

Simultaneous Localization and Mapping for Mobile Robots:

Introduction and Methods

Juan-Antonio Fernández-Madrigal
Universidad de Málaga, Spain

José Luis Blanco Claraco
Universidad de Málaga, Spain

Promotional Copy
Not for Redistribution

Managing Director: Lindsay Johnston
Book Production Manager: Jennifer Romanchak
Publishing Systems Analyst: Adrienne Freeland
Managing Editor: Joel Gamon
Development Editor: Hannah Abelbeck
Assistant Acquisitions Editor: Kayla Wolfe
Typesetter: Lisandro Gonzalez
Cover Design: Nick Newcomer

Published in the United States of America by
Information Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue
Hershey PA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: <http://www.igi-global.com>

Copyright © 2013 by IGI Global. All rights reserved. No part of this publication may be reproduced, stored or distributed in any form or by any means, electronic or mechanical, including photocopying, without written permission from the publisher. Product or company names used in this set are for identification purposes only. Inclusion of the names of the products or companies does not indicate a claim of ownership by IGI Global of the trademark or registered trademark.

Library of Congress Cataloging-in-Publication Data

Fernandez-Madrigal, Juan-Antonio, 1970-
Simultaneous localization and mapping for mobile robots: introduction and methods / by Juan-Antonio Fernandez-Madrigal and Jose Luis Blanco Claraco.
p. cm.
Includes bibliographical references and index.
Summary: "This book investigates the complexities of the theory of probabilistic localization and mapping of mobile robots as well as providing the most current and concrete developments"-- Provided by publisher.
ISBN 978-1-4666-2104-6 (hardcover) -- ISBN 978-1-4666-2105-3 (ebook) -- ISBN 978-1-4666-2106-0 (print & perpetual access) 1. Mobile robots. 2. Geographical positions. 3. Localization theory. I. Blanco Claraco, Jose Luis, 1981- II. Title. TJ211.415.F474 2013
629.8'932--dc23
2012015952

British Cataloguing in Publication Data

A Cataloguing in Publication record for this book is available from the British Library.

All work contributed to this book is new, previously-unpublished material. The views expressed in this book are those of the authors, but not necessarily of the publisher.

Foreword

There are theoretical experts and experimental experts, and often little overlap between the two. Dr. Juan Antonio Fernández Madrigal and Dr. Jose Luis Blanco Claraco are prominent examples of both: they have contributed novel concepts and these concepts have been rigorously tested in extensive real world experiments. I had the distinct pleasure of meeting Jose at Oxford University in the fall of 2007. We had both just joined the Mobile Robotics Group, Jose as a visiting scientist and me as a postdoc. As ever, Oxford was packed with world leaders in robotics and vision—and, in particular, the sub-fields of structure from motion and Simultaneous Localization And Mapping (SLAM). Even among such distinguished company, Jose's contributions are impressive. In his work, one finds efficient, elegant algorithms and robust real-time systems that work on live data.

The area of simultaneous localization and mapping is vast—for decades researchers have recognized SLAM as a fundamental prerequisite to capable autonomous robotics, and have built many theories and systems towards its solution. This present volume represents a monumental undertaking and in itself testifies to the breadth of the authors' experience. It takes the reader through the highlights of the field, providing sufficient historical context and theoretical foundation for the uninitiated to engage in and master this exciting topic. The authors begin with a taxonomy, dividing the problem along axes for spatial knowledge representation, the structure and dynamics of the scene, the availability of prior knowledge, and the types of sensors and actuators the robot has. They then go on to introduce a variety of robots available on the market, their sensing and actuation capabilities, and discuss the varied tasks these platforms are designed to accomplish.

Having introduced the problem and the hardware involved, the authors then dive into tools from probability and statistics (to complement this, they also offer a rich appendix, which helps make the book stand-alone and broadly accessible). In recent years, these tools have been remarkably helpful in building principled autonomous robot systems that actually work in the real world. It has been said that computer vision is estimation theory applied to images, and that SLAM is estimation theory applied to robot sensor data. Indeed, today we find probabilistic estimation theory at the heart of most perception problems. Starting with probability theory, the authors have distilled the core mathematical foundations needed to understand the topics of autonomous localization and mapping.

The problem of SLAM is often factored into two halves: first, solve the localization problem, and then solve the mapping problem. Due to the inherent uncertainty present in any real system, such a factored approach can lead to inconsistencies in a robot internal world-model. However, from a pedagogical point of view, it is favorable to approach SLAM by first discussing localization as a separate and distinct problem. This book takes that route and uses localization to introduce motion models, sensor models, and Bayesian filtering—all core concepts needed to understand the broader picture.

The third section of this book addresses mapping. There are many kinds of “maps” out there. Some scientists will argue that any state saving machine constructs a crude map. Others will argue that the internal representation must somehow “look” like the geometry we see. Generally, the kinds of maps one builds will depend entirely on the anticipated robot task and the sensors at hand. Vision-based maps look nothing like laser-based maps, and geometric maps are different from “appearance”-based maps. Having understood the chapters on mapping, the reader will know how to apply the right mapping tool and sensor suite to robotic mapping problems they may face in the field.

The book concludes with advanced topics in SLAM and directions for future research. The authors note that the problem of long-term autonomy and lifelong learning are attracting increased attention. Robots now routinely operate without human intervention for short periods of time, and a few systems have demonstrated operation over much longer periods. The state-of-the-art in mapping and localization systems has shown convincing results on large-scale environments. Three key lessons learned by the community and discussed in this book include: 1) the importance of properly modeling uncertainty; 2) using graphical, relative manifold representations; and 3) using scalable place recognition techniques. While these lessons are valuable, there are many challenges left to solve. The final chapter crystallizes and identifies the key issues and challenges we face as robotic systems are tasked to operate in increasingly large-scale environments and over long periods of time.

The techniques and algorithms presented in this book are at the heart of mobile robot perception. The authors are both expert theoreticians and experimentalists—they have much to offer and have worked hard to make this text complete and accessible. Having mastered the material in this book, the reader will be well positioned to contribute their own experience and knowledge to the growing field of mobile robotic localization and mapping, and help usher in the era of useful, long-term autonomy for mobile robots.

Gabe Sibley
George Washington University, USA

Preface

Today, robots face a similar challenge to what occurred to many members of human societies of the First World in the last century: they are trying to make their way out from a profitable and well-known position in the industry—mainly as robotic manipulators—to land into a much more unpredictable and undefined place in the service sector where they will have to work side by side with humans; from taking the role of humans at work to live *with* humans all the time; from the nuts and bolts of mechanics to the more ethereal challenges of understanding their place in our world. Mobile robots have left behind their cousins, the manipulator arms, along this way, for our world is much more dynamic, large, and complex than anything a fixed arm could handle.

From the first prototypes resembling home appliances in the 1960s to the present commercially-available humanoids that seem to have jumped out from a *manga* TV series, mobile robots have struggled to freely move among us efficiently and safely. When we look at them today, it is not difficult to imagine how they would interact with people if they had only part of the capabilities claimed by their manufacturing companies, in how many applications they might be employed, and in all the ways they could help us in our daily lives.

The general public would probably be surprised by the actual limitations of these robots. Amazing as they look (and as they truly are, from a scientific perspective), we would do better in remembering that it was only during the last two decades that robots were endowed with the first consistent and successful theory of localization and mapping, which are the two basic operations that underlie any task we could devise for any practical robot: knowing *where it is* within its environment and figuring out what *that environment looks like*. Today, these two fundamental problems cannot be considered to be completely solved for every practical situation yet, in spite of the remarkable scientific corpus developed around them. This book aims at introducing that corpus to the reader. More concretely, we focus on mobile robot localization and mapping approaches that rely on the theory of probability and statistics.

The theory involved in probabilistic localization and mapping methods can become quite cumbersome, in accordance with the importance and quality of the obtained results. Books and papers exploring those complexities are easy to find, but they may be difficult to grasp for those who are not active researchers in the area and do not have a solid background in mathematics. Furthermore, most of the material is quite scattered among journals, books, and conference papers, and in many occasions is addressed from the diverse—and often confusing—terminologies of very different disciplines. Since mobile robots have begun to get out of research labs and into the hands of the general public, we believe it is now time to offer a comprehensive introduction to these subjects that is appropriate for a wider audience than traditional scientific literature, and that gathers in a single place the fundamental concepts needed for fully understanding the problems, whatever area of science they come from.

From the perspective of two authors with many years of experience researching and teaching in this field, we have aimed this goal in the gentlest possible way, while still doing it rigorously. In particular, we have focused on three aspects: firstly, on explaining and justifying most deductions that are involved in the relevant parts of the theory, including step-by-step demonstrations that are typically obviated in specialized literature; secondly, on including the probabilistic, statistical, and robotic bases that other texts take for granted—even after saying otherwise; and thirdly, on providing a glimpse of the historical development of the covered theories and methods, not intending to offer an exhaustive historical timeline but a sufficient background. Our purpose is that the interested reader can really understand the treated issues in scope and depth, instead of just presenting powerful and sophisticated mathematical tools with obscure inner workings.

The book has been designed to be useful for practitioners, graduate and postgraduate students, and researchers mostly interested in a reference guide. No previous knowledge on probability and statistics is required—although it would speed up the reading, since two entire chapters are devoted to providing that background! Also, the prerequisites in physics, calculus, and algebra have been kept to the necessary minimum; alas, self-containment is just an ideal in any finite work these days. Thus, we have had to assume that the reader has the most elemental knowledge of those three disciplines—we provide, in Appendix E, some reinforcement on concepts that are especially important for the understanding of the problems.

This book is structured in three sections. The one that possibly makes this text more distinctive in its kind is section 1, which collects for the reader the robotic, probabilistic, and statistical backgrounds required for a good comprehension of the rest. Sections 2 and 3 follow the logical development of the main problems addressed in the book: localization and mapping, respectively. This organization is intended for both a sequential reading and for an easy selection of material for reference or teaching.

The first idea about writing this book came from the class notes by the first author for a postgraduate course on mobile robotics. Their main contents, and therefore a substantial number of concepts and explanations currently in the book, have been used for that purpose during several years; they should also be amenable for teaching in more introductory courses. In this use, a professor could choose to drop the first part if the mathematical background is assumed for the students, something that will depend on the academic context of the subject. The book also introduces some advanced issues in Simultaneous Localization And Mapping (SLAM) and many recent developments, mainly coming from the experience and continuous work of the second author during his PhD thesis and beyond.

Overall, we expect our book to serve as the starting point of a fascinating journey into this field, by setting the foundations of further detailed and thorough studies. Working in probabilistic robotics can certainly be tough, but we can assure you—this much we know—that it can be highly rewarding too. Our ultimate hope is that this text provides you with most of the tools needed to open a well-marked track into the jungle of probabilistic localization and mapping.

Juan-Antonio Fernández-Madrigal
Universidad de Málaga, Spain

José Luis Blanco Claraco
Universidad de Málaga, Spain

April 2012

Acknowledgment

The present book is the result of several years of continuous work, not only directly on the text, but also on the classes on probabilistic localization and mapping that gave rise to it, on continuing with our research in the area, and on documentation about topics not related to its main corpus, such as the history of mathematics or navigation. All of this has been benefited by the advice and aid of our supporters, colleagues, and students, but the final result would have not been possible at all without the intervention of the editors at IGI Global, the anonymous reviewers of the book, and other people who kindly offered to review early versions—we must thank especially Francisco Ángel Moreno and Eduardo Fernández for that.

The first author wishes to particularly thank several groups of students of the Master on Mechatronics Engineering of the University of Málaga, coordinated by Prof. Alfonso García-Cerezo, who provided continuous feedback to the class notes and lessons on which part of this book is based. Without the insightful and motivational questions and comments of Juan Carlos Aznar, Mariano Jaimez, Ángel Martínez, Antonio Menchero, Andrés San Millán, and many others, he would be less confident in the didactic value of relevant chapters of the book. He cannot forget either the very first students that were exposed to our stuff, especially Eduardo Fernández, Ana Gago, Javier G. Monroy, and Raúl Ruiz, who not only suffered beta versions of parts of the text, but, even now, continue actively doing research on these topics, which means that the probability that they liked the experience was strictly greater than zero.

Concerning the particular case of the methods for localization and mapping implemented by the first author on the *LEGO™ Mindstorms NXT* robots, Dr. Ana Cruz has had an invaluable role due to her pioneering efforts on the use of this robotic platform for educational purposes. Some results shown in this book have been obtained with the robots she has funded through the 2008-2010 educational research project entitled “Innovation in Engineering Control Subjects through Lego Mindstorms NXT Robots” (code PIE-008), through the *Escuela Superior de Ingeniería Informática*, and also through the System Engineering and Automation Dpt., all of them in the University of Málaga.

The second author would like to express his gratitude to the numerous researchers whom he had the luck of meeting in conferences and workshops all over the world, not only for the fun moments, but also for the inspiring talks and discussions which have always had the same effect: a continuous renewal of his motivation for continuing working hard in this exciting area. In particular, he wishes to thank Dr. Paul Newman for supervising his visit to his research lab in Oxford, an experience that enriched and widened the author’s perspectives on many technical and theoretical aspects of mobile robotics. Gabe Sibley deserves a double special mention here: first, for kindly writing the foreword of the book, and second, for his suggestions that put the author on the “right track” of looking at many estimation problems in robotics as sparse, least-squares problems.

In a more practical context, he wants to thank all the researchers, from our lab in Málaga or elsewhere, who have contributed to the *Mobile Robot Programming Toolkit (MRPT)* in one way or another, either coding or providing patches and bug reports. They all have helped improve the reliability of a tool which has proven invaluable during the preparation of many graphs and results presented in this text. Special thanks go to Antonio J. Ortiz de Galisteo for his enthusiastic work in the early versions of MRPT and to Pablo Moreno Olalla for his gigantic contributions to the mathematical and geometry modules, from which some equations of Appendix A have been taken.

Both of us have developed most of our research career within the Machine Perception and Intelligence Robotics group (MAPIR), which has proven to be a fertile context for invaluable discussions and feedback on the topics at hand and, at the same time, has provided us with diverse perspectives for each problem, ranging from computer perception to artificial cognition, which have permeated our personal visions over the years. We both wish to thank the group's permanent members, Dr. Vicente Arévalo, Dr. Ana Cruz, and Dr. Cipriano Galindo, all the PhD students of the group, and also our guest researchers, who have contributed to our work in invaluable ways, particularly Prof. Alessandro Saffiotti, Assoc. Prof. Achim Lilienthal, and Assoc. Prof. Amy Loufti, from the AASS Research Center of the Örebro University (Sweden). Special thanks must go at this point to Prof. Javier González-Jiménez, the efficient lead researcher of the group; furthermore, he has been PhD advisor for both authors; thus, without his trust and constancy, we would not have started our research careers at all.

The authors also wish to thank the public institutions that contributed funding, especially the *Junta de Andalucía* (regional government), which, through a research project, allowed the first author to extend his research on probabilistic robotics, in spite of not being directly related to localization and mapping, and also allowed the second author to work on gas mapping for mobile robots. Both lines of study have had relevant benefits to this book. In addition, several national research projects funded by the Spanish Government and European research projects funded by the EU have provided invaluable support during all these years.

Some of the images that illustrate the text are from a number of researchers and companies who have all willingly granted us permissions for their inclusion here. Therefore, we sincerely thank all the owners for their unselfish contributions (and also the father- and mother-in-law of the first author, who obtained, cleaned, and photographed two astragali for us!). Likewise, our gratitude goes to those researchers who publicly released robotic datasets or source code of their own works, since such contents have also helped enrich the book with more demonstrations of the practical utility of the discussed topics.

Last but not least, our thanks must go to our families. They always helped and motivated us throughout all the years as students and, later on, during the tough (and satisfactory) times in our academic careers. Our warmest thank you is for our wives, Ana and María, who have both unconditionally supported us during the uncountable hours of preparation of this book without the least complaint.

Chapter 8

Maps for Mobile Robots: Types and Construction

ABSTRACT

This is the first chapter of the third section. It describes the kinds of mathematical models usable by a mobile robot to represent its spatial reality, and the reasons by which some of them are more useful than others, depending on the task to be carried out. The most common metric, topological, and hybrid map representations are described from an introductory viewpoint, emphasizing their limitations and advantages for the localization and mapping problems. It then addresses the problem of how to update or build a map from the robot raw sensory data, assuming known robot positions, a situation that becomes an intrinsic feature of some SLAM filters. Since the process greatly depends on the kind of map and sensors, the most common combinations of both are shown.

CHAPTER GUIDELINE

- You will learn:
 - The main kinds of maps a mobile robot can use.
 - Their main characteristics and advantages/disadvantages in the context of recursive Bayesian localization and mapping.
 - The connection between purely geometrical approaches and more abstract (cognitive) ones.
- Provided tools:
 - How to estimate some kinds of maps assuming perfectly known robot localization.
 - A comprehensive discussion on the pros and cons of each kind of map, which can be used to choose the more appropriate for your application.
 - Detailed formulations for building occupancy grids and landmark maps. For the latter we include the particular problems of having range-bearing, bearing-only, and range-only sensors.

DOI: 10.4018/978-1-4666-2104-6.ch008

- Relation to other chapters:
 - The extensive discussion on the different kinds of maps in this chapter complements some concepts that appeared already in chapters 6 and 7.
 - Solving the mapping-only problem in this chapter serves as a base for addressing the more complex SLAM problem in chapters 9 and 10.

1. INTRODUCTION

Some approaches to autonomous robots intend not to use any explicit representation of the environment, even not to use *any* representation at all, aiming at employing the environment itself as its own best model and considering the robot as part of it (Brooks, 1991). However, having an internal model of space—a map—is currently the only known practical way of efficient and optimally planning actions (taking decisions) to operate in the long-term. Furthermore, maintaining such a map has not been ruled out as a real possibility by the modern embodied cognition paradigm (Anderson, 2003), which has extended the seminal work based on not employing any model at all. It has been shown that humans use a so-called *cognitive map* to plan routes and locate in our environments (Tolman, 1948); a different story is whether the map is explicitly stored in our brains—up to date this possibility seems controversial at least—or emerges from a set of complex interactions with the environment, if we choose a developmental perspective. In state-of-the-art mobile robot mapping and localization, the former is the dominant approach, and consequently that is the one followed in this book.

While discussing robot localization in previous chapters we already faced the most common metrical environment representations, namely grid maps and landmark or feature-based maps. This revealed the tight coupling existing between localization and mapping: the robot cannot lo-

calize well if it does not have a good map, but on the other hand, a map cannot be accurately reconstructed if the robot is poorly localized. In this chapter, we will widen and organize this perspective on maps by describing the best-known types of explicit spatial representations for robots. A few of them are quite common in the robotic localization and mapping literature, while others have very restricted niches of usability, i.e., they respond to very specific sensors, environments or tasks. The existing variety makes evident the diversity and complexity of the problems arising when an automatic mobile device is intended to operate autonomously in a given spatial region: no single map representation seems to be universally valid for all the tasks of a given robot; in fact, the choice of the map has important implications in these tasks (e.g. navigation, manipulation, etc.) that extend well beyond the issues of localization and mapping addressed in this book.

After describing the different map types, this chapter addresses how to build and update those maps from the experiences of a mobile robot, that is, from its raw sensory data. Although localization and mapping are tightly coupled, within the scope of this chapter we will assume a perfectly known robot pose in order to clarify the problem of map building. In general, metrical mapping-only can be performed through Bayesian filters much like the ones described in the localization section of this book, although the much higher dimensionality of the problem sometimes forces us to adopt approximations to make it tractable. Note that approaching only the mapping part of the localization+mapping problem could be seen as a dual perspective to that already studied in chapter 7 while explaining robot localization, where a perfect knowledge about the map was assumed in order to study how to estimate the robot pose alone. However, while we can easily think of situations where the robot is endowed with a map of its environment by its designers or operators and then left on its own to localize itself, to know its position without having a map

is a rarer situation—though a possible one. The utility of artificially decoupling these problems also comes from the fact that to update a map assuming perfectly known robot poses has become part of one family of popular SLAM methods, namely those relying on a Rao-Blackwellized Particle Filter (RBPF) approach, which will be introduced in chapter 9. Thus, the mapping-only methods provided in this chapter are not only a suitable way of introducing the mapping topic, but essential algorithms included at the core of many state-of-the-art SLAM implementations.

In summary, the chapter is structured as follows. Section 2 describes the most commonly used types of spatial representations, ranging from purely metrical maps to the most abstract representations. Then, the next two sections are devoted to probabilistic techniques for learning grid and landmark maps, respectively, always under the perspective of recursive Bayesian estimators. Finally, section 5 covers some map building algorithms which are quite common and thus worth knowing, in spite of not relying heavily (or even at all) on probabilistic foundations.

