- [Gambino *et al.*, 1997] F. Gambino, G. Ulivi, and M. Vendittelli. The transferable belief model in ultrasonic map building. In *Proceedings of the Sixth IEEE International Conference on Fuzzy Systems*, pages 601–608, 1997.
- [Greenspan and Godin, 2001] M.A. Greenspan and G. Godin. A nearest neighbor method for efficient ICP. In *Proc. of the 3rd International Conference on 3-D Digital Imaging and Modeling (3DIM01)*, 2001.
- [Gupta *et al.*, 1997] Anshul Gupta, George Karypis, and Vipin Kumar. Highly scalable parallel algorithms for sparse matrix factorization. *IEEE Transactions on Parallel and Distributed Systems*, 8(5):502–520, 1997.
- [Gutmann and Konolige, 1999] J.-S. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *Proc. of the IEEE Int. Symp. on Computational Intelligence in Robotics and Automation (CIRA)*, 1999.
- [Horn *et al.*, 1988] B. Horn, H. Hilden, and S. Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society of America A: Optics, Image Science & Vision (JOSA A)*, 5(7), 1988.
- [Horn, 1987] B. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A: Optics, Image Science & Vision (JOSA A)*, 4(4), 1987.
- [Huot *et al.*, 2000] E. G. Huot, H. M. Yahia, I. Cohen, and I. Herlin. Surface matching with large deformations and arbitrary topology: A geodesic distance evolution scheme on a 3-manifold. In *Proc. of the European Conference on Computer Vision (ECCV)*, 2000.
- [John, 1995] George H. John. Robust decision trees: Removing outliers from databases. In *First International Conference on Knowledge Discovery and Data Mining*, pages 174–179, 1995.
- [Kahn *et al.*, 1996] Roger E. Kahn, Michael J. Swain, Peter N. Prokopowicz, and R. James Firby. Gesture recognition using the perseus architecture. Technical Report TR-96-04, University of Chicago, 19, 1996.

- [Kantor and Singh, 2002] George A Kantor and Sanjiv Singh. Preliminary results in range-only localization and mapping. In *Proceedings of the IEEE Conference on Robotics and Automation (ICRA '02)*, May 2002.
- [Kluge *et al.*, 2001] B. Kluge, C. Koehler, and E. Prassler. Fast and robust tracking of multiple moving objects with a laser range finder. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2001.
- [Konecny, 2002] G. Konecny. *Geoinformation: Remote Sensing, Photogrammetry and Geographical Information Systems*. Taylor & Francis, 2002.
- [Kortenkamp *et al.*, 1996] D. Kortenkamp, E. Huber, and R. P. Bonasso. Recognizing and interpreting gestures on a mobile robot. In *Proc. of the American Conference on Artificial Intelligence*, 1996.
- [Kruse and Wahl, 1998] E. Kruse and F. Wahl. Camera-based monitoring system for mobile robot guidance. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1998.
- [Kuipers and Byun, 1991] B. Kuipers and Y.-T. Byun. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Journal of Robotics and Autonomous Systems*, 8:47–63, 1991.
- [Kuipers, 1978] B. J. Kuipers. Modeling spatial knowledge. *Cognitive Science*, 2:129–153, 1978.
- [Lavalle *et al.*, 1997] S. M. Lavalle, H. H. Gonzalez-Banos, G. Becker, and J.-C. Latombe. Motion strategies for maintaining visibility of a moving target. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 1997.
- [Lenser and Veloso, 2000] S. Lenser and M. Veloso. Sensor resetting localization for poorly modelled mobile robots. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2000.
- [Leonard and Durrant-Whyte, 1991] J. H. Leonard and H. F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. pages 376–382, 1991.
- [Leonard and Feder, 1999] J.J. Leonard and H.J.S. Feder. A computationally efficient method for large-scale concurrent mapping and localization. In *Proc. of the Ninth Int. Symp. on Robotics Research (ISRR)*, 1999.
- [Lu and Milios, 1997] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4:333–349, 1997.

- [Lu and Milios, 1998] F. Lu and E. Milios. Robot pose estimation in unknown environments by matching 2D range scans. *Journal of Intelligent and Robotic Systems*, 1998.
- [Matthies and Elfes, 1988] L. Matthies and A. Elfes. Integration of sonar and stereo range data using a grid-based representation. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, pages 727–733, 1988.
- [Matthies *et al.*, 1995] L. Matthies, E. Gat, R. Harrison, B. Wilcox, R. Volpe, and T. Litwin. Mars microrover navigation: Performance evaluation and enhancement. *Autonomous Robots*, 2(4), 1995.
- [Miller and Amidi, 1998] R. Miller and O. Amidi. 3-D site mapping with the CMU autonomous helicopter. In *Proceedings of the 5th International Conference on Intelligent Autonomous Systems*, Sapporo, Japan, 1998.
- [Miller, 2002] R. Miller. *A 3-D color Terrain Modeling System For Small Autonomous Helicopters*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 2002. Technical Report CMU-RI-TR-02-07.
- [Montemerlo and Thrun, 2002] M. Montemerlo and S. Thrun. Conditional particle filters for simultaneous mobile robot localization and people-tracking (SLAP). In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2002.
- [Montemerlo and Thrun, 2003] M. Montemerlo and S. Thrun. Simultaneous localization and mapping with unknown data association using FastSLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2003.
- [Montemerlo *et al.*, 2002] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, Edmonton, Canada, 2002. AAAI.
- [Montemerlo *et al.*, 2003] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*, Acapulco, Mexico, 2003. IJCAI.
- [Moravec and Elfes, 1985] Hans P. Moravec and Alberto Elfes. High resolution maps from wide angle sonar. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 116–121, 1985.

- [Moravec, 1988] H.P. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, pages 61–74, Summer 1988.
- [Moutarlier and Chatila, 1989] P. Moutarlier and R. Chatila. An experimental system for incremental environment modeling by an autonomous mobile robot. In *1st International Symposium on Experimental Robotics*, Montreal, June 1989.
- [Murphy *et al.*, 1999] K.P. Murphy, Y. Weiss, and M.I. Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the Conference on Uncertainty in AI (UAI)*, pages 467–475, 1999.
- [Murphy, 1999] K. Murphy. Bayesian map learning in dynamic environments. In *Neural Info. Proc. Systems (NIPS)*, 1999.
- [Murray and Little, 2001] D. Murray and J. Little. Interpreting stereo vision for a mobile robot. *Autonomous Robots*, 2001.
- [Neira and Tardós, 2001] J. Neira and J.D. Tardós. Data association in stochastic mapping using the joint compatibility test. *IEEE Transactions on Robotics and Automation*, 17(6):890–897, 2001.
- [Ng *et al.*, 2003] A. Y. Ng, H. J. Kim, M. I. Jordan, and S. Sastry. Autonomous helicopter flight via reinforcement learning. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Proceedings of Conference on Neural Information Processing Systems (NIPS)*. MIT Press, 2003.
- [Nguyen *et al.*, 2002] H. G. Nguyen, H. R. Everett, N. Manouk, and A. Verma. Autonomous mobile communication relays. In *Proceedings of the SPIE conference on AeroSense/Unmanned Ground Vehicle Technology IV*, 2002.
- [Nguyen *et al.*, 2003] H. G. Nguyen, N. Pezeshkian, M. Raymond, A. Gupta, and J.M. Spector. Autonomous communication relays for tactical robots. In *Proceedings of the 11th International Conference on Advanced Robotics (ICAR)*, 2003.
- [Nilsson, 1982] N. J. Nilsson. *Principles of Artificial Intelligence*. Springer Publisher, Berlin, New York, 1982.
- [Oriolo *et al.*, 1997] G. Oriolo, G. Ulivi, and M. Vendittelli. Fuzzy maps: A new tool for mobile robot perception and planning. 14(3):179–197, 1997.
- [Pagac *et al.*, 1998] D. Pagac, E.M. Nebot, and H. Durrant-Whyte. An evidential approach to map-building for autonomous vehicles. 14(4):623–629, 1998.