2. EXPLICIT REPRESENTATIONS OF THE SPATIAL ENVIRONMENT OF A MOBILE ROBOT

In this section, we describe the most relevant types of maps in mobile robotics. Many of them arose in the first decades of the discipline, and some have survived almost unchanged until today. Others were so specific that their use is now rather limited or obsolete.

Robotic maps can be classified, at a first glance, into two broad classes. Although there exists no consensus about how to refer to them, we will write here about *symbolic* maps and *sub-symbolic* maps. Maps within the former category comprise discrete elements (typically, real-world objects of certain complexity) that can be distinguished from the much less informative “background” of the

environment through some sort of processing of the sensory data. This does not necessarily imply non-metrical maps: maybe such a map includes topological relations between these elements or maybe only their metrical location. In general, symbolic maps minimize storage needs, but at the expense of a higher computational cost for detecting, identifying (the problem of data-association introduced in chapter 6 section 4), tracking and maintaining the distinctive elements. On the contrary, sub-symbolic maps do not deal (much) with the problem of distinguishing things: they aim at representing perceptions, either containing special objects or not. Obviously, this shifts the cost from computation to storage, following the general principle in computer science that if one wishes to reduce the computational cost of any program it is likely that he or she will have to pay an increase in its storage needs.

A problem with representing the environment through a symbolic approach (with elements that have some kind of “meaning” or “semantics,” at least for the human observer) is the automatic creation of the first symbols from sub-symbolic information, which at the end is the only one available through the robot sensors: this is the linguistic and also philosophical *symbol grounding problem* (Harnad, 1990). Few practical solutions to this problem have been reported in the robotics literature, except for *anchoring* (Coradeschi & Saffiotti, 2003), a general framework that allows the robot to link perceptions of external physical objects to symbols in its internal map—the symbol grounding problem also copes, in addition, with non-physical phenomena. A comprehensive and coherent implementation of anchoring is still an open issue, but some promising research regarding the emergence of symbols from the interaction of the robot with its environment is being conducted in the areas of developmental robotics and others.

To choose the right kind of representation for a particular problem is not straightforward, and a good knowledge of the pros and cons of each one is required. Furthermore, the frontier between

symbolic and sub-symbolic representations is far from clear, as we will see in the next paragraphs. The particular kind of map to use depends on: the robot tasks, the scale of the environment, whether the scenario is indoor or outdoor, the required precision and available computational resources, the need of fusing information from different sensors, the nature of these sensors, etc. In addition, the kind of map will determine which estimation procedures will be applicable. For instance, in the case of sub-symbolic, metrical maps, occupancy grids are usually linked to non-parametric filters, while landmark maps, where the uncertainty is typically assumed to be Gaussian, are more akin to parametric (EKF-based) estimators; in the case of symbolic, topological maps, Bayesian estimation has been applied only recently.

It is worth to notice from the beginning that the current most common and practical approaches to mapping lie on the sub-symbolic category: they represent the environment using just metrical data such as landmarks, distances to obstacles, free-space, etc., that are obtained from sensory data with little processing. That is, the most used maps are currently pure geometrical representations. Therefore, there exists a solid mathematical background to automatically construct metrical maps, as we will see. However, the problem of modeling under such approaches large-scale space—physical space that cannot be entirely perceived from a single vantage point (Kuipers, 1977)—does not have a well-established solution yet, mostly due to the hurdles associated to the problem of revisiting already known places, or *loop closure* in a sub-symbolic setting. Metrical simultaneous localization and mapping in large-scale spaces will be dealt with in chapter 10.

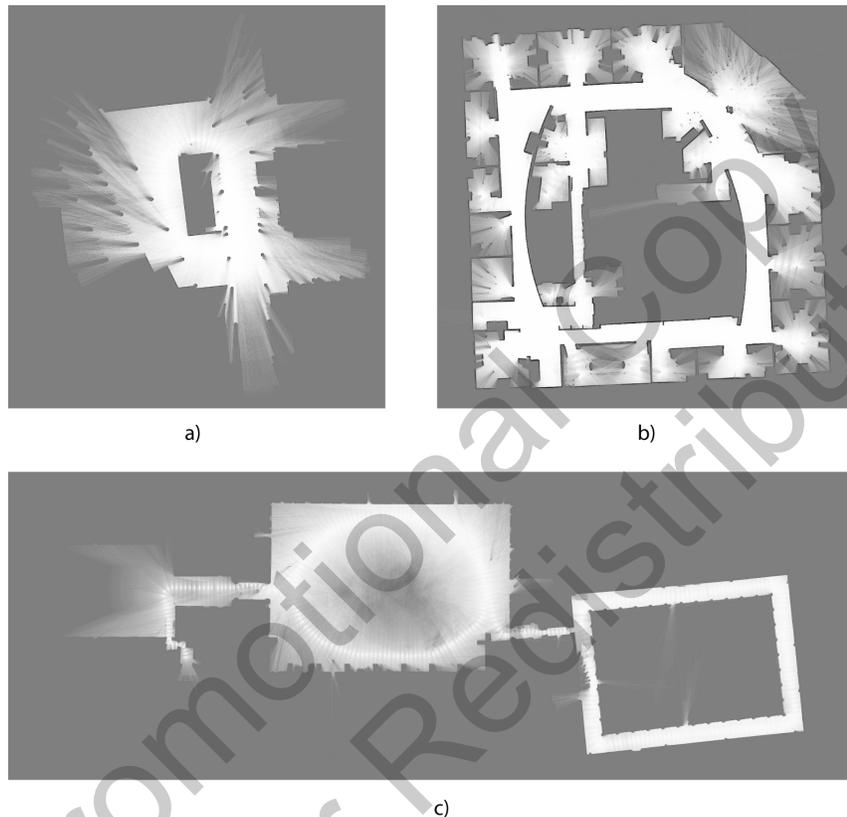
In the next paragraphs, we review the most representative kinds of explicit maps that have been in use during the history of mobile robots, ordered from the “less symbolic” to the most. We provide a few bibliographical references for each map type, such that the reader could explore deeper if interested.

Grid Maps

Possibly the best known, most popular and most basic type of map for mobile robots—in the sense of the little processing done on sensory data—is a regular tessellation of space, or grid, which cells represent the probability of the corresponding spatial region to be occupied by solid objects (see Figure 1). A grid map or occupancy grid is a *random field* (a spatial region where each point has an associated r.v.) composed of cells from which we are interested just in one property: its occupancy. Being this occupancy a discrete-valued property with only two possible real values (occupied or free), each cell can be modeled as a Bernoulli distribution. Hence, a grid map consists of a collection of such distributions, one for each cell, although that does not mean that the r.v.s are independent. The most common form of grid map is two-dimensional (where cells are square areas), although sometimes it has been employed in a three-dimensional form (with cells being small cubes, also called *voxels*). The main ideas date back to almost thirty year ago, when grid maps were proposed for mapping the environment of mobile robots by using ultrasound range sensors (Moravec & Elfes, 1985), but today they are still pervasively used. They are closely related to localization and mapping methods as remarkable as Markov Localization or Particle Filters.

Since the cells of a grid map are used for holding evidence-of-occupation values, this kind of map seamlessly fits into Bayesian recursive estimators. In particular, it naturally fits into certain family of non-parametric RBE techniques for SLAM, as will be seen in chapter 9 section 3. These maps also fit well into possibilistic representations of uncertainty, as in the case of fuzzy grid maps. In addition, most grid maps assume that each cell is conditionally independent on each other (as discussed in section 3), thus there is no a priori knowledge about the spatial structure of the environment. For example, these maps do not assume we are in an indoor or outdoor scenario.

Figure 1. Real examples of occupancy grid maps built for (a) our Málaga 2006 dataset (Blanco, Fernández-Madrigal, & González, 2008), (b) the Intel dataset (Fox, 2003), and (c) the New College dataset (Smith, Baldwin, Churchill, Paul, & Newman, 2009). All maps have been created with applications from the MRPT (2011). In section 3, we present an RBE that is able to estimate occupancy grids conditioned on the (known) poses and observations of the robot.



Since the only information of interest for each cell is its occupancy, grid maps are a valuable tool when merging information coming from different sensors, as long as all of them can provide evidence of occupancy. When a robot is equipped with sensors capable of detecting different kinds of obstacles (e.g. ultrasonic sensors detect glass doors but cannot perceive some textile fabrics, while a laser range scanner behaves exactly oppositely), care must be taken with sensor fusion since the evidence for occupation yielded by different sensors may be contradictory to each other, so different grid maps should be maintained in that case and fused together only through functions that mix

appropriately the uncertainty. Letting this issue apart, the general adequacy for sensor fusion is a valuable characteristic of grid maps that increases the robustness of many operations: localization, navigation, exploration, etc. Still another advantage of grid maps is their close correspondence to the metrical motion of the robot: both the map and the motion refer to the same frame of reference, thus, for instance, translating motion into the map to find out the robot position within the grid is straightforward.

The most evident disadvantage of grid maps is that they impose important storage requirements, especially in their three-dimensional version,

which can render them unfeasible for large-scale scenarios. For coping with that, they can be constrained to represent separate, small areas of the environment, linking several grids for modeling the whole space; this approach, however, is associated to hybrid metric-topological SLAM, which introduces its own challenges (see chapter 10). Another solution to the curse of dimensionality is adaptive tessellation, i.e., increasing the resolution of the grid around the most interesting areas, those with a higher variability in their occupancy. A recent development in this sense is the use of the abstract data type called *octree* to structure 3D grids: an octree is a mathematical tree in which each node represents the space contained in a cubic volume of space, or voxel; this volume is recursively subdivided into eight sub-volumes until a given minimum voxel size is reached, typically constrained by the sensor noise. Unknown regions or regions containing uniform information are not subdivided further. Different views of space can be obtained by selecting the appropriate amount of detail. An octree implemented for robot mapping is called an *octomap* (Wurm, Hornung, Bennewitz, Stachniss, & Burgard, 2010).

Another difficulty with grid maps is to integrate them with mobile robots that use high-level reasoning, which is mostly symbolic. Some methods exist to extract “more symbolic” information out from grids, for instance distinguishing wide open spaces and narrow passages connecting them (Fabrizi & Saffiotti, 2000), which leads to a topology of the environment very convenient for planning routes for navigation. Processing the information from a grid map to obtain higher level knowledge—i.e., a “more symbolic” map—can be done, for example, by means of image processing and computer vision techniques, considering the grid to be a gray-level image; one could identify then discrete entities such as walls by means of the Hough transform for lines (Duda & Hart, 1972), and employ those objects as landmarks within a landmark-based map.

Finally, a few techniques have been proposed in the literature for merging different grid maps (Konolige, Fox, Limketkai, Ko, & Stewart, 2003; Birk & Carpin, 2006). This problem requires identifying which parts of two grids overlap, provided that an arbitrary rotation may exist and that the overlap may typically be only partial, and it is related to the map matching methods described in chapter 6.

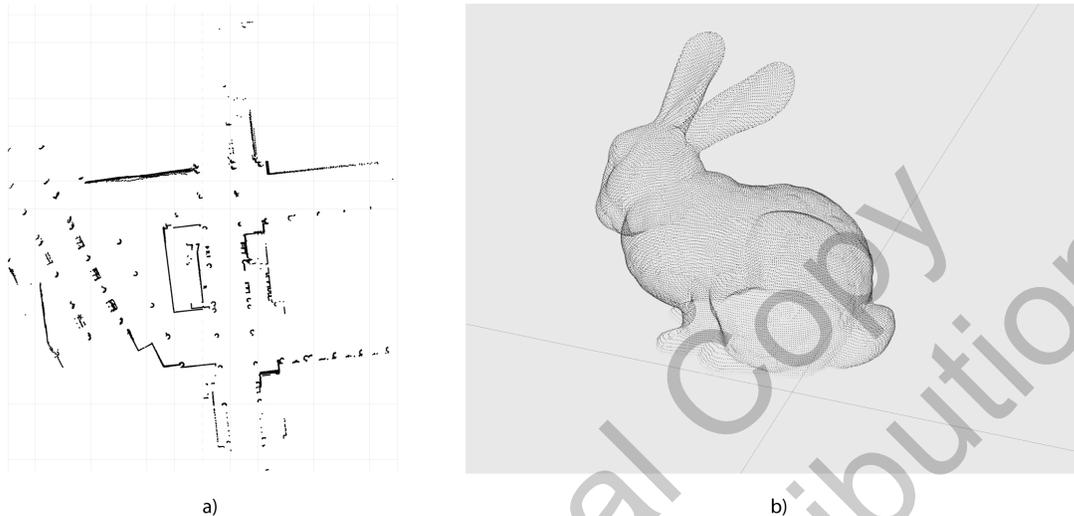
Point-Based Maps

An alternative to grid maps which alleviates their problems with storage is to use a discrete metrical representation that does not include empty space. For example, point maps represent solid parts of the environment that can be sampled by the robot through suitable sensors, usually range or range-bearing devices (recall chapter 2 sections 6 – 8). Due to that sample-based nature, it is common for these maps to be informally called *point clouds*, especially when representing three-dimensional objects or environments (see Figure 2).

Point maps were first devised in the early 90s for working with range scanners. They do not include any explicit model of uncertainty, which is one of their main drawbacks for our purposes. Also, this lack of probabilistic foundations complicates the fusion of point clouds produced by different kinds of sensors; unlike with occupancy grid maps, there is not a mathematically well-founded method for fusing several observations, even when they are captured by the same scanner from different positions in an environment, being the most common approach the brute-force accumulation of all the sensed points.

Point maps can also be processed to obtain “more symbolic” items, for example finding segments that represent walls or corners, a kind of processing that was common in mobile robotics in the nineties. The result can be considered as a sort of feature or landmark map (see further on). Nowadays, in part thanks to the recently renewed

Figure 2. (a) A point map of the same environment already shown in Figure 1a as a grid map. (b) 3D point cloud representing a “bunny figure” scanned with a Cyberware 3030MS optical triangulation scanner (data set courtesy of the Computer Graphics Laboratory/Stanford University).



interest in point clouds especially when coming from 3D range cameras, the identification of interest points in point clouds is again an active topic in the research community (Johnson & Hebert, 1999; Rusu, Marton, Blodow, Dolha, & Beetz, 2008; Tipaldi & Arras, 2010).

Free-Space Maps

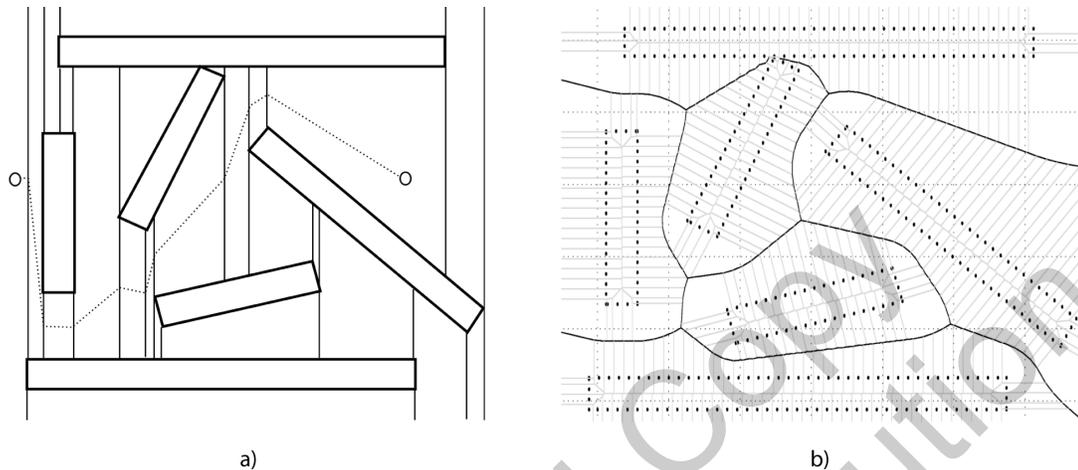
A dual alternative to point-based maps is that of representing *only free space*. Historically, this representation had an especial interest for motion planning in mobile robotics, since the robot can only move in the portion of the environment that contains no obstacles. Today, however, these maps have less interest since free space can be easily deduced from its dual information (the presence of solid objects, or obstacles), which is provided, for example, by point maps or occupancy grid maps.

Maps can represent free-space in different ways (see Figure 3): by using geometrical shapes such as trapezoids, generalized cones and others (Brooks, 1982); by partitioning space (Lozano-

Perez & Wesley, 1979); or by finding regions of particular interest for some operation (typically, for collision-free navigation). For instance, Generalized Voronoi Graphs represent a convenient tool for modeling those regions of space that are equidistant to all obstacles, that is, those places where the robot minimizes its risk to collide with the environment (Rotwat, 1979; Choset & Burdick, 1995). Since in all these approaches free space must be deduced from occupied space—the only one perceived by sensors—some processing of the latter is needed (i.e., computational cost). In addition, since they do not explicitly include uncertainty, they are not a direct choice for probabilistic frameworks. On the contrary, they were suitable in classic motion planning algorithms (Latombe, 1991).

As any map resulting from processing raw data gathered by sensors, free-space maps are closer to symbolic representations than point-based or grid maps; for example, it is easy or even straightforward to obtain a topology of space from them.

Figure 3. Two ways of representing free space for the same environment: (a) by means of geometrical shapes and (b) with generalized Voronoi graphs



Feature or Landmark Maps

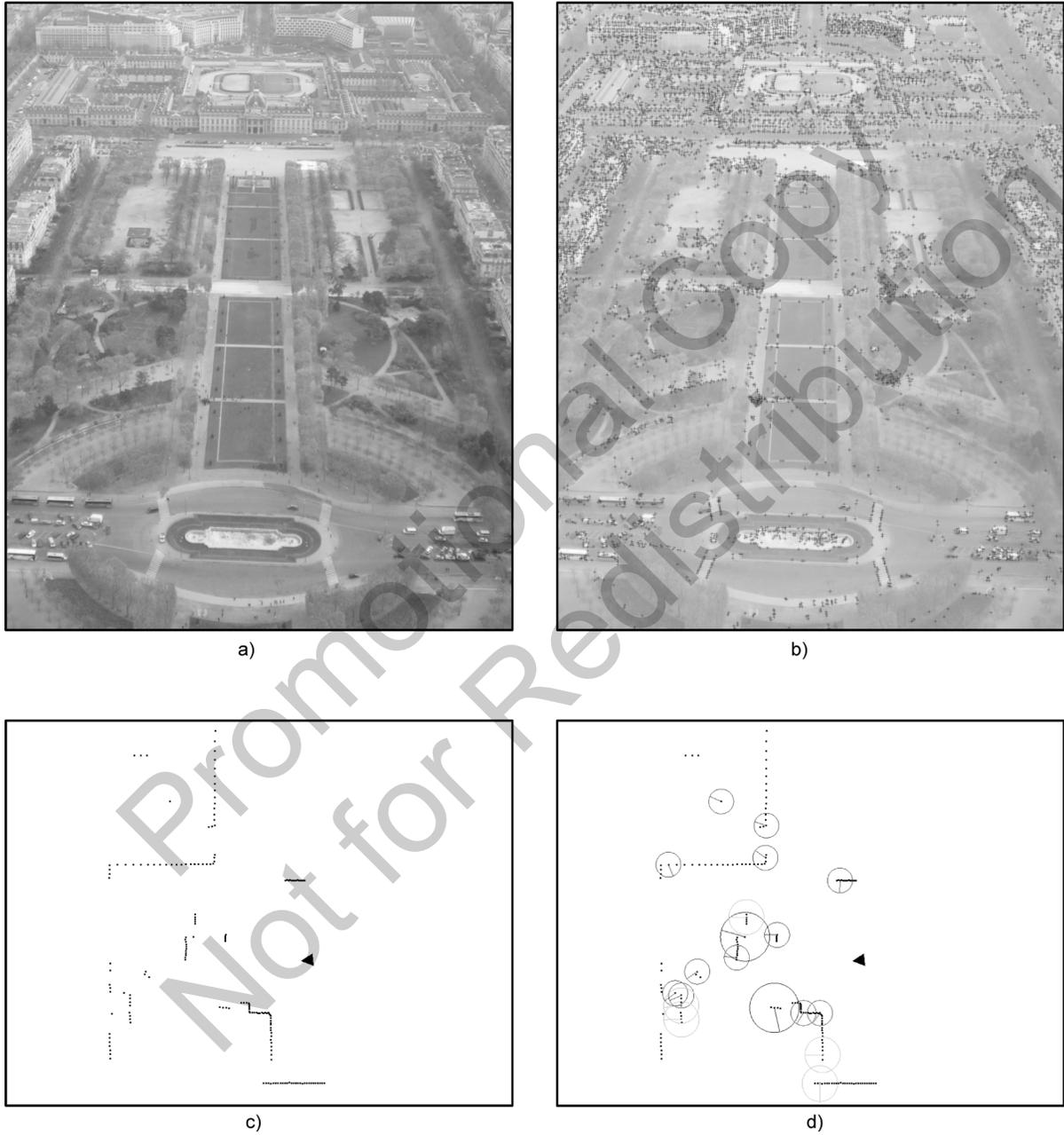
The maps described above reflect either the data coming directly from sensors (with no processing) or sensory data processed just to obtain information about regions of the environment that are of general interest. The next logical step is to process the data less generically in order to detect distinctive physical elements of the environment. These elements may have some meaning for us humans (walls, corners, doors—in indoor scenarios—trees, roads, other vehicles—in outdoor scenarios) or not (abrupt changes in perceived brightness, regions filled with some color, lines, textures, etc.). All of these lead to the category of *feature* or *landmark* maps, which are, along with grid and point maps, one of the most important kinds of maps used today in mobile robotics.

In contrast to grid maps, which do not infer any spatial structure from the data beyond the imposed, fixed tessellation of space, landmark maps do contain well-distinguished elements along with their (estimated) spatial location. Both of them are metric maps—when a topology is included in feature maps they fall into a different category—and both can include assumptions about the uncertainty of their content, therefore

both can be (and are) used extensively in probabilistic frameworks, as we have already seen in the chapters on localization. In particular, it is common in feature maps to model uncertainty with Gaussians; hence, they are commonly associated to parametric EKF-like filters (Leonard & Durrant-Whyte, 1991).

Landmark maps can be built from any sensor from which salient, well-distinguished features are extracted by means of a suitable detection algorithm. Such detectors have been proposed for 3D range cameras (Johnson & Hebert, 1999; Rusu, Marton, Blodow, Dolha, & Beetz, 2008), for the more conventional 2D range scanners (Núñez, Vázquez-Martín, del Toro, Bandera, & Sandoval, 2010) and even for sequences of sonar readings (Tardós, Neira, Newman, & Leonard, 2002). However, the most active research field for landmark maps in the last years has probably been localization and SLAM with imaging sensors. Some examples of features detected in different sensory data are shown in Figure 4. One reason for this favoring of landmarks in the computer vision community is that working with a few landmarks extracted from each video frame (ranging from a dozen up to a few hundreds, depending on the approach) means an immense reduction of the

Figure 4. Examples of features detected in readings from different types of sensors. (a) A picture of the Champ de Mars and (b) the visual features detected on it by the FAST detector algorithm. (c) An original points sensed by a laser scanner, and (d) the multiresolution features identified by FLIRT. The latter two pictures were generated from data and software courtesy of Gian Diego Tipaldi (Tipaldi & Arras, 2010).



information volume provided by the sensor, that would be hard to handle in any other way if one pretends to achieve real-time performance. Additionally, the process of detecting salient features assures that the entities or objects considered in the maps are easy to redetect from nearby locations, thus facilitating the association problem. The consequence is that methods to extract features from the images provided by vision sensors can be easily found in the extensive existing literature about computer vision (Lowe, 2004; Mikolajczyk, & Schmid, 2005; Trucco & Verri, 1998; Nixon & Aguado, 2008).

We must make a warning here. Since performing vision-based localization and SLAM in static, indoor scenarios has become increasingly functional and robust during the last decade with state-of-the-art approaches, and that research progresses with steady pace, readers without a computer vision background could fall in the wrong preconception that detecting interesting objects or features in images is an uncomplicated task, due to the astonishing effortlessly way in which we humans can interpret our surroundings from a quick look around. We must remark now that even with the latest techniques from computer vision, we are still far from automatically and robustly detecting in images what we, humans fitted with a powerful and not well understood sense of sight—which does not include only our “sensors,” the eyes—consider natural and distinctive characteristics. Today, computers have reached a performance level that makes most theoretical methods applicable, but not all mobile robots can carry on such a computational power, and anyway the cost grows with the complexity and number of features to detect, becoming impractical in some situations. Not to mention the lack of intelligence of computers performing this detection task (when a human “sees” a distinctive object, many cognitive processes are running to make the perception task converge).

An additional issue with landmark maps when working with cameras is that the spatial location

of features is not an observable variable, or at least not all its dimensions are observable. This simply means that we cannot tell the depth of a particular pixel in the image from one single video frame. Probabilistic (and some non-probabilistic) techniques can however deal with this ambiguity by means of collecting observations of features from slightly different point of views and then fusing all the information (Klein & Murray, 2007; Civera, Davison, & Montiel, 2008). All those methods, however, must introduce a *scale factor* for the map, since the real size of objects cannot be estimated from pure mathematics out of images gathered by one single camera. When we see a picture or a movie, our brain is able to interpret the scale of the objects only because we identify them within some particular context, from which our experience—cognitive processing!—infers the correct scale. In particular, a problem still open in mapping with a single camera is to avoid the drift of this world scale factor over time, which in practice leads to important inconsistencies in the reconstructed maps (Strasdat, Montiel, & Davison, 2010). The non-observability of depth is a trouble not found when employing more than one camera (e.g. stereo camera pairs) or 3D range cameras (refer to chapter 2 section 8).

Returning to the more generic discussion on landmark maps, they have a unique facet not shared by any other map type: as long as we attempt to detect individual features from within the environment, we could wrongly detect a feature (false positive) or could miss it (false negative). Even more importantly: features must be associated to previously detected features in order to decide whether they refer to the same physical element, since they should not be introduced in the map twice as different landmarks. This is what leads to the Data Association problem (DA), already discussed in chapter 6, which is computationally intractable in its exact form and must be approximated.

On the other hand, a clear advantage of the extraction of features from sensory information

is that they do not contain all the details of the underlying physical elements, that is, they are more abstract, and consequently, more robust: they vary less over time. A feature does not include all the noise conveyed by the sensor, either because we have corrected it or because we have ignored it. In contrast, a grid map, for example, is more subjected to reflect small variations in the data due to the stochastic sensor behavior, which is directly reflected in the map.

Another advantage, as we have already mentioned in the case of cameras, is that, in general, feature maps can highly reduce the storage needs with respect to grid maps, which leads to important improvements in efficiency in mobile robot operations, if we disregard the cost of feature extraction and data association.

Finally, feature maps are sub-symbolic maps but very close to be symbolic. As we have explained before, there does not exist a crisp frontier between both broad map categories. We can highlight here the fact that most types of features in the literature do not represent objects to which a human would assign any meaning, and logic relationships between them are not included in the map, but feature-based maps can be a logical and very natural basis for constructing higher-level, purely symbolic representations of the environment; therefore, they play an important role when the robot is to be enhanced with artificial reasoning capabilities.

Relational Maps and Topological Maps

A map from any of the kinds discussed above contains an intrinsic relationship between its constituents (cells, points, free-space regions, features) and the underlying space where the corresponding physical elements exist: the location of the formers onto the latter. This spatial relationship is the one that provides those maps with their metrical nature, but it is not the only one that can be included and exploited in a metrical map.

The simplest types of explicit relationship that can be added to a metrical map, apart from location, are still metrical, although they are not defined between the elements of the map and the underlying space, but among the elements themselves. For example, we can include metrical constraints that should be satisfied by the spatial locations of elements or groups of elements of the map. These so-called *constraint maps* are particularly suited for minimizing the number of hypotheses arising in a probabilistic framework and for correcting errors that are inconsistent with the imposed constraints. When the elements in the map are not world objects but robot poses, we have the popular *pose constraint map* representation (Konolige, 2005; Arras, Castellanos, Schilt, & Siegwart, 2003; Grisetti, Grzonka, Stachniss, Pfaff, & Burgard, 2007), which lies at the core of graph-SLAM approaches, described in chapter 10. To provide these maps with information, individual robot observations are stored at each pose node, leading to the so-called *view-based maps* (Konolige, Bowman, Chen, Mihelich, Calonder, Lepetit, & Fua, 2010), where an explicit representation of the environment itself does not exist. Those maps can be used to aid in splitting the environment into different areas or sub-maps, an approach that facilitates dealing with large-scale environments (Blanco, González, & Fernández-Madriral, 2009).

Whenever some method is available to split the spatial environment into regions, these regions can be connected by *topological* relations, which are spatial properties unaffected by continuous changes of shape or size of the regions; examples of use of such relations are “element A is to the left of element B” or “region A can be reached from region B through navigation.” A map that explicitly includes this type of relations is called in mobile robotics a *topological map* (Kuipers, 1978). A topological relation can also be constructed from non-metrical data, although that is rather uncommon. Topological relations and, in general, any non-metrical relation, are “elastic”:

you can project a topological map onto a plane, placing its elements at arbitrary positions of the metrical space, and the map will always be valid. They have a direct use in mobile robot operations at levels higher than motion control, such as route planning (navigation) or manipulation, and can also serve for spatial reasoning. Finally, metrical and topological maps can be combined in the so-called *hybrid metrical-topological maps* or *multi-resolution metrical maps*, which are a promising approach to represent large-scale space efficiently (Blanco, Fernández-Madrigal, & González, 2008).

Most maps including explicit relationships existing among their elements can be implemented through the abstract data type and corresponding mathematical entity called mathematical *graph* (Trudeau, 1994). A graph is a set of nodes (elements) and arcs (relations). It can be directed (if arcs represent a one-way relation) or not, can contain loops (relations affecting only one element), can have some information attached to arcs (annotated graphs, also called *networks* in the mathematical literature) and can provide support to different types of relations in the same map (*multigraphs*). A huge body of literature on graphs and their uses is available, and thus we do not include here any formalization or computational specification of graphs. Basic computational operations on graphs include path searching or routing, that is, finding a sequence of adjacent arcs with certain properties, which is at the core of the route planning operation for mobile robots. Since mathematical trees are a special kind of graphs, graph theory is useful more generally in mobile robotics, and applied on diverse problems: data association, octomaps, etc. However, care must be taken when using graphs since many operations on them are computationally intractable.

An issue with maps that include explicit relationships is their fitting into probabilistic frameworks, especially when those relations are not metrical, as is the case of topological maps. In hybrid metrical-topological maps, the metrical part does not suffer from this, and thus a com-

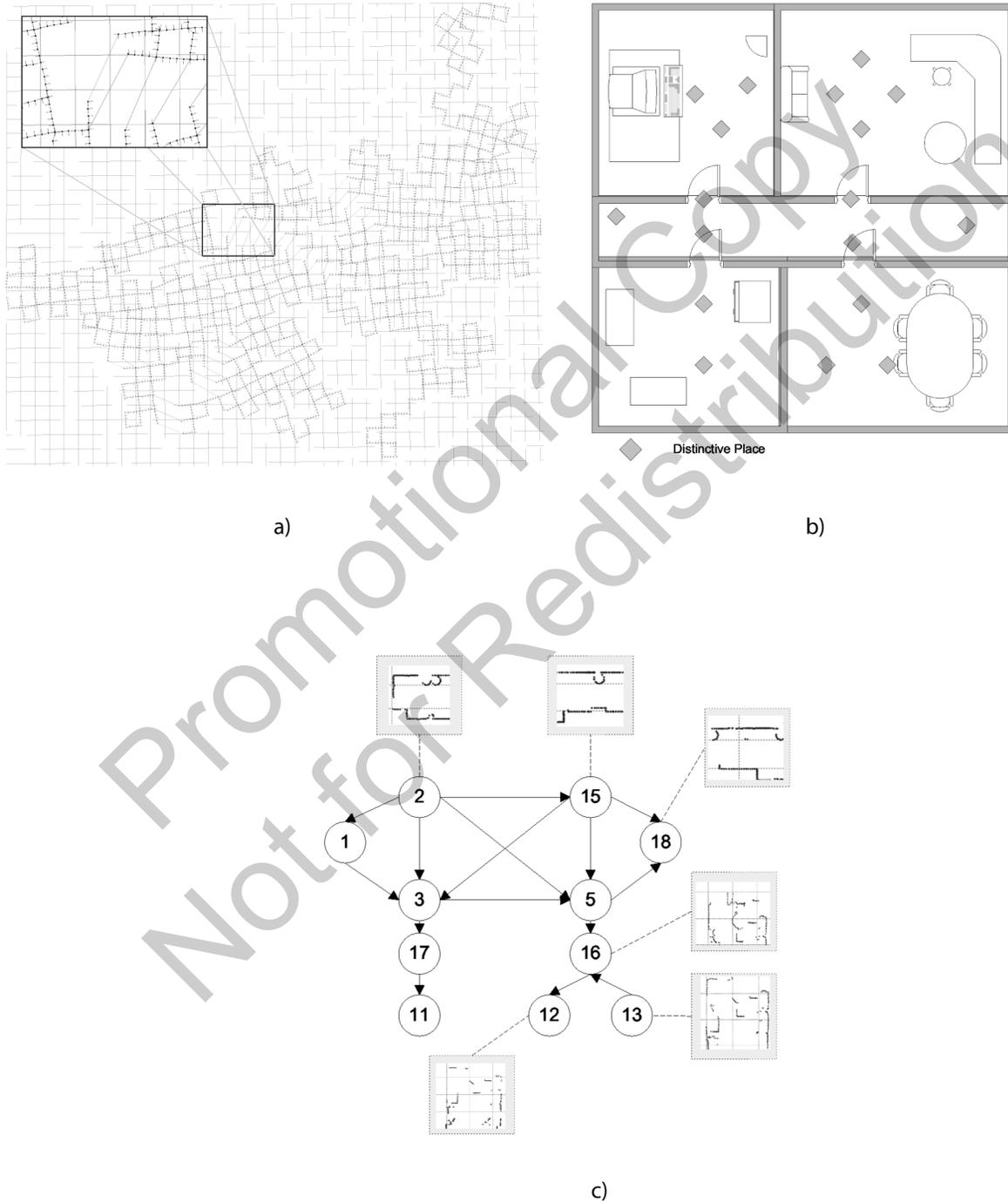
mon solution is to handle the uncertainty of the topology separately from the metrical uncertainty, using especial approaches for that (of course, you have to take into account the fact that both uncertainties are interdependent). Recently, this has been addressed by dealing probabilistically with the space of topological maps: a probability is assigned to each possible topology (Ranganathan & Dellaert, 2011; Blanco, Fernández-Madrigal, & González, 2008). The obvious problem with that is the combinatorial size of this space, which makes intractable its exhaustive exploration and forces us to deal with approximations.

Explicit spatial relationships are, in general, well suited for representing information from the environment that does not change much over time, i.e., that is robust, as we have mentioned about feature maps. Some examples of these maps are shown in Figure 5. They are the second step towards symbolic maps.

Symbolic Maps and Semantic Maps

We must insist once more in the fact that the frontier between sub-symbolic and symbolic representations of space is quite vague. To begin with, *symbols* belong to the scientific area that studies human cognition. Although they are obviously also used in other areas, such as linguistics, it is necessary to always start from the concept of what a symbol is in our brains. It seems very likely that the human mind uses symbols in some form (and probably some other animal species), but it has not been demonstrated yet which entity within the processes of a brain could possibly be associated with symbols: neural connection patterns? neural activation patterns? the dynamics of other cells different from neurons? The hypothetical cognitive processes that produce symbols from sub-symbolic information are also unknown. There is, in fact, a philosophical problem concerning symbols, the already mentioned symbol grounding problem: since symbols, by definition, must convey some meaning, how that meaning is

Figure 5. (a) A pose-constraint map, where robot poses (or “frames”) and constraints are represented as corners and edges among them, respectively. The representation has been built with applications from our software library MRPT (2011), using a two-dimensional dataset published by (Grisetti, Stachniss, Grzonka, & Burgard, 2007). (b) Example of how an environment can be assigned a set of “distinctive” places. (c) An example of a topology for a map.



associated to them from within, not from the perspective of an external observer, if the robot—or the human—only has non-symbolic information coming in from sensors? (Harnad, 1990). Actually, the problem is more general, including not only the emergence, but the maintenance of the “links” between symbols and the non-symbolic reality on which they must be grounded. A way of reducing this complexity is to consider only symbols that refer to physical elements of the world, obtaining the so-called anchoring problem, mentioned before too, but anchoring is currently far from being figured out.

In spite of this, we have to provide some concrete definition for *symbol* and for *symbolic map* in order to give form to maps of higher levels of abstraction than the ones introduced before, assuming that other definitions, even contradictory with ours, may also exist. In this chapter—and in most robotics literature—we will call *symbol* an explicit computational representation, (i.e., one with storage needs in a computer and associated algorithms that use it), of some physical, distinctive and stable element of the robot environment that can be used explicitly in high level, usually called *cognitive*, processes.

We should explain with some more detail the adjectives used in this definition to bind its vagueness a little. With “physical,” we refer to representations of parts of the environment that can be perceived by the robot sensors. All the maps previously described contain information about physical elements of the environment, but we need to explicitly include that word in our definition in order to reduce the intractable (up to date) symbol grounding problem to the intuitively more practical anchoring problem. With “distinctive,” we stress the fact that if a part of the environment is to be represented by a symbol, it must be distinguished from the background. This rules out both points and grid cells: they are clearly not considered symbolic maps. Finally, with “stable” we do not mean static or unchanging, but the quality of a physical element to exist during a certain period

of time that is enough to perceive its distinctive constituents and also to process them.

We also mentioned in our definition that a symbol “can be used explicitly in high level processes.” By “high level processes,” we refer to those that resemble human cognition: planning, reasoning, decision making, communicating with other cognitive agents, etc. Without this constraint in our definition, free-space, and feature maps could enter the symbolic class of maps: the highest level operation that can be carried out by a mobile robot with a free-space map is to control its motion to navigate (and not collide with obstacles); the one with a feature map is to locate itself in the environment. It is for certain that both kinds of maps can be used for more sophisticated operations... if they are enriched with explicit relationships or semantic knowledge, or their elements are further processed to obtain more complex ones. That is the reason why we do not consider them as symbolic.

Thus, we reach to relational maps. For a map to contain explicit relationships, it must certainly contain elements to relate (features, free-space regions, etc.), and these elements could be interpreted as symbols: they correspond to physical, distinctive and stable parts of the environment—in the sense explained before—and they *can* be used by high level processes. The emphasis in “can” is the reason why we have left relational maps as a separate class of maps and not consider them as symbolic: because they can be thought of as symbolic maps or not, depending on the situation. It is true that any symbolic map should be relational: cognitive processes require the existence of a diversity of relationships between the elements of discourse. But the reverse may not always hold: one can have relationships in a map and use them for finding routes in the environment, an operation (typical for topological maps) that is just in the edge of what can be considered a cognitive process, much more when taking into account that finding a path in a graph is a quite simple computational operation.

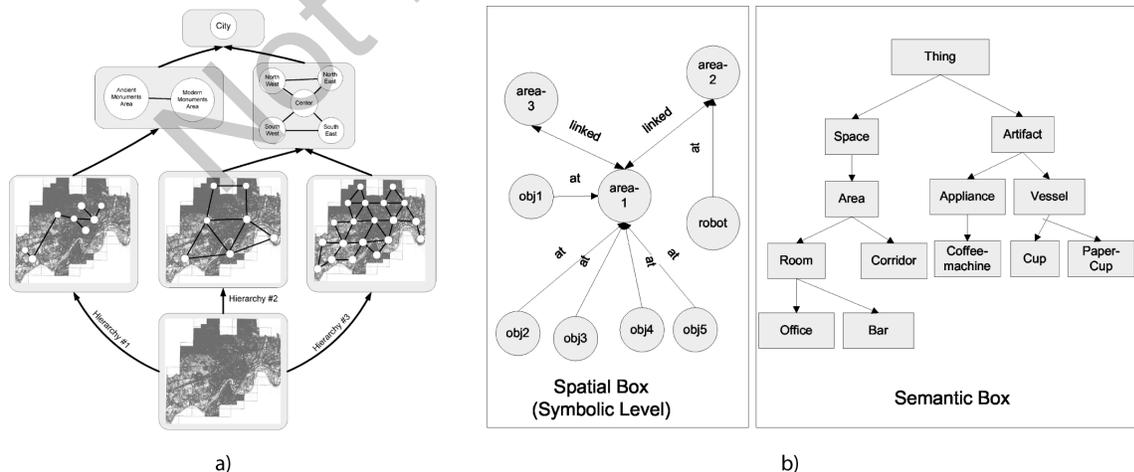
Therefore, our proposed rule is, if the map contains more than one type of explicit relation, and its elements have enough complexity in their defining characteristics, true high-level processes are possible and thus we can confidently classify the map as symbolic. We insist in that this is our way of classifying maps in this book, and others, perfectly valid, may exist.