- [Pauley *et al.*, 2002] E. Pauley, T. Shumaker, and B. Cole. Preliminary report of investigation: Underground bituminous coal mine, non-injury mine inundation accident (entrapment), July 24, 2002, Quecreek, Pennsylvania, 2002. Black Wolf Coal Company, Inc. for the PA Bureau of Deep Mine Safety.
- [Philipose *et al.*, 2003] M. Philipose, K. Fishkin, D. Fox, H. Kautz, D. Patterson, and M. Perkowitz. Guide: Towards understanding daily life via auto-identification and statistical analysis. In *Proc. of the Int. Workshop on Ubiquitous Computing for Pervasive Healthcare Applications (Ubihealth)*, 2003.
- [Ramaswamy *et al.*, 2000] S. Ramaswamy, R. Rastogi, and S. Kyuseok. Efficient algorithms for mining outliers from large data sets. In *Proc. of the ACM SIGMOD International Conference on Management of Data*, 2000.
- [Röfer, 2002] T. Röfer. Using histogram correlation to create consistent laser scan maps. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2002.
- [Rusinkiewicz and Levoy, 2001] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *Proc. Third International Conference on 3D Digital Imaging and Modeling (3DIM)*, Quebec City, Canada, 2001. IEEEComputer Society.
- [Samet, 1990] Hanan Samet. *Applications of Spatial Data Structures*. Addison-Wesley Publishing Company, 1990.
- [Saripalli *et al.*, 2002] S. Saripalli, J.F. Montgomery, and G.S. Sukhatme. Visually-guided landing of an autonomous aerial vehicle. *IEEE Transactions on Robotics and Automation*, 2002.
- [Schulz *et al.*, 2001] D. Schulz, W. Burgard, D. Fox, and A.B. Cremers. Tracking multiple moving objects with a mobile robot. In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001.
- [Schulz *et al.*, 2003] D. Schulz, W. Burgard, D. Fox, and A.B. Cremers. People tracking with a mobile robot using sample-based joint probabilistic data association filters. *International Journal of Robotics Research (IJRR)*, 22(2):99–116, 2003.
- [Shatkay, 1998] H. Shatkay. *Learning Models for Robot Navigation*. PhD thesis, Computer Science Department, Brown University, Providence, RI, 1998.
- [Siegwart, R., 2002] <http://robotics.epfl.ch/>, 2002.

- [Singh *et al.*, 2002] Sanjiv Singh, George Kantor, and Dennis Strelow. Recent results in extensions to simultaneous localization and mapping. In *International Symposium on Experimental Robotics*, 2002.
- [Smith and Cheeseman, 1986] R. C. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. *International Journal of Robotics Research*, 5(4):56–68, 1986.
- [Smith *et al.*, 1990] R. C. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In I.J. Cox and G.T. Wilfong, editors, *Autonomous Robot Vehicles*, pages 167–193. Springer-Verlag, 1990.
- [Stachniss and Burgard, 2002] C. Stachniss and W. Burgard. An integrated approach to goal-directed obstacle avoidance under dynamic constraints for dynamic environments. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2002.
- [Strang, 1998] G Strang. *Introduction to Linear Algebra*. Wellesley-Cambridge Press, 1998.
- [Tadokoro *et al.*, 1995] S. Tadokoro, M. Hayashi, Y. Manabe, Y. Nakami, and T. Takamori. On motion planning of mobile robots which coexist and cooperate with human. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1995.
- [Tardós *et al.*, 2001] J.D. Tardós, J. Neira, P. Newman, and J. Leonard. Robust mapping and localization in indoor environments using sonar data. Technical Report TM 2001-04, MIT Marine Robotics Lab, 2001.
- [Teller *et al.*, 2001] S. Teller, M. Antone, Z. Bodnar, M. Bosse, S. Coorg, M. Jethwa, and N. Master. Calibrated, registered images of an extended urban area. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001.
- [Thrun *et al.*, 1998a] S. Thrun, D. Fox, and W. Burgard. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, 31:29–53, 1998. also appeared in *Autonomous Robots* 5, 253–271 (joint issue).
- [Thrun *et al.*, 1998b] S. Thrun, D. Fox, and W. Burgard. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning and Autonomous Robots (joint issue)*, 1998.
- [Thrun *et al.*, 2000] S. Thrun, W. Burgard, and D. Fox. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping. In

- Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2000.
- [Thrun *et al.*, 2002] S. Thrun, D. Koller, Z. Ghahramani, H. Durrant-Whyte, and A.Y. Ng. Simultaneous mapping and localization with sparse extended information filters. In J.-D. Boissonnat, J. Burdick, K. Goldberg, and S. Hutchinson, editors, *Proceedings of the Fifth International Workshop on Algorithmic Foundations of Robotics*, Nice, France, 2002.
- [Thrun *et al.*, 2003] S. Thrun, D. Hähnel, D. Ferguson, M. Montemerlo, R. Triebel, W. Burgard, C. Baker, Z. Omohundro, S. Thayer, and W. Whittaker. A system for volumetric robotic mapping of abandoned mines. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2003.
- [Thrun, 2001] S. Thrun. A probabilistic online mapping algorithm for teams of mobile robots. *International Journal of Robotics Research*, 20(5):335–363, 2001.
- [Thrun, 2002] S. Thrun. Robotic mapping: A survey. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millennium*. Morgan Kaufmann, 2002.
- [Tsukiyama, 2003] T. Tsukiyama. Navigation system for mobile robots using RFID tags. In *Proceedings of the International Conference on Advanced Robotics (ICAR)*, 2003.
- [Urmson *et al.*, 2001] C. Urmson, B. Shamah, J. Teza, M. D. Wagner, D. Apostolopoulos, and W. R. Whittaker. A sensor arm for robotic antarctic meteorite search. In *Proceedings of the 3rd International Conference on Field and Service Robotics (FSR)*, Helsinki, Finland, 2001.
- [van der Merwe *et al.*, 2000] R. van der Merwe, A. Doucet, N. de Freitas, and E. Wan. The unscented particle filter. Technical Report CUED/F-INFENG/TR 380, Cambridge University, Department of Engineering, 2000.
- [Wainwright, 2002] M. J. Wainwright. *Stochastic processes on graphs with cycles: geometric and variational approaches*. PhD thesis, Dept. of Electrical Engineering and Computer Science, MIT, Cambridge, MA, January 2002.
- [Waldherr *et al.*, 1998] S. Waldherr, S. Thrun, R. Romero, and D. Margaritis. Template-based recognition of pose and motion gestures on a mobile robot. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 1998.

- [Wang and Staib, 2000] Y. Wang and L. H. Staib. Shape-based 3D surface correspondence using geodesics and local geometry. In *Proc. of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2000.
- [Wang and Thorpe, 2002] C.-C. Wang and C. Thorpe. Simultaneous localization and mapping with detection and tracking of moving objects. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2002.
- [Wang *et al.*, 2003] C.-C. Wang, C. Thorpe, and S. Thrun. Online simultaneous localization and mapping with detection and tracking of moving objects: Theory and results from a ground vehicle in crowded urban areas. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2003.
- [Weiβ *et al.*, 1994] G. Weiβ, C. Wetzler, and E. von Puttkamer. Keeping track of position and orientation of moving indoor systems by correlation of range-finder scans. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 595–601, 1994.
- [Winkler, 1995] G. Winkler. *Image Analysis, Random Fields, and Dynamic Monte Carlo Methods*. Springer Verlag, Berlin, 1995.
- [Zhu, 1991] Q. Zhu. Hidden Markov model for dynamic obstacle avoidance of mobile robot navigation. *IEEE Transactions on Robotics and Automation*, 7(3), 1991.