The first kind of symbolic map that we can implement in a mobile robot comes from the inclusion of explicit *hierarchical* relations on a topological map: groups of distinctive places in the environment can be abstracted to a single symbol that represents, for instance, the room that they define from the perspective of the sensorimotor apparatus of the robot. This abstraction process can go on recursively, building up a hierarchy of abstraction that ends with a single symbol that represents the whole environment. Furthermore, the abstraction operation can be defined in different ways for the same base data, leading to multiple hierarchies of abstraction on a single map. These kinds of maps have demonstrated to improve efficiency of some mobile robot operations, to be able to optimize these operations in the long term (adapting to the particular environment where

they are used) and also to make easier the communication with humans (Fernández-Madrigal & González, 2002; Galindo, Fernández-Madrigal, & González, 2007). You can see an example in Figure 6a.

Another kind of symbolic map comes from the enrichment of the attributes of symbols such that they can be classified into “semantic categories.” If explicit relationships are added to these categories, especially the “is-a” relation, you obtain the so-called *semantic maps* (Galindo, Fernández-Madrigal, González, & Saffiotti, 2008). A semantic map is a symbolic map that allows the robot to deduce new knowledge from the general properties (semantics) of the categories of objects in the world. This can be exploited by the robot to perform better in some particular task. For example, while planning complex tasks involving moving, manipulating, communicating results, etc., semantic inference can extend the scope of the planner by providing the robot with the possibility of reasoning about elements of the environment that it has not perceived yet; the semantic structure of the map can also be used to plan at a more abstract level than with a topological representation, thus reducing computa-

Figure 6. (a) Example of a multi-hierarchy in a topological map and (b) a semantic map used by our robots (Images courtesy of Cipriano Galindo-Andrades, University of Málaga)



tional cost. An example of a semantic map is shown in Figure 5b.

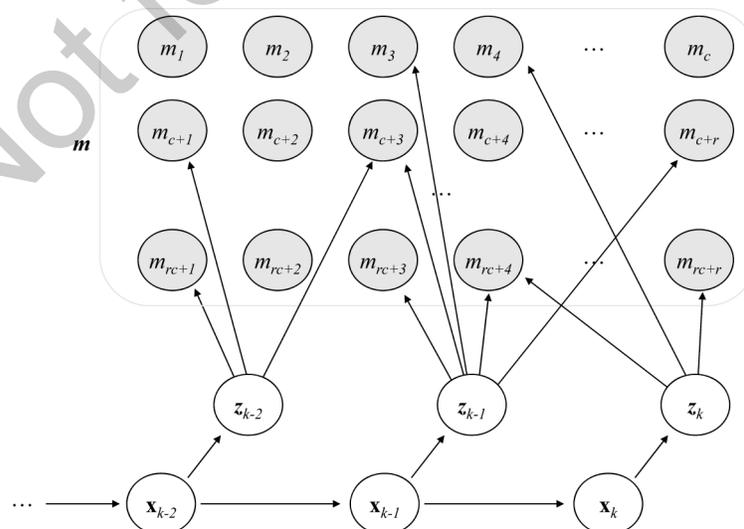
Still a different approach to symbolic maps is to create relations and symbols from non-metrical data. The work of the psychologist Jean Piaget (1948) showed that children acquire *first* relational information from the environment and *then* metrical one. This idea led to Benjamin J. Kuipers and his colleagues to propose a computational implementation of the human cognitive map with several ontological levels, the *Spatial Semantic Hierarchy* (Kuipers, 2000), in which firstly symbols represent stable operation points of basic motion behaviors (e.g., distinctive junctions or spots in free-space), then causal relations are added representing the fact of getting at that distinctive place if executing that operation from that origin place, then a topology is deduced from the causal map, and finally metrical information is added to the topological relations. A hierarchy of abstraction similar to the one described in previous paragraphs can also fit at the highest ontologies of this model (Remolina, Fernández-Madriral, Kuipers, & González-Jiménez, 1999).

In general, symbolic maps are out of the scope of the problems of this book, thus they are entirely confined in this section. Their use for localization and mapping is an interesting way to explore in the future, though. The inclusion of uncertainty in this kind of maps is in general an open issue.

3. BAYESIAN ESTIMATION OF GRID MAPS

Now that we have a general and broad vision on the kinds of explicit representations of space that a robot can have, and assuming that we know the pose of the robot, we can consider the problem of estimating a map. As we have seen along this book, a rigorous mathematical way of taking into account the uncertainty in measurements and motion is probability theory and statistics and, in particular, Bayesian recursive estimation (in the sense of *sequential* with time) is especially well suited for on-line estimation of the dynamics of continuous systems that do not exhibit abrupt changes. When estimating maps, they are usually

Figure 7. The DBN for the problem of grid map building. Shaded nodes stand for the hidden variables that will be estimated. Notice how the map variable \mathbf{m} can be considered as a vector of scalar variables m_i , one for each cell in the grid map. The indices of map variables have been numbered as for a grid map with c columns and $r - 1$ rows.



considered static, thus the reason to use a sequential approach is not for coping with changes in the map, but rather for allowing us to learn them incrementally, as new observations are gathered while the robot explores its environment.

One of most common types of map that can be estimated under a probabilistic framework is the occupancy grid map. The Dynamic Bayesian Network (DBN) for the case of estimating a grid map from robot observations, considering the map a static random variable, is shown in Figure 7. We can pose the problem mathematically as finding out the following pmf:

$$P(\mathbf{m} | \mathbf{z}_{1:t}, \mathbf{x}_{1:t}) \quad (1)$$

A problematic aspect of this representation is the extremely high dimensionality of that pmf. Notice that in a grid map, the probability distribution should include the interdependences (covariances) between any pair of map cells. If we know that a cell shows evidence of containing an obstacle, the probability of nearby cells containing an obstacle too would not be the same as if we learned that the first one was free. Estimating the full probability distributions in such a way soon becomes intractable even for small maps. Therefore, the first simplification we will make is to consider the individual elements of the map conditionally independent on each other, with which our estimation problem becomes:

$$\begin{aligned} P(\mathbf{m} | \mathbf{z}_{1:t}, \mathbf{x}_{1:t}) &= P(\{m_i\}_{i=1}^N | \mathbf{z}_{1:t}, \mathbf{x}_{1:t}) \approx \\ & \text{(assuming } m_i \perp m_j | \mathbf{z}_{1:t}, \mathbf{x}_{1:t} \text{ for any } i \neq j) \\ & \approx \prod_{i=1}^N P(m_i | \mathbf{z}_{1:t}, \mathbf{x}_{1:t}) \end{aligned} \quad (2)$$

that is, the factoring of the full joint distribution into the product of the individual pmf for each

one of the N map cells m_i . Hence, in the following we will focus on estimating those individual distributions $P(m_i | \mathbf{z}_{1:t}, \mathbf{x}_{1:t})$ instead of the joint. Notice that this simplification is not realistic at all, since each observation z_i (e.g. a laser range scan or a sonar range) is actually affected by several grid cells, thus, statistically speaking, estimating all those cells from the observation should introduce a strong correlation between them. However, the assumption of conditional independence is almost universal in the literature because it leads to efficient and convenient update equations whose results are, in practice, quite satisfactory.

Recall that each variable m_i can yield only two values: occupied or free, representing the state of that portion of space. That is, all m_i are binary random variables. Actually, we could also write down the problem in this alternative form:

$$P(\neg m_i | \mathbf{z}_{1:t}, \mathbf{x}_{1:t}) \quad (3)$$

being $\neg m_i$ the r.v. that represents the i -th cell being free (and not occupied). Of course, $P(\neg m_i | \mathbf{z}_{1:t}, \mathbf{x}_{1:t}) = 1 - P(m_i | \mathbf{z}_{1:t}, \mathbf{x}_{1:t})$.

In order to find out the RBE for this setting, we can apply the Bayes' rule to the distribution of the i -th cell at the last line of Equation 2:

$$\begin{aligned} P(m_i | \mathbf{z}_{1:t}, \mathbf{x}_{1:t}) &= P(m_i | \mathbf{z}_t, \underbrace{\mathbf{z}_{1:t-1}, \mathbf{x}_{1:t}}_{\substack{\text{this conditions} \\ \text{the entire} \\ \text{expression}}}) = \\ &= \frac{p(\mathbf{z}_t | m_i, \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t}) P(m_i | \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})}{=} \end{aligned}$$

(by the conditional independence $\mathbf{z}_t \perp \mathbf{z}_{1:t-1} | \mathbf{x}_t, m_i$ that can be seen in the DBN)

$$= \frac{p(\mathbf{z}_t | m_i, \mathbf{x}_{1:t}) P(m_i | \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})}{p(\mathbf{z}_t | \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})} =$$

(using again the Bayes' rule in the first term of the numerator, considering that $\mathbf{x}_{1:t}$ is conditioning every factor of the decomposition)

$$= \frac{\frac{P(m_i | \mathbf{z}_t, \mathbf{x}_{1:t}) p(\mathbf{z}_t | \mathbf{x}_{1:t})}{P(m_i | \mathbf{x}_{1:t})} P(m_i | \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})}{p(\mathbf{z}_t | \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})} =$$

(applying $m_i \perp \mathbf{x}_{1:t}$ in some places)

$$= \frac{\frac{P(m_i | \mathbf{z}_t, \mathbf{x}_t) p(\mathbf{z}_t | \mathbf{x}_{1:t})}{P(m_i)} P(m_i | \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})}{p(\mathbf{z}_t | \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})} =$$

$$= \frac{P(m_i | \mathbf{z}_t, \mathbf{x}_t) p(\mathbf{z}_t | \mathbf{x}_{1:t}) P(m_i | \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})}{P(m_i) p(\mathbf{z}_t | \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})} \quad (4)$$

This is the farthest we can reach by pursuing terms with known expressions. However, there still appear probabilities that are difficult to calculate, such as $p(\mathbf{z}_t | \mathbf{x}_{1:t})$ or $p(\mathbf{z}_t | \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})$. Since these terms do not depend on the r.v. we are estimating (m_i), we can get rid of them by firstly using the fact that the same deduction shown in Equation 4 applies to the dual problem of estimating the “freeness” of the cell (Equation 3):

$$P(\neg m_i | \mathbf{z}_{1:t}, \mathbf{x}_{1:t}) =$$

$$= \frac{P(\neg m_i | \mathbf{z}_t, \mathbf{x}_t) p(\mathbf{z}_t | \mathbf{x}_{1:t}) P(\neg m_i | \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})}{P(\neg m_i) p(\mathbf{z}_t | \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})} \quad (5)$$

Then, as long as $P(\neg m_i | \mathbf{z}_{1:t}, \mathbf{x}_{1:t}) > 0$, we can divide Equation 4 by Equation 5, obtaining what is called the *odds* of m_i :

$$\frac{P(m_i | \mathbf{z}_{1:t}, \mathbf{x}_{1:t})}{P(\neg m_i | \mathbf{z}_{1:t}, \mathbf{x}_{1:t})} =$$

$$= \frac{P(m_i | \mathbf{z}_t, \mathbf{x}_t) P(m_i | \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1}) P(\neg m_i)}{P(\neg m_i | \mathbf{z}_t, \mathbf{x}_t) P(\neg m_i | \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1}) P(m_i)}$$

and applying logarithms to both sides in order to simplify calculations, we get the so-called *log-odds*:

$$\ln \frac{P(m_i | \mathbf{z}_{1:t}, \mathbf{x}_{1:t})}{P(\neg m_i | \mathbf{z}_{1:t}, \mathbf{x}_{1:t})} = \ln \frac{P(m_i | \mathbf{z}_t, \mathbf{x}_t)}{P(\neg m_i | \mathbf{z}_t, \mathbf{x}_t)} +$$

$$+ \ln \frac{P(m_i | \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})}{P(\neg m_i | \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})} + \ln \frac{P(\neg m_i)}{P(m_i)} \quad (6)$$

$$= \ln \frac{P(m_i | \mathbf{z}_t, \mathbf{x}_t)}{P(\neg m_i | \mathbf{z}_t, \mathbf{x}_t)} + \ln \frac{P(m_i | \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})}{P(\neg m_i | \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})} -$$

$$- \ln \frac{P(m_i)}{P(\neg m_i)}$$

This expression involves terms that either can be calculated or are recursive functions of other terms. We name each term for convenience:

$$\underbrace{\ln \frac{P(m_i | \mathbf{z}_{1:t}, \mathbf{x}_{1:t})}{P(\neg m_i | \mathbf{z}_{1:t}, \mathbf{x}_{1:t})}}_{\ell_t(m_i)} = \underbrace{\ln \frac{P(m_i | \mathbf{z}_t, \mathbf{x}_t)}{P(\neg m_i | \mathbf{z}_t, \mathbf{x}_t)}}_{\tau_t(m_i)} +$$

$$+ \underbrace{\ln \frac{P(m_i | \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})}{P(\neg m_i | \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})}}_{\ell_{t-1}(m_i)} - \underbrace{\ln \frac{P(m_i)}{P(\neg m_i)}}_{\ell_0(m_i)}$$

$$\ell_t(m_i) = \tau_t(m_i) - \ell_0(m_i) + \ell_{t-1}(m_i) \quad (7)$$

This is the final *log-odds* recursive formulation for the on-line estimation of a binary r.v. m_i . It is not equivalent to estimating the posterior occupancy *probability* of m_i , but the latter can be deduced from it, since:

$$\begin{aligned}\ell_t(m_i) &= \ln \frac{P(m_i | \mathbf{z}_{1:t}, \mathbf{x}_{1:t})}{P(\neg m_i | \mathbf{z}_{1:t}, \mathbf{x}_{1:t})} = \\ &= \ln \frac{P(m_i | \mathbf{z}_{1:t}, \mathbf{x}_{1:t})}{1 - P(m_i | \mathbf{z}_{1:t}, \mathbf{x}_{1:t})} \leftrightarrow\end{aligned}$$

$$e^{\ell_t(m_i)} = \frac{P(m_i | \mathbf{z}_{1:t}, \mathbf{x}_{1:t})}{1 - P(m_i | \mathbf{z}_{1:t}, \mathbf{x}_{1:t})} \leftrightarrow$$

$$e^{\ell_t(m_i)} - e^{\ell_t(m_i)} P(m_i | \mathbf{z}_{1:t}, \mathbf{x}_{1:t}) = P(m_i | \mathbf{z}_{1:t}, \mathbf{x}_{1:t}) \leftrightarrow$$

$$e^{\ell_t(m_i)} = P(m_i | \mathbf{z}_{1:t}, \mathbf{x}_{1:t}) (e^{\ell_t(m_i)} + 1) \leftrightarrow$$

$$(e^{\ell_t(m_i)} \neq -1, \text{ always})$$

$$P(m_i | \mathbf{z}_{1:t}, \mathbf{x}_{1:t}) = \frac{e^{\ell_t(m_i)}}{1 + e^{\ell_t(m_i)}} = 1 - \frac{1}{1 + e^{\ell_t(m_i)}} \quad (8)$$

Therefore, we could store the log-odds values (in the range $\ell_t(m_i) \in (-\infty, +\infty)$) for the grid cells at each time step, and then retrieve their posterior occupancy estimation (in the range $P(m_i | \mathbf{z}_{1:t}, \mathbf{x}_{1:t}) \in (0, 1)$) just when needed by using Equation 8.

The log-odds representation stands as the most convenient implementation form for Bayesian grid maps in mobile robotics. In practice, updating a cell amounts to adding a positive or negative value to its current log-odds, according to the sensor readings (Equation 7). Furthermore, it is advisable for the sake of efficiency to always store log-odds as *integer* values to avoid the more costly operation of floating point addition, as long as saturation arithmetic is observed while adding the integers (i.e., taking care of avoiding overflow and underflow conditions). As demonstrated in grid map implementations such as the one of the

second author within the MRPT (2011), representing the log-odds value of each cell as one 8-bit signed integer provides an excellent computational performance and negligible rounding errors for virtually any practical mapping application.

Returning to the log-odds formulation of Equation 7, we need to provide values for $\tau_t(m_i)$ and $\ell_0(m_i)$ at each step (the latter is a constant), which requires to provide values for $P(m_i | \mathbf{z}_t, \mathbf{x}_t)$ and $P(m_i)$. The former is called the *inverse sensor model*, while the latter is the a priori information that we have about the map occupancy. If we do not know anything in particular about the occupancy of each cell at the first step, we could set $P(m_i) = P_{\text{undefined}}(m_i) = 0.5$ as a matter of convenience, since then

$$\ell_0(m_i) = \ln(P(m_i) / P(\neg m_i)) = \ln(0.5 / 0.5) = 0$$

and we would save one sum in the update of each cell.

Obtaining an expression for the inverse sensor model is more complicated, and different approaches exist. We already studied the *forward* sensor model in chapter 6, used there as the likelihood of the sensor, which in the case of the complete map would be $p(\mathbf{z}_t | \mathbf{m}, \mathbf{x}_t)$. If this is known, using Bayes' rule we can deduce the inverse model:

$$P(\mathbf{m} | \mathbf{z}_t, \mathbf{x}_t) = \frac{p(\mathbf{z}_t | \mathbf{m}, \mathbf{x}_t) P(\mathbf{m} | \mathbf{x}_t)}{p(\mathbf{z}_t | \mathbf{x}_t)} =$$

(since $\mathbf{m} \perp \mathbf{x}_t$)

$$\begin{aligned}&= \frac{p(\mathbf{z}_t | \mathbf{m}, \mathbf{x}_t) P(\mathbf{m})}{p(\mathbf{z}_t | \mathbf{x}_t)} \propto \\ &\propto p(\mathbf{z}_t | \mathbf{m}, \mathbf{x}_t) P(\mathbf{m})\end{aligned} \quad (9)$$

In the case of grid maps, we are interested in $P(m_i = a | \mathbf{z}_t, \mathbf{x}_t)$ for every cell m_i , which can be retrieved from Equation 9 by considering all the possible maps that have the value a at that cell:

$$\begin{aligned}
 P(m_i = a | \mathbf{z}_t, \mathbf{x}_t) &= \sum_{\mathbf{m}:m_i=a} P(\mathbf{m} | \mathbf{z}_t, \mathbf{x}_t) \\
 &= \sum_{\mathbf{m}:m_i=a} \frac{p(\mathbf{z}_t | \mathbf{m}, \mathbf{x}_t) P(\mathbf{m})}{p(\mathbf{z}_t | \mathbf{x}_t)} \\
 &\propto \sum_{\mathbf{m}:m_i=a} p(\mathbf{z}_t | \mathbf{m}, \mathbf{x}_t) P(\mathbf{m})
 \end{aligned}
 \tag{10}$$

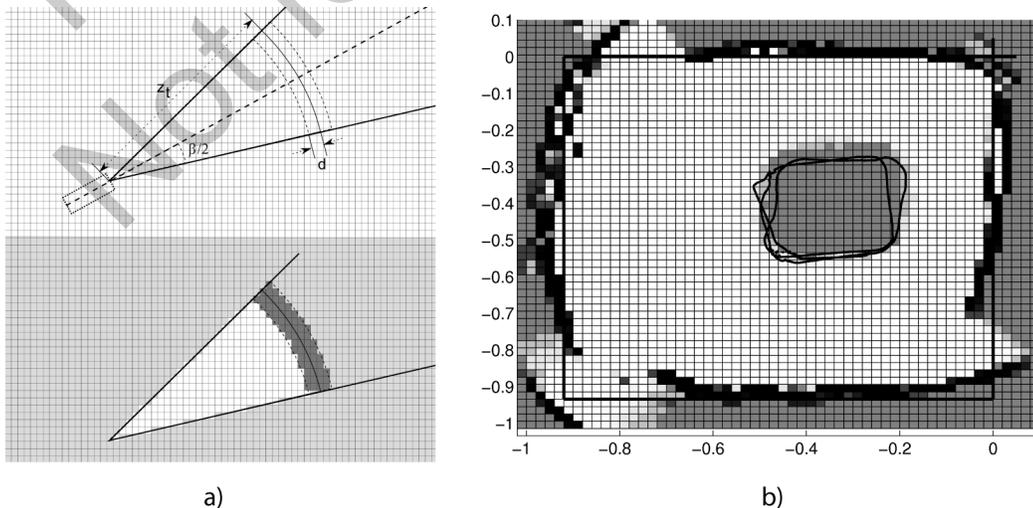
Obviously, this sum is intractable: the number of potential maps that have a certain value in a cell is overwhelming. See Figure 8. It could be approximated though through advanced techniques such as neural networks that fall out of the scope of this book.

A more practical, albeit not rigorous, approximation to the inverse sensor model is as follows (Thrun, Burgard, & Fox, 2005). Con-

sider a general beam model for the sensor consisting of a cone defined by an angle of aperture β —see Figure 7a. We have to provide a value $P(m_i | \mathbf{z}_t, \mathbf{x}_t)$ of the inverse sensor model for every cell in the grid after acquiring observation \mathbf{z}_t from location \mathbf{x}_t , in order to update the log-odds of the map. The focus of the cone, therefore, will be placed at \mathbf{x}_t , and its bisector line will point along the orientation of the line of sight of the sensor. In such a situation we can distinguish three zones in the grid—commonly, the region a cell of the grid lie in is decided by considering its center of mass or geometrical center.

Firstly, it is reasonable to assume that all the cells that fall outside the cone do not obtain any new evidence about their occupancy after that observation, thus they can keep the old log-odds previously stored in the grid. The same is valid for those that lie within the area of the cone but beyond the observed distance \mathbf{z}_t , since the observation of \mathbf{z}_t indicates the presence of something solid at that distance (with which the beam has hit) that is occluding the sight beyond. In order

Figure 8. (a) Grid mapping beam model. (b) An example of a grid map built from an ultrasonic sensor for the same navigation experiment we have conducted for the EKF and the PF localization methods in chapter 7, with the previously introduced educational robot Lego Mindstorms NXT. The robot had to move within a small square box of 92 cm x 92 cm.



to include a margin of error in this measurement, we can take $\mathbf{z}_t + d$ instead, with some small $d > 0$. Thus, all the cells outside the cone remain unchanged. We will denote this first region of the heuristic inverse model as R_u .

Secondly, those cells lying within the cone of the beam and closer than the observation \mathbf{z}_t have evidence of being free—otherwise the beam would have not reached as far as \mathbf{z}_t . For including a security margin in this area, we can take $\mathbf{z}_t - d$ instead. This region will be denoted as R_f .

Finally, all the cells lying in a region R_o defined within the distances $[\mathbf{z}_t - d, \mathbf{z}_t + d]$ from the location of the sensor and within angles $[-\beta / 2, \beta / 2]$ from the bisector of the beam have evidence of being occupied since we do not know at which point (or points) in that region did the beam bounce exactly.

In summary, we only need to update, according to Equation 7, the log-odds of the cells belonging to areas R_f and R_o . The value that we need for that is $P(m_i | \mathbf{z}_t, \mathbf{x}_t)$. We can use any value for this inverse sensor model distribution, for example

$$P(m_i | \mathbf{z}_t, \mathbf{x}_t, m_i \in R_f) = P(\neg m_i | \mathbf{z}_t, \mathbf{x}_t) = 0.25$$

and

$$\begin{aligned} P(m_i | \mathbf{z}_t, \mathbf{x}_t, m_i \in R_o) &= \\ &= P(m_i | \mathbf{z}_t, \mathbf{x}_t) = 1 - P(\neg m_i | \mathbf{z}_t, \mathbf{x}_t) = 0.75, \end{aligned}$$

as long as

$$P(\neg m_i | \mathbf{z}_t, \mathbf{x}_t) < P_{\text{undefined}}(m_i) < P(m_i | \mathbf{z}_t, \mathbf{x}_t),$$

which is satisfied with the proposed values: $0.25 < 0.5 < 0.75$. In practice, the values of

$P(\neg m_i | \mathbf{z}_t, \mathbf{x}_t)$ and $P(m_i | \mathbf{z}_t, \mathbf{x}_t)$ may be chosen heuristically or from trial and error: values too close to 0.5 will require many repeated observations of the same cells for their probability to noticeably change from their default initial value, while, on the other hand, too extreme values (close to 0 and 1, respectively) may give too much weight to spurious or noisy readings. In any case, the exact values of zero and one should never be employed as the probabilities of the sensor model update, since they lead to inconsistencies, which reflect as infinities in the log-odds formulation.

Finally, you can see in Figure 7b the result of the application of this inverse sensor model to the mapping of an environment by a simple mobile robot endowed with an ultrasound sensor; the figure displays the occupancy probabilities, recovered from the log-odds through Equation 8. Notice the near-circular area inside the map that shows an undefined occupancy: it is bounded by the circumference along which the sensor has rotated.

4. BAYESIAN ESTIMATION OF LANDMARK MAPS: GENERAL APPROACH

We now address the estimation of a map of landmarks (or features) assuming a perfect knowledge of the robot poses from which the landmark observations took place. As it will be shown, the approach may vary depending on the kind of information provided by the observations: range and bearing, bearing only or range only. The kind of available information is all we need to know in order to build the map, thus it will be irrelevant for us if the landmarks are directly detected by some sensor or if they come from some sort of post-processing over the raw sensory data. An example of range-bearing observations are the results of applying salient feature detectors to

2D laser range scans, such as the detection of tree trunks in an outdoor environment (Guivant & Nebot, 2001) or the detection of corners in indoors (Tipaldi & Arras, 2010). The most common case of bearing-only observations is, by far, image features detected in video frames from a camera: given a feature in an image, all we can say about the three-dimensional spatial location of the corresponding landmark is that it falls somewhere along a semi-infinite line emerging from the camera focus along a direction specified by the pixel coordinates (Davison, Reid, Molton, & Stasse, 2007). Finally, sensors providing range-only observations are more uncommon but include interesting families of devices such as radio or ultrasonic beacons—recall chapter 2 section 9.

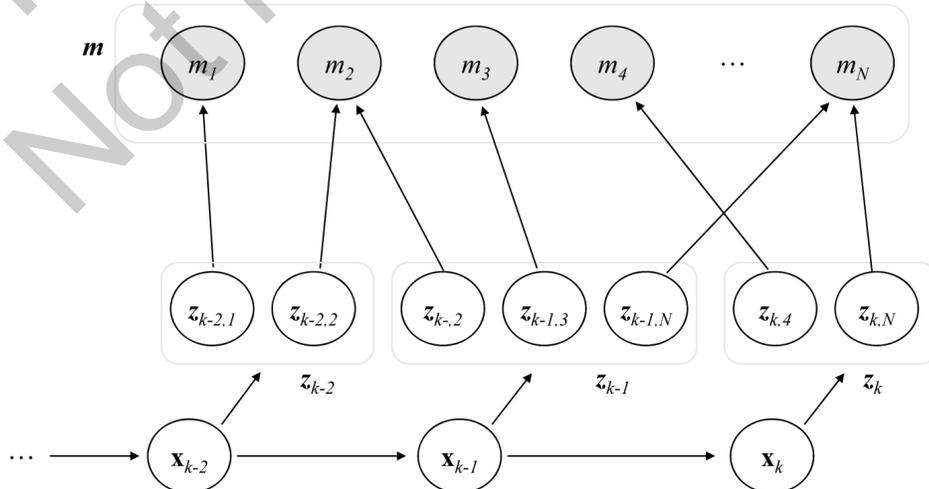
In all cases, however, our aim is the same: obtaining the spatial location of each observed landmark within an arbitrary global frame of reference along with an estimation of the uncertainty of its position. We could state the problem mathematically as the estimation of the following pdf:

$$p(\mathbf{m} | \mathbf{z}_{1:t}, \mathbf{x}_{1:t}) = p\left(\left\{\mathbf{m}_i\right\}_{i=1}^N \mid \mathbf{z}_{1:t}, \mathbf{x}_{1:t}\right) \quad (11)$$

where the map \mathbf{m} comprises in this case a set of N variables \mathbf{m}_i , each one representing the spatial location of the i -th landmark in the map. Notice that since these quantities are continuous values, the distribution of interest is a pdf instead of a pmf as it was the case with grid mapping—compare Equation 11 to Equation 1.

The DBN for this estimation problem, shown in Figure 9, is also independent of the actual kind of observations (range-bearing, bearing-only, or range-only). Observe how we have expanded in the DBN the map variable \mathbf{m} into the corresponding sequence of individual map elements: the spatial location of each landmark. Unlike in grid mapping, where a single observation was affected by several grid cells, observations in landmark mapping always consist of a sequence of *individual* landmark measurements. For instance, a range-bearing observation \mathbf{z}_k taken at some time step i may contain the range and bearing of the first and the second landmarks stored in the map; we would denote such observation as $\mathbf{z}_k = \{\mathbf{z}_{k,1}, \mathbf{z}_{k,2}\}$, with the second subscript index denoting the index in the map of the observed

Figure 9. The DBN for landmark map estimation from a sequence of known robot poses. The map and observation variables have been split into its elemental constituents. As usual, shaded nodes represent the hidden variables to be estimated.



landmark. Obviously, finding out those indices may be not straightforward since it implies solving the data association problem, already explained in chapter 6 section 4.

Observing this graphical model, we arrive at two important realizations: (1) conditioned on the whole robot path and all observations, map landmarks are conditionally independent of each other; and (2) each landmark is conditionally independent of all the robot poses and observations that do not directly observe it, given all the poses and observations that do really observe it directly. Such conclusions easily emerge by applying the concept of d-separation to the graph, as we discussed in chapter 3 section 10. Then, denoting the set of time steps from which the i -th landmark was observed as Ω_i , we can factor and simplify the target pdf in Equation 11 as follows:

$$\begin{aligned}
 & p\left(\{\mathbf{m}_i\}_{i=1}^N \mid \mathbf{z}_{1:t}, \mathbf{x}_{1:t}\right) = \\
 & \text{(factoring due to } \mathbf{m}_i \perp \mathbf{m}_j \mid \mathbf{z}_{1:t}, \mathbf{x}_{1:t} \text{)} \\
 & = \prod_{i=1}^N p\left(\mathbf{m}_i \mid \mathbf{z}_{1:t}, \mathbf{x}_{1:t}\right) = \\
 & \text{(and since)} \\
 & \mathbf{m}_i \perp \left\{ \mathbf{z}_{k,j} \right\}_{\substack{k \notin \Omega_i \\ j \neq i}}, \left\{ \mathbf{x}_k \right\}_{k \notin \Omega_i} \mid \left\{ \mathbf{z}_{k,i} \right\}_{k \in \Omega_i}, \left\{ \mathbf{x}_k \right\}_{k \in \Omega_i} \text{)} \\
 & = \prod_{i=1}^N p\left(\mathbf{m}_i \mid \left\{ \mathbf{z}_{k,i} \right\}_{k \in \Omega_i}, \left\{ \mathbf{x}_k \right\}_{k \in \Omega_i}\right) \quad (12)
 \end{aligned}$$

It is important not to get lost into the formulation details but keep clear the real significance of the expression at which we have arrived. The r.v. of each landmark position in the world can be estimated independently, just from the observations directly associated with it. There exists no cross-covariance terms linking different landmarks, but

not due to any approximation or simplification, only because we assumed a perfect knowledge of the robot pose at each time step (that cross-covariances do arise in the SLAM problem). In this mapping-only scenario, the only possible source of cross-covariance between landmarks would be in the sensor noise, which in virtually all sensors would never be a real possibility.

To realize why this independence between landmarks is so important, assume we are estimating a map with N three-dimensional landmarks. According to Equation 12, all we need is to estimate N separate pdfs of dimensionality 3. In turn, directly estimating the joint pdf $p(\mathbf{m} \mid \mathbf{z}_{1:t}, \mathbf{x}_{1:t})$ implies estimating a pdf of dimensionality $3N$. Since it is common to implement the RBE with Kalman-like filters, which typically exhibit a cubic computational complexity with the problem dimensionality (recall chapter 7 section 3), the factoring of the pdf means to pass from a complexity $O((3N)^3) \equiv O(N^3)$ down to a much simpler $O(N^3) \equiv O(N)$. More on performance will be discussed in chapter 9 section 3 when dealing with RBPF-based SLAM.

An additional advantage of this pdf factoring, not always exploited in the literature, is the possibility of modeling each landmark with a different pdf parameterization (Blanco, González, & Fernández-Madriral, 2008b), an issue touched later on.

So far our aim is to address the Bayesian estimation of the $p\left(\mathbf{m}_i \mid \left\{ \mathbf{z}_{k,i} \right\}_{k \in \Omega_i}, \left\{ \mathbf{x}_k \right\}_{k \in \Omega_i}\right)$ distributions in Equation 12, which for the sake of clarity we will refer to simply as $p\left(\mathbf{m}_i \mid \mathbf{z}_{1:t}, \mathbf{x}_{1:t}\right)$ in the following, advising the reader to keep in mind that the robot poses and the observations appearing in this expression are only those directly related to the i -th landmark. The first step is to apply Bayes' rule conditioned on the

latest observation and some conditional independences that follow from the DBN:

$$\underbrace{p(\mathbf{m}_i | \mathbf{z}_{1:t}, \mathbf{x}_{1:t})}_{\text{Posterior pdf for } t} = p(\mathbf{m}_i | \mathbf{z}_t, \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t}) =$$

(Bayes' rule applied on \mathbf{z}_t)

$$= \frac{p(\mathbf{m}_i | \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t}) p(\mathbf{z}_t | \mathbf{m}_i, \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})}{\underbrace{p(\mathbf{z}_t | \mathbf{m}_i, \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})}_{\text{This term is a constant w.r.t. the posterior}}} \propto$$

$$\propto p(\mathbf{m}_i | \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t}) p(\mathbf{z}_t | \mathbf{m}_i, \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t}) =$$

(and since $\mathbf{m}_i \perp \mathbf{x}_t | \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1}$)

$$= p(\mathbf{m}_i | \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1}) p(\mathbf{z}_t | \mathbf{m}_i, \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t}) \quad (13)$$

Looking at the second term in the last product, one can see that the landmark location \mathbf{m}_i appears as a conditioning variable, while in fact we do not know its real value (that is exactly the problem we are trying to solve!). The solution is to sum all the contributions to this density conditioned on the likelihood of each possible value of the landmark location; since \mathbf{m}_i belongs to a continuous domain, the sum becomes an integration. Doing so and using further simplifications leads to:

$$p(\mathbf{m}_i | \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1}) p(\mathbf{z}_t | \mathbf{m}_i, \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t}) =$$

(integrating over all the possible values of \mathbf{m}_i)

$$= p(\mathbf{m}_i | \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1}) \int_{-\infty}^{\infty} \underbrace{p(\mathbf{z}_t | \tilde{\mathbf{m}}_i, \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})}_{\text{can be simplified}} p(\tilde{\mathbf{m}}_i | \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t}) d\tilde{\mathbf{m}}_i =$$

(from the DBN we have $\mathbf{z}_t \perp \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1} | \tilde{\mathbf{m}}_i, \mathbf{x}_t$)

$$= p(\mathbf{m}_i | \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1}) \int_{-\infty}^{\infty} p(\mathbf{z}_t | \tilde{\mathbf{m}}_i, \mathbf{x}_t) \underbrace{p(\tilde{\mathbf{m}}_i | \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t})}_{\text{can be simplified}} d\tilde{\mathbf{m}}_i =$$

(and applying again that $\mathbf{m}_i \perp \mathbf{x}_t | \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1}$)

$$= \underbrace{p(\mathbf{m}_i | \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})}_{\text{Posterior pdf at } t-1} \int_{-\infty}^{\infty} \underbrace{p(\mathbf{z}_t | \tilde{\mathbf{m}}_i, \mathbf{x}_t) p(\tilde{\mathbf{m}}_i | \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-t})}_{\text{Posterior pdf at } t-1} d\tilde{\mathbf{m}}_i \quad (14)$$

Since we aim at estimating the individual pdf for the i -th landmark, we must assume that the correspondence between observations and the i -th map landmarks has been already done following any of the methods explained in chapter 6 section 4 for data association. As an outcome of such methods we could get two possible results: either the observed landmark (1) corresponds to any of the existing ones or (2) it is a new one not mapped yet (and the index i simply stands for any unoccupied landmark index). The former means that we need to fuse the new information with the latest filtered posterior, which must be done in a way that depends on the kind of available observations (this will be addressed below). In the latter case, if we interpret what occurs to Equations 13 and 14, we obtain:

$$\underbrace{p(\mathbf{m}_i | \mathbf{z}_{1:t}, \mathbf{x}_{1:t})}_{\text{New posterior pdf}} \propto \underbrace{p(\mathbf{m}_i | \mathbf{z}_{1:t-1}, \mathbf{x}_{1:t-1})}_{\text{Previous posterior pdf}} \int_{-\infty}^{\infty} \underbrace{p(\mathbf{z}_t | \tilde{\mathbf{m}}_i, \mathbf{x}_t)}_{\text{Previous posterior pdf}} p(\tilde{\mathbf{m}}_i) d\tilde{\mathbf{m}}_i$$

(instantiating for the first observation of a landmark, that is, $t = 1$)

$$p(\mathbf{m}_i | \mathbf{z}_1, \mathbf{x}_1) \propto \underbrace{p(\mathbf{m}_i)}_{\text{A priori pdf}} \int_{-\infty}^{\infty} p(\mathbf{z}_1 | \tilde{\mathbf{m}}_i, \mathbf{x}_1) \underbrace{p(\tilde{\mathbf{m}}_i)}_{\text{A priori pdf}} d\tilde{\mathbf{m}}_i \quad (15)$$

where it can be seen that, as could be expected, the first time a landmark is observed some sort of *a priori* distribution will be needed for its spatial location. The most generic attitude we could take here is to assume that no information is available apart from the observations themselves, thus the a priori distributions become uniform pdfs:

$$p(\mathbf{m}_i | \mathbf{z}_1, \mathbf{x}_1) \propto \underbrace{p(\mathbf{m}_i)}_{\text{Constant}} \int_{-\infty}^{\infty} p(\mathbf{z}_1 | \tilde{\mathbf{m}}_i, \mathbf{x}_1) \underbrace{p(\tilde{\mathbf{m}}_i)}_{\text{Constant}} d\tilde{\mathbf{m}}_i \propto p(\mathbf{z}_1 | \mathbf{m}_i, \mathbf{x}_1) \quad (16)$$

that is, the pdf of the landmark after its first observation coincides with the *inverse sensor model*, which is the name of the distribution $p(\mathbf{z}_t | \mathbf{m}_i, \mathbf{x}_t)$ when all the terms are known values except the map. Recall that the same distribution was named *sensor observation model* in chapter 6 when the unknown term was the observation itself.

To sum up, we have learned that updating a landmark map requires an inverse sensor model for the first time a landmark is detected, and a generic Bayesian filtering algorithm for solving Equation 14 in subsequent observations. The next sections expose some solutions for those two situations for the three different kinds of observations that we can obtain from robotic sensors.

5. BAYESIAN ESTIMATION OF LANDMARK MAPS: RANGE-BEARING SENSORS

For simplicity in the exposition, we will assume a robot moving on a planar surface, and all landmarks contained in a single plane. In this set up, we can model each map landmark as a r.v. comprising its two coordinates, that is,

$\mathbf{m}_i = (m_{x_i} \ m_{y_i})^T$. As already discussed in

chapter 6 section 3, a range-bearing sensor provides us with observations $\mathbf{z}_k = (r_k \ b_k \ \varphi_k)^T$,

having one range r_k and one bearing angle b_k that describe the landmark position as detected from the instantaneous pose of the sensor

$\mathbf{s}_k = (s_{x_k} \ s_{y_k} \ s_{\theta_k})^T$. The value φ_k represents the identification of the sensed landmark, and will be present only if the sensor is able to uniquely identify it in the environment. Since it was assumed above that data association was already solved at this point, we will go on with

$\mathbf{z}_k = (r_k \ b_k)^T$. Regarding the sensor pose, it is straightforwardly computed given the robot pose

$\mathbf{x}_k = (x_k \ y_k \ \theta_k)^T$ but for clarity we will assume that both coincide, i.e., the sensor is exactly at the origin of the robocentric coordinate reference.

The Inverse Sensor Model

Once we stated the parameterization of the problem variables, we aim at providing the inverse sensor model, for which we have to start from the sensor observation model (or direct model):

$$\mathbf{z}_k = \begin{bmatrix} r_i \\ b_i \end{bmatrix} = \mathbf{h}_i(\mathbf{x}_k, \mathbf{m}_i) + \mathbf{n}_k \quad (17)$$

Here, $\mathbf{n}_k \sim N(\mathbf{0}, \mathbf{R})$ is an additive zero-mean Gaussian noise and the observation function is the one already presented in chapter 6's Equation 6, repeated here for convenience:

$$\mathbf{h}_i(\mathbf{x}_t, \mathbf{m}_i) = \begin{pmatrix} \sqrt{(m_{x_i} - x_t)^2 + (m_{y_i} - y_t)^2} \\ \text{atan2}(m_{y_i} - y_t, m_{x_i} - x_t) - \phi_t \end{pmatrix} \quad (18)$$

Obtaining an inverse sensor model implies having a probabilistic relationship such that, given a sensor reading from a known robot pose, yields a pdf for the location of a landmark \mathbf{m}_i that explains the known data. By finding the value of \mathbf{m}_i in Equation 17 we have:

$$\begin{aligned} \mathbf{z}_k &= \mathbf{h}_i(\mathbf{x}_k, \mathbf{m}_i) + \mathbf{n}_k && \rightarrow \\ \mathbf{z}_k - \mathbf{n}_k &= \mathbf{h}_i(\mathbf{x}_k, \mathbf{m}_i) && \rightarrow \\ \mathbf{m}_i &= \mathbf{h}_i^{-1}(\mathbf{x}_k, \mathbf{z}_k - \mathbf{n}_k) \end{aligned} \quad (19)$$

where the inverse observation function¹ takes one robot pose \mathbf{x}_k and an observation \mathbf{z}_k and returns the projected location of the landmark. In this case, it can be shown that this function is:

$$\begin{pmatrix} m_{x_i} \\ m_{y_i} \end{pmatrix} = \mathbf{m}_i = \mathbf{h}_i^{-1}(\mathbf{x}_k, \mathbf{z}_k) = \begin{pmatrix} x_k + r_i \cos(\phi_k + b_i) \\ y_k + r_i \sin(\phi_k + b_i) \end{pmatrix} \quad (20)$$

Up to this point, all the derivation was based on statistical bases and was totally generic, but now we need to decide what specific distribution will be used to model the uncertainty in the map landmarks. For range-bearing sensors, it turns out that approximating that uncertainty as Gaussian is quite reasonable. Thus, the inverse sensor model $p(\mathbf{z}_k | \mathbf{m}_i, \mathbf{x}_k)$ in this case equals the distribution $N(\mathbf{m}_i; \bar{\mathbf{m}}_i, \Sigma_{\mathbf{m}_i})$, the parameters of which are derived next.

Notice from Equation 19 that the noise \mathbf{n}_k is the unique input to the inverse function $\mathbf{h}_i^{-1}(\cdot)$ that is not perfectly known, but a pdf: it is the sensor noise uncertainty, the only one that leads to uncertainty in the landmark position. As usual when faced with uncertainty transformations, an appealing solution is to apply linearization, arriving at an expected value of:

$$\bar{\mathbf{m}}_i = \mathbf{h}_i^{-1}(\mathbb{E}[\mathbf{x}_k], \mathbb{E}[\mathbf{z}_k - \mathbf{n}_k]) \quad \rightarrow$$

(since \mathbf{x}_k and \mathbf{n}_k are known values and the mean of \mathbf{n}_k is zero)

$$\bar{\mathbf{m}}_i = \mathbf{h}_i^{-1}(\mathbf{x}_k, \mathbf{z}_k) \quad (21)$$

and a covariance of:

$$\Sigma_{\mathbf{m}_i} = \frac{\partial \mathbf{h}_i^{-1}(\cdot)}{\partial \mathbf{z}_k} \mathbf{R} \frac{\partial \mathbf{h}_i^{-1}(\cdot)}{\partial \mathbf{z}_k}^T$$

With the Jacobian, evaluated at the mean of \mathbf{n}_k , given by:

$$\frac{\partial \mathbf{h}_i^{-1}(\cdot)}{\partial \mathbf{z}_k} = \begin{pmatrix} \cos(\phi_k + b_i) & -r_i \sin(\phi_k + b_i) \\ \sin(\phi_k + b_i) & r_i \cos(\phi_k + b_i) \end{pmatrix} \quad (22)$$

Notice that all of this is possible only because landmarks are completely *observable* with range-bearing sensors, as was already mentioned in chapter 2 section 3. That means that there exists a well-defined inverse sensor function $\mathbf{h}_i^{-1}(\cdot)$, something that not always occur with other sensors.

Recursive Bayesian Estimation

We now address the update of a map with successive observations of a landmark, which was already mapped upon its first detection. This implies providing a concrete implementation for Equation 15. Recall that we already decided to represent uncertainty in landmarks as Gaussian distributions and that the perfect knowledge of the robot path allows us to update the pdf of each landmark independently. The Gaussianity assumption and the non-linearity of the observation function $\mathbf{h}_i(\cdot)$ defined in Equation 18 makes the Extended Kalman Filter (EKF) a good choice as the algorithm to estimate the pdf for each individual landmark. In this particular problem the EKF equations are simpler since there is no prediction stage (the transition or motion model) due to the assumption of a static map. Recall that each landmark pdf is parameterized as $\mathbf{m}_i \sim N(\bar{\mathbf{m}}_i, \Sigma_{\mathbf{m}_i})$, thus the EKF equations must provide the updated mean and covariance matrix from the previous ones and the new robot pose and observation. From chapter 7's Equation 44, the equations turn out to be:

$$\bar{\mathbf{m}}_i \leftarrow \bar{\mathbf{m}}_i + \mathbf{K}_{k,i} (\mathbf{z}_k - \mathbf{h}_i)$$

$$\Sigma_{\mathbf{m}_i} \leftarrow (\mathbf{I} - \mathbf{K}_{k,i} \mathbf{H}_{k,i}) \Sigma_{\mathbf{m}_i}$$

with the Kalman gain matrix being:

$$\mathbf{K}_{k,i} = \Sigma_{\mathbf{m}_i} \mathbf{H}_{k,i}^T (\mathbf{R} + \mathbf{H}_{k,i} \Sigma_{\mathbf{m}_i} \mathbf{H}_{k,i}^T)^{-1} \quad (23)$$

$\mathbf{H}_{k,i}$ is the Jacobian of $\mathbf{h}_i(\cdot)$ with respect to the landmark coordinates, which easily follows from Equation 18 to be:

$$\mathbf{H}_{k,i} = \frac{\partial \mathbf{h}_i(\mathbf{x}_k; \mathbf{m}_i)}{\partial \mathbf{m}_i} = \begin{pmatrix} \frac{m_{x_i} - x_k}{r_i} & \frac{m_{y_i} - y_k}{r_i} \\ -\frac{m_{y_i} - y_k}{r_i^2} & \frac{m_{x_i} - x_k}{r_i^2} \end{pmatrix} \quad (24)$$

In the case that the uncertainties for a particular experimental set up are so large that the linearization in EKF represents a poor approximation to the actual pdfs, the UKF algorithm could be used instead. The reader can review all the properties and equations of both filters in chapter 7 section 3.

6. BAYESIAN ESTIMATION OF LANDMARK MAPS: BEARING-ONLY SENSORS

This kind of landmark observations is of the greatest interest since they are the ones obtained from image features while performing vision-based mapping or SLAM. It is possible to address two-dimensional mapping with these observations, for example, by detecting vertical features in images of the environment (like doorframes or corridor corners). Although ideas like this were proposed during the last two decades, the truth is that they never became really popular, probably because of the much richer information that one can obtain by considering the full 3D information embedded in the images instead.

Therefore, we will focus here on a full 3D approach, which complicates the formulation and introduces new challenges with respect to previously seen methods. When working on a planar environment, a robot pose simply consists of a pair of coordinates that define the translation and a third parameter for the rotation angle; it will be always like that and there is no room for further complications. In contrast, the first hurdle when dealing with three-dimensional localization, mapping or SLAM is to choose a parameterization for

the spatial pose of the camera—quite often visual mapping is performed with hand-held cameras, thus we will refer to the camera pose instead of the robot pose all along this section. Employing unit *quaternions* to represent 3D rotations is the preferred approach when those rotations are to be directly estimated within a Bayesian filter, as is our case—an alternative using more mathematically advanced techniques is briefly discussed in chapter 10. This approach, common in the literature (Civera, Davison, & Montiel, 2008; Davison, Reid, Molton, & Stasse, 2007), consists in describing the camera pose by means of its three spatial coordinates (the translational part) plus other four coordinates interpreted as a unit quaternion (rotational part), that is:

$$\mathbf{x}_k = \left(x_k \quad y_k \quad z_k \quad q_{k,r} \quad q_{k,x} \quad q_{k,y} \quad q_{k,z} \right)^T \quad (25)$$

In order to grasp the geometrical meaning of the four quaternion coordinates, one can visualize them as a rotation of a magnitude proportional to $q_{k,r}$ around the spatial direction defined by the vector $\left(q_{k,x} \quad q_{k,y} \quad q_{k,z} \right)^T$ —refer to Appendix A.

The next important complication that arises with visual-based mapping or SLAM is the need to perform data association from visual and geometrical information. Typically, feature points

are selected with salient keypoint detectors, and then are assigned some sort of feature descriptor, which can be as simple as a patch of the surrounding image or quite involved high-dimensional descriptors (Lowe, 2004). The challenge is to achieve a combination of detectors, descriptors and matching algorithms for pairing features between different frames such that it is fast enough to be executed in real-time (at the camera frame rate) and while keeping low the ratio of false positives and false negatives. In any case, all those operations are far beyond the scope of this text, thus we will assume next that data association is already solved, just like in the previous case for range-bearing observations.

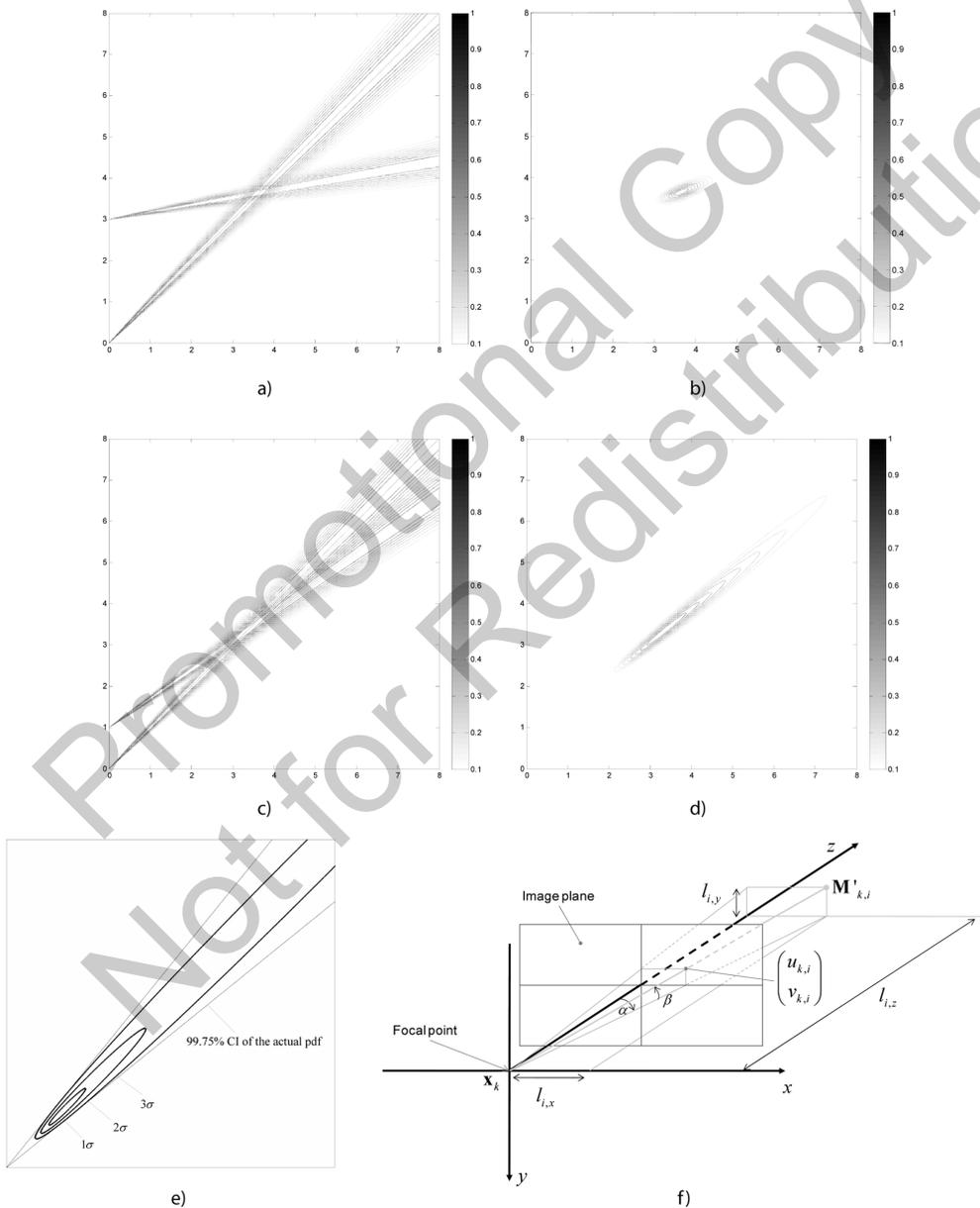
Thus, we will define a bearing-only observation as $\mathbf{z}_{k,i} = \left(\alpha_{k,i} \quad \beta_{k,i} \right)^T$, comprising an azimuth (*yaw*) and elevation (*pitch*) angles, respectively. The relationship of these angles with the location of the landmark can be easily established from geometry to be:

$$\begin{cases} \alpha_{k,i} = \text{atan2}(l_{i,z}, l_{i,x}) \quad , \text{ or } \tan^{-1} \left(\frac{l_{i,z}}{l_{i,x}} \right) \\ \beta_{k,i} = \sin^{-1} \left(\frac{l_{i,y}}{\sqrt{l_{i,x}^2 + l_{i,y}^2 + l_{i,z}^2}} \right) \end{cases} \quad (26)$$

Box 1.

$$\begin{aligned} \mathbf{M}'_{k,i}(\mathbf{M}_i, \mathbf{x}_k) &= \mathbf{M}_i \ominus \mathbf{x}_k = \begin{pmatrix} l_{i,x} \\ l_{i,y} \\ l_{i,z} \end{pmatrix} = \\ &= \begin{pmatrix} g_{i,x} - x_k \\ g_{i,y} - y_k \\ g_{i,z} - z_k \end{pmatrix} + 2 \begin{pmatrix} -q_{k,y}^2 - q_{k,z}^2 & q_{k,x}q_{k,y} - q_{k,r}q_{k,z} & q_{k,r}q_{k,y} + q_{k,x}q_{k,z} \\ q_{k,r}q_{k,z} + q_{k,x}q_{k,y} & -q_{k,x}^2 - q_{k,z}^2 & q_{k,y}q_{k,z} - q_{k,r}q_{k,x} \\ q_{k,x}q_{k,z} - q_{k,r}q_{k,y} & q_{k,r}q_{k,x} + q_{k,y}q_{k,z} & -q_{k,x}^2 - q_{k,y}^2 \end{pmatrix} \begin{pmatrix} g_{i,x} - x_k \\ g_{i,y} - y_k \\ g_{i,z} - z_k \end{pmatrix} \end{aligned} \quad (27)$$

Figure 10. The real uncertainty of an inverse sensor model for bearing-only observations are cone-shaped, as shown in the four individual observations shown in (a) and (c). In each situation, the estimation of the potential landmark location from two observations can be obtained by fusing both cone-like shapes, leading to the pdfs of (b) and (d). A larger parallax, as in (a) – (b), leads to a pdf closer to a Gaussian, while a reduced parallax, as in (c)—(d), makes the Gaussian a poorer approximation and favors the inverse depth parameterization. (e) Confidence intervals for an inverse-depth parameterization, compared to the actual cone-like pdf which is being approximated. (f) The convention used in the text regarding the axes of local coordinates with respect to the camera.



when given in relative (local) coordinates $\mathbf{M}'_{k,i} = \begin{pmatrix} l_{i,x} & l_{i,y} & l_{i,z} \end{pmatrix}^T$ with respect to the camera location \mathbf{x}_k —please refer to Figure 10f for the axes convention. These local coordinates can be computed from the landmark global coordinates $\mathbf{M}_i = \begin{pmatrix} g_{i,x} & g_{i,y} & g_{i,z} \end{pmatrix}^T$ and the quaternion-based representation of the camera pose as shown in Box 1.

The Inverse Sensor Model

Let us clearly define first the sensor observation model for this kind of observations. It consists of an observation function $\mathbf{h}(\cdot)$ which takes as input the relative position of the landmark ($\mathbf{M}'_{k,i}$) and, via Equation 26 gives us the pair of observed angles:

$$\mathbf{z}_{k,i} = \begin{pmatrix} \alpha_{k,i} \\ \beta_{k,i} \end{pmatrix} = \mathbf{h}(\mathbf{M}'_{k,i}) + \mathbf{n}_{k,i} \quad (28)$$

As usual, we assume an additive zero-mean Gaussian noise $\mathbf{n}_{k,i} \sim N(\mathbf{0}, \mathbf{R})$ on this observation, which in this case models the uncertainty in the image feature detectors due to the discrete nature of pixels, possibly blurred images, etc. As a matter of fact, given the wide-spread application of bearing-only observations in computer vision, the observation model can be (and usually is) directly formulated in terms of pixel coordinates, which are much closer to the vision-based front-end algorithms than the pair of angles in the equations above (Nüchter, 2009). In this form, the two components of the observation are the pixel coordinates $\begin{pmatrix} u_{k,i} & v_{k,i} \end{pmatrix}^T$, computed as:

$$\mathbf{z}_{k,i} = \begin{pmatrix} u_{k,i} \\ v_{k,i} \end{pmatrix} = \mathbf{h}(\mathbf{M}'_{k,i}) + \mathbf{n}_{k,i}$$

with:

$$\mathbf{h}(\mathbf{M}'_{k,i}) = \begin{pmatrix} c_x + f_x \frac{l_{i,x}}{l_{i,z}} \\ c_y + f_y \frac{l_{i,y}}{l_{i,z}} \end{pmatrix} \quad (29)$$

where c_x and c_y and the pixel coordinates of the camera optical center and f_x and f_y are both the focal distance, measured in units of horizontal and vertical pixels, respectively. This equation above is called the camera *pinhole projective model*, and can be further improved to reflect real cameras by the inclusion of distortion parameters. For the introductory nature of this book, no more details will be given here on camera projective geometry, thus we recommend the interested reader to consult the rich existing literature (Mikhail, Bethel, & McGlone, 2001; Hartley & Zisserman, 2003).

What is really significant for the present discussion is the realization that bearing-only observations reduce the dimensionality of the observed landmarks from three (i.e. its spatial location with respect to the observer) to only two (i.e. the two pixel coordinates). Therefore, there is one degree of freedom, which is irremediably lost: the depth of the landmark. This has one practical consequence of paramount importance: if we were to try finding out the inverse sensor model just as with range-bearing sensors, we would not find any such function $\mathbf{h}^{-1}(\cdot)$ that maps pairs of pixel coordinates into three dimensional coordinates, simply because such a bijective relationship does not exist.

Still, under a probabilistic viewpoint this is not a limitation. A probabilistic inverse sensor model can assign a uniform distribution to the unknown depth, starting at the camera location and extending up to some arbitrarily large maximum distance. Combining that depth uncertainty with

the uncertainty of the two angles (or equivalently, of the pixel coordinates) we end up with a pdf for the potential location of a landmark upon its first observation which is cone-like shaped, as illustrated in Figure 10.

The fundamental problem with initializing a landmark from a bearing-only observation relies on that uncommon, cone-like shape of the uncertainty, which should be modeled. A Gaussian in three-dimensional space whose variables are the three Cartesian coordinates (x_i, y_i, z_i) makes quite a poor job in approximating the real shape of the inverse sensor model. As a workaround, tricks have been proposed in the literature such as delaying the insertion of landmarks in the Bayesian filter until their location uncertainty has been reduced, with other auxiliary Bayesian filters, and can be appropriately modeled as Gaussian (Davison, 2003). That approach discards valuable information until each landmark converges, which could help localizing the camera when these maps are used within a SLAM framework; also, it discards distant landmarks (“features at the infinity”) which are known to help estimating the orientation of the camera.

In order to avoid all those disadvantages, a different and smart solution was proposed in the literature (Civera, Davison, & Montiel, 2008): instead of parameterizing landmarks with their three Cartesian coordinates $\mathbf{M}_i = (g_{i,x} \ g_{i,y} \ g_{i,z})^T$, one can explicitly store the first pose from which they were first observed $(o_{x_i}, o_{y_i}, o_{z_i})$, the associated observation direction (two angles θ_i and ϕ_i) and the inverse of the depth (ρ_i) from the observing point. These parameters are related to the Cartesian coordinates by means of:

$$\mathbf{M}_i(\mathbf{m}_i) = \begin{pmatrix} o_{x_i} \\ o_{y_i} \\ o_{z_i} \end{pmatrix} + \frac{1}{\rho_i} \begin{pmatrix} \cos \phi_i \sin \theta_i \\ -\sin \phi_i \\ \cos \phi_i \cos \theta_i \end{pmatrix} \quad (30)$$

At the cost of a clear over-parameterization, it has been demonstrated that a Gaussian distribution over these six parameters resembles the cone-like shape of the actual uncertainty—refer to the examples in Figure 10. Historically, this was the first parameterization that succeeded in unifying the representation of close and distant features in such a way that both kinds could be used simultaneously for localization, mapping, and SLAM.

Therefore, we can summarize the probabilistic inverse sensor model based on the inverse-depth parameterization as follows. Each landmark will be represented as a vector:

$$\mathbf{m}_i = (o_{x_i} \ o_{y_i} \ o_{z_i} \ \theta_i \ \phi_i \ \rho_i)^T \quad (31)$$

over which we define a Gaussian distribution such that $\mathbf{m}_i \sim N(\bar{\mathbf{m}}_i, \Sigma_{\mathbf{m}_i})$, with:

$$\underbrace{p(\mathbf{z}_k | \mathbf{m}_i, \mathbf{x}_k)}_{\text{Inverse sensor model}} = N(\mathbf{m}_i; \bar{\mathbf{m}}_i, \Sigma_{\mathbf{m}_i})$$

$$\rightarrow \begin{cases} \bar{\mathbf{m}}_i = (\bar{o}_{x_i} \ \bar{o}_{y_i} \ \bar{o}_{z_i} \ \bar{\theta}_i \ \bar{\phi}_i \ \bar{\rho}_i)^T \\ \Sigma_{\mathbf{m}_i} = \begin{pmatrix} \Sigma_x & & & & & \\ & \sigma_\theta^2 & 0 & 0 & & \\ & 0 & \sigma_\phi^2 & 0 & & \\ & 0 & 0 & \sigma_\rho^2 & & \end{pmatrix} \end{cases} \quad (32)$$

The first five values in the mean vector $\bar{\mathbf{m}}_i$ directly correspond to the camera position $(o_{x_i} \ o_{y_i} \ o_{z_i})^T$ and to the direction in which the landmark was observed, with respect to the global frame of reference. Within the covariance matrix, the first 3×3 diagonal block represents the uncertainty in the camera location. Since in this chapter we are assuming it is perfectly known, all the matrix entries become zeros. However, when we will revisit bearing-only observations

in the context of SLAM in chapter 9, this matrix shall be set to the actual uncertainty of the camera pose. Notice how the covariance of the camera location and the three other parameters (the pair of 3×3 off-diagonal symmetric blocks) is exactly zero. This follows from the realistic assumption of statistical independence of the sensor noises (θ_i, ϕ_i) and the depth (ρ_i) with respect to the uncertainty of the camera pose.

The sixth value in the mean vector (i.e., the inverse depth $\bar{\rho}_i$) and the three standard deviations σ_θ , σ_ϕ and σ_ρ are free parameters to be settled heuristically. Firstly, σ_θ and σ_ϕ can be adjusted to account for the inaccuracies in the bearing angles, typically, due to errors in the interest keypoint detector on the images. And secondly, $\bar{\rho}_i$ and its associated standard deviation σ_ρ can be determined by defining some arbitrary minimum and maximum depths (d_0 and d_1 , respectively) as the desired limits of the (for example) 99.7% confidence interval of the modeled uncertainty. Then, $\bar{\rho}_i$ and σ_ρ can be found from: (using the equivalence of ± 3 sigmas for a 99.7% confidence interval of a one-dimensional Gaussian distribution)

$$\begin{cases} d_0 = \frac{1}{\bar{\rho}_i + 3\sigma_\rho} \\ d_1 = \frac{1}{\bar{\rho}_i - 3\sigma_\rho} \end{cases}$$

(and solving for the desired parameters)

$$\rightarrow \begin{cases} \bar{\rho} = \frac{1}{2} \left(\frac{1}{d_0} + \frac{1}{d_1} \right) \\ \sigma_\rho = \frac{1}{6} \left(\frac{1}{d_0} - \frac{1}{d_1} \right) \end{cases} \quad (33)$$

Recursive Bayesian Estimation

Once the three-dimensional landmark has been initialized in the map with an inverse-depth parameterization, information from subsequent observations must be fused to reduce its uncertainty. For this kind of observations, as soon as the landmark is observed from a slightly different direction, which in computer vision is called the observation *parallax*, the uncertainty in its depth will be drastically reduced.

Since we employ one Gaussian distribution for each landmark, the particular implementation of Equation 15 of choice for this case is also an Extended Kalman filter. Due to the assumption of a static map (i.e., landmarks do not move on their own), the recursive form of the EKF in chapter 7's Equation 44 simplifies in our case to:

$$\begin{aligned} \bar{\mathbf{m}}_i &\leftarrow \bar{\mathbf{m}}_i + \mathbf{K}_{k,i} (\mathbf{z}_k - \mathbf{h}_{k,i}) \\ \Sigma_{\mathbf{m}_i} &\leftarrow (\mathbf{I} - \mathbf{K}_{k,i} \mathbf{H}_{k,i}) \Sigma_{\mathbf{m}_i} \end{aligned}$$

with the Kalman gain matrix being

$$\mathbf{K}_{k,i} = \Sigma_{\mathbf{m}_i} \mathbf{H}_{k,i}^T (\mathbf{R} + \mathbf{H}_{k,i} \Sigma_{\mathbf{m}_i} \mathbf{H}_{k,i}^T)^{-1} \quad (34)$$

The Jacobian matrix $\mathbf{H}_{k,i}$ contains the derivatives of the function $\mathbf{h}(\cdot)$ defined in Equations 27 and 28 respect to the six parameters of the landmark. Due to the complexity of the transformation involved, here is more convenient to apply the chain rule in order to evaluate this Jacobian. Recalling that we denote the parameterization of a landmark as \mathbf{m}_i , its Cartesian coordinates in the global frame of reference as \mathbf{M}_i and its local coordinates with respect to the camera as $\mathbf{M}'_{k,i}$, we can write:

$$\begin{aligned} \mathbf{H}_{i,t} &= \frac{\partial \mathbf{h}(\mathbf{x}_i, \mathbf{m}_i)}{\partial \mathbf{m}_i} = \\ &= \frac{\partial \mathbf{h}(\mathbf{M}'_i)}{\partial \mathbf{M}'_{k,i}} \frac{\partial \mathbf{M}'_{k,i}(\mathbf{M}_i)}{\partial \mathbf{M}_i} \frac{\partial \mathbf{M}_i(\mathbf{m}_i)}{\partial \mathbf{m}_i} \end{aligned} \quad (35)$$

where we approached the complex projection of a landmark into an observation as a sequence of three, more handy steps: (1) $\mathbf{M}_i(\cdot)$ converts the set of inverse-depth parameters into their corresponding global coordinates, as shown in Equation 30; (2) $\mathbf{M}'_{k,i}(\cdot)$ is in charge of transforming them into local coordinates in the camera frame of reference as specified in Equation 27; and (3) the observation model $\mathbf{h}(\cdot)$ finally projects the landmark into pixel coordinates according to Equation 29. Recall that all these partial Jacobians, which are straightforward to obtain, must be evaluated at the latest estimated value of each of the involved variables.

It is worth mentioning that the advantages of the inverse-depth parameterization come only at the cost of employing six components for each landmark instead of the minimum of three as would correspond to a simple point in the space. Assuming an EKF implementation with a computational complexity cubic with the dimensionality, those extra parameters imply multiplying the execution time by eight. Therefore, it comes as no surprise the existence of proposals in the literature that recover the simpler (x, y, z) parameterization as long as the reduction in the depth uncertainty makes it an acceptable approximation (Civera, Davison, & Montiel, 2008).

7. BAYESIAN ESTIMATION OF LANDMARK MAPS: RANGE-ONLY SENSORS

We finally arrive at the third type of landmark observations: those that only measure a range or

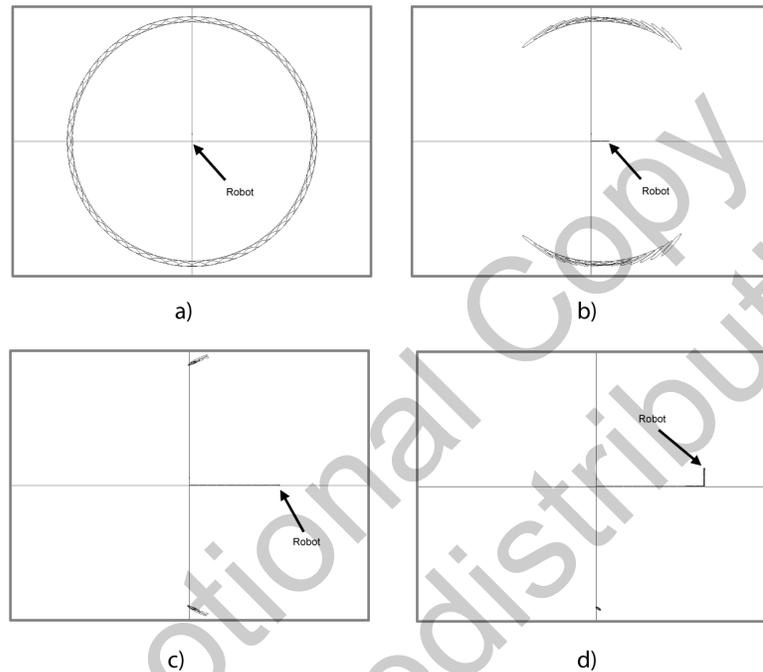
distance from the sensor to one or a set of fixed points in the environments. As discussed in chapter 2 section 9, Range-Only (RO) sensors are commonly employed in submarine robotics but some practical applications have also appeared during the last decade for ground robots. In contrast to the *passive* nature of landmarks studied in previous sections, we deal here with observations that measure purposely-placed *active* devices in most cases, emitting radio or ultrasonic signals, thus, for differentiation we will refer to them as *beacons*. Beacons, landmarks, and features are all names of interchangeable entities in our present context of object-based mapping.

Disregarding the specific device utilized to obtain RO measurements, we will approach the problem in an abstract way focusing on the properties that all these sensors share. Firstly, they are typically able to detect several beacons simultaneously and to identify each one unequivocally by means of some sort of identification code, which is transmitted wirelessly: with RO sensors, we can avoid the hard problem of data association. Consequently, a RO observation for one beacon simply contains a range value, that is, $\mathbf{z}_k = (r_k)$.

Secondly, another fundamental characteristic of RO sensors is that they naturally lead to multiple hypotheses about the location of the beacons. To illustrate this point, refer to the example in Figure 11, where a robot moving in a straight line makes three range measurements for a beacon with location \mathbf{m}_i . In a two-dimensional approach, each range observation tells us that the beacon must lie around a circle centered at the robot position \mathbf{x}_k with a radius of r_k . Under our probabilistic viewpoint, observations actually are assumed to be corrupted with an additive zero-mean Gaussian noise $n_k \sim N(0, \sigma_n^2)$, that is:

$$r_k = \|\mathbf{m}_i - \mathbf{x}_k\| + n_k \quad (36)$$

Figure 11. An example we have simulated of map building from range-only observations. (a) The first time the robot detects a landmark, it is introduced in the map via the inverse sensor model. In this case we employ a sum of Gaussians approximation to the actual, ring-shaped pdf (see text). (b) – (d) Subsequent observations further reduce the uncertainty in the landmark location. Notice how two potential locations for the landmark remain until the robot turns in (d) breaking the symmetry that existed up to that instant.



which suggests us to consider instead a “thick ring”-shaped pdf instead of a unidimensional circumference. One of such rings is depicted in the figure, centered at the robot pose where an observation was made. Intuitively speaking (the exact Bayesian approach is discussed below), the most likely locations at which the beacon may be actually located are those where the different rings intersect with each other.

Unlike range-bearing and bearing-only observations, in this case it is not only possible but quite common that the explained rings end up marking several separate locations as the likely location of the beacon. Therefore, probabilistic mapping with RO observations should be addressed with multimodal distributions, a requisite somewhat unique in mapping and SLAM.

The Inverse Sensor Model

Provided the sensor model of Equation 36 we find that just like with bearing-only observations, these ones also reduce the dimensionality of the observed beacon location from two (if we assume a planar map) to only one, the range measurement.

Therefore, we face again the non-existence of an inverse sensor function. A probabilistic inverse sensor model is easy to devise instead, since all we need is a pdf that assigns each potential beacon location \mathbf{m}_i a likelihood according to Equation 36 and the known distribution of the additive n_t noise:

$$\underbrace{p(\mathbf{z}_k | \mathbf{m}_i, \mathbf{x}_k)}_{\text{Inverse sensor model}} = \frac{1}{\sigma_n \sqrt{2\pi}} \exp \left\{ -\frac{1}{2} \left(\frac{|\mathbf{m}_i - \mathbf{x}_k| - r_k}{\sigma_n} \right)^2 \right\} \quad (37)$$

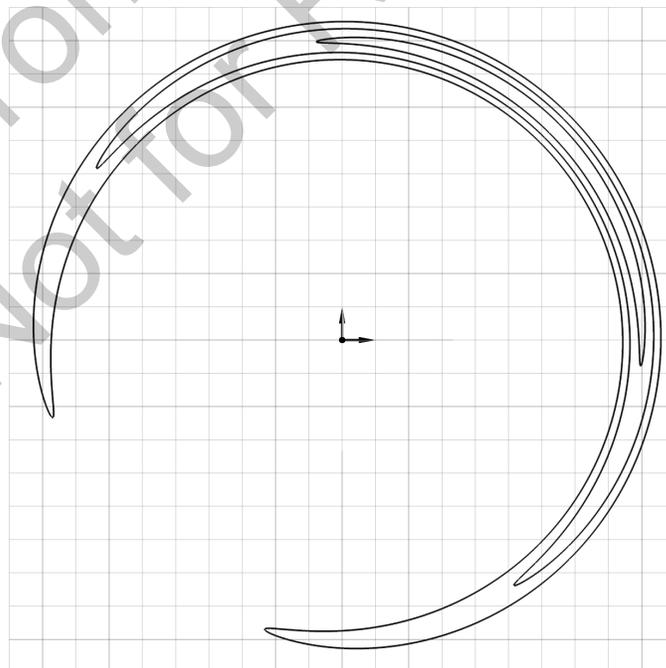
Notice that the thick ring displayed in figure 11a closely corresponds to this pdf, where the thickness is associated to the noise variance σ_n^2 . The problem of this distribution is that it cannot be written as a Gaussian in terms of the (x, y) coordinates of the beacon \mathbf{m}_i .

Among others, two workarounds have been proposed in the literature for approximating the ring-shape distribution in Equation 37 while still being able to perform Bayesian filtering. The first one resembles the change of landmark parameterization explained above for bearing-only sensors. In this method, over-parameterization of the beacon position with the location where

the robot made its first observation plus a pair of polar coordinates that identify a global direction and a distance from that robot pose to the beacon is used (Djugash, Singh, & Grocholsky, 2008). As illustrated in Figure 12, this idea allows representing each beacon with just one Gaussian over the five abovementioned parameters. However, in order to deal with multi-modality one would need to introduce additional heuristics to determine when and how to split the Gaussian into two or more pdf modes.

The second approach, introduced in (Blanco, González, & Fernández-Madriral, 2008b) and already employed in the example of Figure 11, consists of approximating Equation 37 with a Sum Of Gaussians (SOG) over the ordinary Cartesian coordinates, such that:

Figure 12. An approximation of the inverse sensor model of RO observations by means of a single Gaussian, in the parameter space of range-bearing with respect a robot pose. In this example, three confidence intervals are shown, corresponding to 1σ , 2σ , and 3σ . The observation has a mean orientation of 45° and a mean distance of 9m, while the associated standard deviations is 50° and 0.2m, respectively. Notice how a large enough uncertainty in the orientation could approximate well a complete ring-like pdf.



$$p(\mathbf{z}_k | \mathbf{m}_i, \mathbf{x}_k) = \sum_j \omega_j N(\mathbf{m}_i; \bar{\mathbf{m}}_i^j, \Sigma_{\mathbf{m}_i}^j),$$

$$\text{with } \sum_j \omega_j = 1$$
(38)

where ω_j are the weights of each Gaussian mode. Upon its first observation, all the SOG modes have equal weights and are distributed in such a way that they approximate the thick ring-shape of the actual inverse sensor pdf. At the cost of maintaining several Gaussians for each beacon, the clear advantage of this method is its natural capability of representing an arbitrary number of modes just by readjusting the parameters of each SOG mode and their weights, which can be achieved by Bayesian filtering.

Recursive Bayesian Estimation

If beacons were modeled according to the first approach above, namely the over-parameterization that includes polar coordinates, we would have a single Gaussian for each beacon, which should be updated with subsequent observations. A natural implementation here is to employ the EKF as already explained above for the other types of landmarks. However, the strong non-linearities present in this parameterization would render the UKF, explained in chapter 7 section 3, a better candidate.

In the case that beacons are modeled as a weighted SOG, we would face a new problem not dealt with yet in this book. Nonetheless, it can be easily shown that updating a SOG with a new observation $\mathbf{z}_k = (r_k)$ can be easily realized through the following three steps:

1. The weights ω_j must be updated to reflect how well each Gaussian mode explains the observation. Quantitatively, this implies evaluating the likelihood of the observation against the prediction of the j -th mode:

$$\omega_j \leftarrow \omega_j N(\mathbf{z}_k; \bar{h}_i^j, \sigma_i^{2j})$$

with the mean obtained through the sensor model in Equation 36

$$\bar{\mathbf{h}}_i^j = |\bar{\mathbf{m}}_i^j - \mathbf{x}_k|$$

and the covariance got through first-order Taylorseries linearization (see chapter 3 section 8):

$$\sigma_i^{2j} = \mathbf{H}_i^j \Sigma_{\mathbf{m}_i}^j \mathbf{H}_i^{jT} + \sigma_n^2$$
(39)

2. Weights are then renormalized such that they sum the unity, in order to assure that the SOG is kept as a pdf.
3. The parameters of each Gaussian mode $N(\mathbf{m}_i; \bar{\mathbf{m}}_i^j, \Sigma_{\mathbf{m}_i}^j)$ are updated following the standard EKF equations—refer for example to Equation 34. Unlike with the polar-coordinate parameterization, the more complex UKF is not required here since the uncertainty of each SOG mode is quite small in comparison to the entire ring-like pdf, and assuming linearity in the observation of one mode is perfectly acceptable.

Notice that this technique exploits the freedom, assumed in this chapter, in modeling of pdfs for each map element independently: each beacon may be represented by a different number of SOG modes. In practice, the weights of many modes will soon become negligible after a few observations. Thus, it becomes convenient to discard those SOG nodes that have weights below some certain threshold. After the robot moves around, the number of modes will reduce, dynamically adapting itself to the actual uncertainty in the beacon location, as shown with the example of Figure 11c.

8. OTHER MAP BUILDING ALGORITHMS

Previous sections have addressed the problem of initializing and updating the most common probabilistic representations of metric maps from sequences of sensor readings: grid maps and feature or landmark maps. In the following, we study other mapping algorithms, which either do not rely on a probabilistic foundation or are not so widely spread.

Point Maps

There exist two main families of sensors which are used to generate point maps: laser range scanners and 3D cameras—reviewed in chapter 2 sections 7 and 8, respectively. As already mentioned above, point maps are among the most “sub-symbolic” representations, in the sense that very little (or none) post-processing is required for the raw sensor readings to be merged within the maps.

In its most basic approach, maintaining a point map could be as simple as keeping a list of 2D or 3D point coordinates and appending new ones with each sensor reading, taking into account the corresponding geometrical transformation for the pose of the sensor within the global frame of reference. However, given that state-of-the-art range scanners provide dozens of completed scans per second, if we were to insert all of them in a point map, it would grow as rapidly as to render any localization or SLAM algorithm useless after a few seconds of operation. This is why all practical implementations discard most of the 2D or 3D range scans provided by the sensors. This may seem a waste of information, but, in fact, consecutive scans are highly redundant and hence provide very little new relevant information. One alternative for not dropping scans is to *fuse* points which are established to correspond to previously observed points, which can drastically reduce the

number of points in a map at the expense of introducing the complexities and potential mistakes of data association.

Typical heuristics to determine which scans to keep are: (1) the usage of a fixed subsample rate (e.g., keeping only one scan out of ten) and (2) discarding all the captured scans until the robot moves or turns more than a certain threshold distance or angle, respectively. Selecting the parameters of any of these two heuristics is a critical step since they heavily condition the performance of any posterior localization or SLAM method, which has to work on the point map. Unfortunately, and to the best of our knowledge, this topic has not been properly addressed in the scientific literature, thus it requires the experience of the operator and some doses of manual tuning with trials and errors.

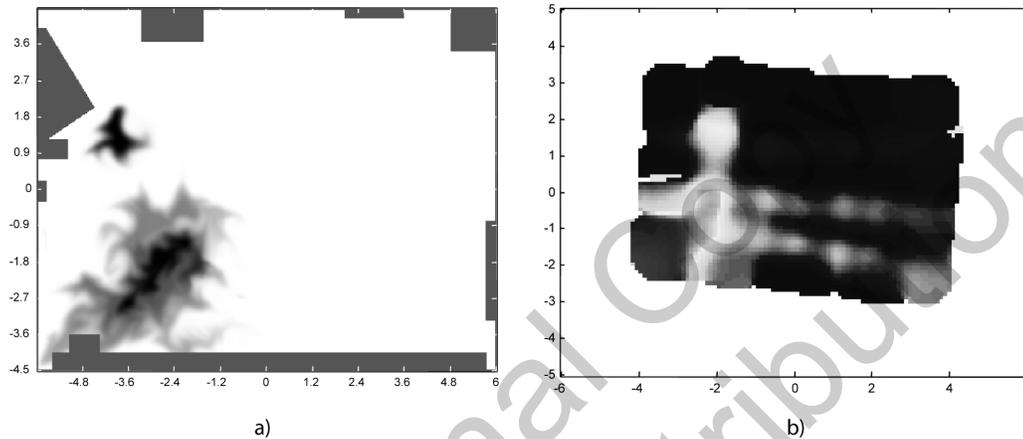
Continuous Markov Random Fields

Recall that we loosely defined a random field in section 2 as “a spatial domain where each point is associated a random variable.” An example of a discrete random field has already been provided with occupancy grid maps. The property of interest at each location of the space was there the occupancy (or freeness), a discrete random variable with only two possible outcomes.

There exists, however, a variety of other properties we might be interested in while building a map with a mobile robot, and most of them are continuous magnitudes. For instance, we could map the height at each (x, y) position for an outdoor environment; or the temperature along the inside of an office building; or the different concentration of certain gases within a factory facility.

All these are examples of physical properties that vary from one point to the other—we will leave the dynamics aside and assume static environments, as usual. However, it is typical that the spatial variations of the magnitude of interest

Figure 13. Example of Kernel-based mapping with gas-concentration sensors mounted in a mobile robot of our lab. (a) The actual gas concentration in a simulation environment, from which a sequence of gas sensor observations have been simulated in order to build the gas map shown in (b). Notice that typical hurdles found in real sensors, such as a high response time, were taken into account in this simulation, hence the poor resolution attainable in the reconstruction, a common problem found in most kinds of gas maps (Images courtesy of Javier González Monroy, University of Málaga).



are somewhat limited, with no abrupt changes between two nearby points, or, in more technical terms, a probability distribution could exist that, *conditioned* on any point of the environment, satisfactorily models the properties of its surroundings. In general, although the magnitude of interest for mapping may tend to remain constant around any given point, the variance of this prediction grows as we get farther from any central point. It is in those cases where the magnitude of interest satisfies these generic requirements when a map of the environment could be modeled as a Markov Random Field (MRF) (Winkler, 1995) (see Figure 13).

As already discussed in previous chapters, a MRF is a random field in which random variables can be organized on a discrete lattice and where the Markov condition holds between adjacent variables. This is in contrast to the common application of the Markov condition to sequences of r.v. that are consecutive in time. If we denote the r.v. for the property of interest at some arbitrary

map coordinates (i, j) as $m_{i,j}$, and define the finite set of all adjacent coordinates as N , the *spatial* Markov property reads:

$$p\left(m_{i,j} \mid \left\{m_p\right\}_{p \in N}, \left\{m_p\right\}_{p \notin N}\right) = p\left(m_{i,j} \mid \left\{m_p\right\}_{p \in N}\right) \quad (40)$$

that is: a map element, conditioned on all its neighbors, is conditionally independent of the rest of the map. Moreover, if that conditional distribution can be properly modeled as a Gaussian, the MRF becomes a *Gaussian Markov Process*. Due to their general applicability, Gaussian Processes (GP) have received an immense attention in the research community (Rasmussen & Williams, 2006).

There exist different map building methods relying on the assumption of a MRF or GP-like map. We find classical references in the branch of geology and mining literature called *geostatistics*. In particular, dating back to the 1950s, we find

the Kriging technique (Matheron, 1963) which applies what later on were to be called GP to interpolate geological measurements and therefore reconstruct non-measured areas. Closer to modern mobile robotics, we can find algorithms for mapping the concentration of gases relying on a simplified estimation method known as the Kernel method (Lilienthal & Duckett, 2004). A more direct application of GP to mapping was reported in Stachniss, Plagemann, Lilienthal, and Burgard (2008).

Pose Constraint Maps

As introduced in section 2, relational maps are all those representations that model some kind of relationship existing between the mapped elements instead of, for example, their global coordinates in some fixed frame of reference. Among this family of models, Pose Constraint Maps (PCMs) are probably the most representative nowadays due to their popularity.

Building a PCM implies creating and populating a directed graph, a computational abstract data type that has already been explained, instantiated such that its nodes represent robot poses (in an abstract sense, since the numeric values of those poses are unknowns) and its arcs represent the relative coordinates of poses with respect to each other. Typically, a PCM is built incrementally as the robot moves and explores its environment. New nodes (also called *keyframes*) are created wherever some heuristic is fulfilled, such as when it is reached a minimum distance from the last node. It is convenient to store (or *annotate*) the most recent robot observations (e.g. laser range scans, images from its cameras, etc.) within each node in order to provide it with some metrical information suitable to the determination of potential arcs. Indeed, for each newly created node, arcs should also be defined between nodes for which their relative spatial pose could be deduced from the annotated sensory data. The specific approach

for obtaining such relative poses strongly depends on the employed sensors. As an example, the ICP matching algorithm, explained in chapter 6, could be used for the common case of working with laser range scanners. In order to cope with uncertainty, arcs often hold probability distributions of those relative poses, typically in the form of a Gaussian distribution.

Although the apparent simplicity of such a graph representation is appealing, its practical realization reveals two important hurdles, which are still an active area of research. Firstly, notice how global coordinates do not appear anywhere in the discussion above: all the map building process relies entirely on node-to-node relative coordinates. If for some reason the map is needed in a common frame of reference, computing the global coordinates for each node may become not an easy task at all. To illustrate the challenge, consider the map in Figure 5a which contains many closed loops in the graph topology. The global coordinates for that figure were obtained by arbitrarily fixing one of the nodes as the origin of coordinates, then creating a spanning tree from that node to all the others, e.g. using the Dijkstra's algorithm (Dijkstra, 1959). In a PCM with a tree structure, the global pose of each node is the simple accumulation of all the edges from the root node at the origin. However, there exist obvious mismatches or inconsistencies in the coordinates so obtained—which correspond to the arcs crossing wide gaps in the figure. The bunch of techniques generically dubbed as *Graph SLAM* (and reviewed in chapter 10 section 3) precisely address the problem of estimating a set of *consistent* global coordinates from PCMs with arbitrarily topologies that may include any number of loops.

The second problem of these relational maps is related to the creation of arcs when the robot closes a loop. This is the well-known *loop-closure* problem and consists in reliably detecting such a situation in order to define the correspond-

ing arc with a relative pose obtained from the metrical registration of the observations in each node. Some approaches to efficiently detect loop closures include geometrical information, that is, firstly solving the graph for its global coordinates, then considering all the potential matches of the latest node with all its neighbors. When working with images, other authors propose purely topological methods where node-to-node pairings are established uniquely from the detection of similar visual features (Cummins & Newman, 2008). We will further explore the process of building maps of this kind in chapter 10.

REFERENCES

- Anderson, M. L. (2003). Embodied cognition: A field guide. *Artificial Intelligence*, 149, 91–130. doi:10.1016/S0004-3702(03)00054-7
- Arras, K. O., Castellanos, J. A., Schilt, M., & Siegwart, R. (2003). Feature-based multi-hypothesis localization and tracking using geometric constraints. *Robotics and Autonomous Systems*, 44, 41–53. doi:10.1016/S0921-8890(03)00009-5
- Birk, A., & Carpin, S. (2006). Merging occupancy grid maps from multiple robots. *Proceedings of the IEEE*, 96(7), 1384–1397. doi:10.1109/JPROC.2006.876965
- Blanco, J. L., Fernández-Madrigo, J. A., & González, J. (2008). Towards a unified Bayesian approach to hybrid metric-topological SLAM. *IEEE Transactions on Robotics*, 24(2), 259–270. doi:10.1109/TRO.2008.918049
- Blanco, J. L., Fernández-Madrigo, J. A., & González, J. (2008b). Efficient probabilistic range-only SLAM. In *Proceedings of the IEEE/RSJ 2008 International Conference on Intelligent Robots and Systems*, (pp. 1017-1022). IEEE Press.
- Blanco, J. L., González, J., & Fernández-Madrigo, J. A. (2009). Subjective local maps for hybrid metric-topological SLAM. *Robotics and Autonomous Systems*, 57(1), 64–74. doi:10.1016/j.robot.2008.02.002
- Brooks, R. A. (1982). Solving the find-path problem by good representation of free space. *IEEE Systems, Man and Cybernetics*, 13, 190–197.
- Brooks, R. A. (1991). Intelligence without representation. *Artificial Intelligence Journal*, 47, 139–159. doi:10.1016/0004-3702(91)90053-M
- Choset, H., & Burdick, J. W. (1995). Sensor based planning, part 1: The generalized Voronoi graph. In *Proceedings of the IEEE International Conference on Robotics and Automation*. IEEE Press.
- Civera, J., Davison, A., & Montiel, J. (2008). Inverse depth parametrization for monocular SLAM. *IEEE Transactions on Robotics*, 24(5), 932–945. doi:10.1109/TRO.2008.2003276
- Coradeschi, S., & Saffiotti, A. (2003). An introduction to the anchoring problem. *Robotics and Autonomous Systems*, 43(2-3), 85–96. doi:10.1016/S0921-8890(03)00021-6
- Cummins, M., & Newman, P. (2008). Accelerated appearance-only SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation*, (pp. 1828-1833). IEEE Press.
- Davison, A. (2003). Real-time simultaneous localization and mapping with a single camera. In *Proceedings of the International Conference on Computer Vision*, (pp. 1403-1410). IEEE.
- Davison, A. J., Reid, I. D., Molton, N. D., & Stasse, O. (2007). MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6), 1052–1067. doi:10.1109/TPAMI.2007.1049

- Dijkstra, E. W. (1959). A note on two problems in connection with graphs. *Numerische Mathematik, 1*, 269–271. doi:10.1007/BF01386390
- Djugash, J., Singh, S., & Grocholsky, B. (2008). Decentralized mapping of robot-aided sensor networks. In *Proceedings of the IEEE International Conference on Robotics and Automation*. IEEE Press.
- Duda, R. O., & Hart, P. E. (1972). Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM, 15*(1), 11–15. doi:10.1145/361237.361242
- Fabrizi, E., & Saffiotti, A. (2000). Augmenting topology-based maps with geometric information. In *Proceedings of the 6th International Conference on Intelligent Autonomous Systems*. IEEE.
- Fernández-Madrigal, J. A., & González, J. (2002). Multi-hierarchical representation of large-scale space: Applications to mobile robots. *Intelligent Systems, Control and Automation: Science and Engineering, 24*.
- Fox, D. (2003). *The Intel lab dataset*. Retrieved Mar 1, 2012, from http://cres.usc.edu/radishrepository/view-one.php?name=intel_lab
- Galindo, C., Fernández-Madrigal, J. A., & González, J. (2007). Multiple abstraction hierarchies for mobile robot operation in large environments. *Studies in Computational Intelligence, 68*.
- Galindo, C., Fernández-Madrigal, J. A., González, J., & Saffiotti, A. (2008). Robot task planning using semantic maps. *Robotics and Autonomous Systems, 56*(11), 955–966. doi:10.1016/j.robot.2008.08.007
- Grisetti, G., Grzonka, S., Stachniss, C., Pfaff, P., & Burgard, W. (2007). Efficient estimation of accurate maximum likelihood maps in 3D. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, (pp. 3472–3478). IEEE Press.
- Grisetti, G., Stachniss, C., Grzonka, S., & Burgard, W. (2007). A tree parameterization for efficiently computing maximum likelihood maps using gradient descent. In *Proceedings of Robotics: Science and Systems (RSS)*. IEEE.
- Guivant, J., & Nebot, E. (2001). Optimization of the simultaneous localization and map building algorithm for real-time implementation. *IEEE Transactions on Robotics and Automation, 17*(3), 242–257. doi:10.1109/70.938382
- Harnad, S. (1990). The symbol grounding problem. *Physica D. Nonlinear Phenomena, 42*, 335–346. doi:10.1016/0167-2789(90)90087-6
- Hartley, R., & Zisserman, A. (2003). *Multiple view geometry in computer vision*. Cambridge, UK: Cambridge University Press.
- Johnson, A. E., & Hebert, M. (1999). Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 21*(5), 433–449. doi:10.1109/34.765655
- Klein, G., & Murray, D. (2007) Parallel tracking and mapping for small AR workspaces. In *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*. IEEE Press.
- Konolige, K. (2005). SLAM via variable reduction from constraint maps. In *Proceedings of the IEEE International Conference on Robotics and Automation*. IEEE Press.
- Konolige, K., Bowman, J., Chen, J. D., Michelich, P., Calonder, M., Lepetit, V., & Fua, P. (2010). View-based maps. *The International Journal of Robotics Research, 29*(8), 941–957. doi:10.1177/0278364910370376

- Konolige, K., Fox, D., Limketkai, B., Ko, J., & Stewart, B. (2003). Map merging for distributed robot navigation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, (pp. 212-227). IEEE Press.
- Kuipers, B. J. (1977). *Representing knowledge of large-scale space*. (Doctoral Dissertation). Massachusetts Institute of Technology. Cambridge, MA.
- Kuipers, B. J. (1978). Modeling spatial knowledge. *Cognitive Science*, 2, 129–153. doi:10.1207/s15516709cog0202_3
- Kuipers, B. J. (2000). The spatial semantic hierarchy. *Artificial Intelligence*, 119, 191–233. doi:10.1016/S0004-3702(00)00017-5
- Latombe, J. C. (1991). *Robot motion planning*. Boston, MA: Kluwer Academic Publishers.
- Leonard, J. J., & Durrant-Whyte, H. F. (1991). Mobile robot localization by tracking geometric beacon. *IEEE Transactions on Robotics and Automation*, 7(3), 376–382. doi:10.1109/70.88147
- Lilienthal, A., & Duckett, T. (2004). Building gas concentration gridmaps with a mobile robot. *Robotics and Autonomous Systems*, 48(1), 3–16. doi:10.1016/j.robot.2004.05.002
- Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2), 91–110. doi:10.1023/B:VISI.0000029664.99615.94
- Lozano-Pérez, T., & Wesley, M. A. (1979). An algorithm for planning collision-free paths among polyhedral obstacles. *Communications of the ACM*, 22(10), 560–570. doi:10.1145/359156.359164
- Lu, F., & Milios, E. (1997). Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4(4), 333–349. doi:10.1023/A:1008854305733
- Matheron, G. (1963). Principles of geostatistics. *Economic Geology and the Bulletin of the Society of Economic Geologists*, 58(8), 1246. doi:10.2113/gsecongeo.58.8.1246
- Microsoft. (2011). *Xbox Kinect web site*. Retrieved July 19, 2011, from <http://www.xbox.com/en-US/kinect>
- Mikhail, E., Bethel, J., & McGlone, J. C. (2001). *Introduction to modern photogrammetry*. New York, NY: Wiley.
- Mikolajczyk, K., & Schmid, C. A. (2005). Performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10), 1615–1630. doi:10.1109/TPAMI.2005.188
- Moravec, H. P., & Elfes, A. (1985). High resolution maps from wide angle sonar. In *Proceedings of the IEEE International Conference on Robotics and Automation*. IEEE Press.
- MRPT. (2011). *The mobile robot programming toolkit website*. Retrieved Mar 1, 2012, from <http://www.mrpt.org/>
- Nixon, M., & Aguado, A. S. (2008). *Feature extraction and image processing*. New York, NY: Academic Press.
- Nüchter, A. (2009). *3D robotic mapping: The simultaneous localization and mapping problem with six degrees of freedom*. Berlin, Germany: Springer Verlag.
- Núñez, P., Vázquez-Martín, R., del Toro, J. C., Bandera, A., & Sandoval, F. (2006). Feature extraction from laser scan data based on curvature estimation for mobile robotics. In *Proceedings of the IEEE International Conference on Robotics and Automation*, (pp. 1167-1172). IEEE Press.
- Piaget, J. (1948). *The child's conception of space*. London, UK: Routledge.

- Ranganathan, A., & Dellaert, F. (2011). Online probabilistic topological mapping. *The International Journal of Robotics Research*, 30(6), 755–771. doi:10.1177/0278364910393287
- Rasmussen, C. E., & Williams, C. K. I. (2006). *Gaussian processes for machine learning*. Cambridge, MA: The MIT Press.
- Remolina, E., Fernández-Madrigal, J. A., Kuipers, B. J., & González-Jiménez, J. (1999). Formalizing regions in the spatial semantic hierarchy: An AH-graphs implementation approach. *Lecture Notes in Computer Science*, 1661, 109–124. doi:10.1007/3-540-48384-5_8
- Rotwat, P. F. (1979). *Representing the spatial experience and solving spatial problems in a simulated robot environment*. (Doctoral Dissertation). University of British Columbia. British Columbia, Canada.
- Rusu, R. B., Marton, Z. C., Blodow, N., Dolha, M., & Beetz, M. (2008). Towards 3D point cloud based object maps for household environments. *Robotics and Autonomous Systems*, 56(11), 927–941. doi:10.1016/j.robot.2008.08.005
- Smith, M., Baldwin, I., Churchill, W., Paul, R., & Newman, P. (2009). The new college vision and laser data set. *The International Journal of Robotics Research*, 28(5), 595–599. doi:10.1177/0278364909103911
- Stachniss, C., Plagemann, C., Lilienthal, A., & Burgard, W. (2008). Gas distribution modeling using sparse Gaussian process mixture models. In *Proceedings of Robotics: Science and Systems*. IEEE. doi:10.1007/s10514-009-9111-5
- Strasdat, H., Montiel, J. M. M., & Davison, A. J. (2010). Scale-drift aware large scale monocular SLAM. In *Proceedings of Robotics Science and Systems*. IEEE.
- Tardós, J. D., Neira, J., Newman, P., & Leonard, J. (2002). Robust mapping and localization in indoor environments using sonar data. *The International Journal of Robotics Research*, 21(4), 311–330. doi:10.1177/027836402320556340
- Thrun, S., Burgard, W., & Fox, D. (2005). *Probabilistic robotics*. Cambridge, MA: The MIT Press.
- Tipaldi, G. D., & Arras, K. O. (2010). FLIRT – Interest regions for 2D range data. In *Proceedings of the IEEE International Conference on Robotics and Automation*. IEEE Press.
- Tolman, E. C. (1948). Cognitive maps in rats and men. *Psychological Review*, 55(4), 189–208. doi:10.1037/h0061626
- Trucco, E., & Verri, A. (1998). *Introductory techniques for 3-D computer vision*. Upper Saddle River, NJ: Prentice Hall.
- Trucco, E., & Verri, A. (1998). *Introductory techniques for 3-D computer vision*. Upper Saddle River, NJ: Prentice Hall.
- Trudeau, R. J. (1994). *Introduction to graph theory*. New York, NY: Dover Publications.
- Winkler, G. (1995). *Image analysis, random fields and dynamic Monte Carlo methods*. Berlin, Germany: Springer.
- Wurm, K. M., Hornung, A., Bennewitz, M., Stachniss, C., & Burgard, W. (2010). OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic systems. In *Proceedings of the ICRA Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation*. ICRA.

ENDNOTES

- ¹ Notice that denoting the “inverse observation function” as $\mathbf{h}^{-1}(\cdot)$ may be seen as an abuse of notation. Strictly speaking, if the observation function (disregarding the additive noise) is represented as the function

$\mathbf{z}_k = \mathbf{h}_i(\mathbf{x}_k, \mathbf{m}_i)$, its mathematical inverse function should be $(\mathbf{x}_k, \mathbf{m}_i) = \mathbf{h}_i^{-1}(\mathbf{z}_k)$ instead of the commonly employed $\mathbf{m}_i = \mathbf{h}_i^{-1}(\mathbf{x}_k, \mathbf{z}_k)$, which is the function of our interest for map building and SLAM.

Promotional Copy
Not for Redistribution

About the Authors

Juan-Antonio Fernández-Madrigo holds a PhD in Computer Science and is tenured Associate Professor in the University of Málaga (Spain). He has been teaching since 1998 in graduate and post-graduate courses on real-time systems, control engineering, and robotics. He has supervised PhD theses on cognitive robotics and probabilistic localization and mapping for mobile robots, and a relevant number of BSc and MSc theses on very diverse subjects. His research work has been developed mainly on different aspects of the modeling of the environment for mobile robots, cognitive robotics, and robotic software development. He has three books published internationally and nearly 80 scientific papers on these and other topics. He has been involved in different roles in regional, national, and European research projects, and is co-inventor of several patents. Regarding his personal interests, they currently include programming, drawing, and writing, having authored five sci-fi books in Spanish and more than a hundred short stories.

José Luis Blanco Claraco is a Lecturer in the University of Málaga (Spain), where he has taught in graduate courses on mechatronics and material science. He obtained a PhD in Mobile Robotics from the same university in 2009. His research interests include estimation theory, mobile robot navigation, large-scale map building, and computer vision. Since 2005, he has participated in several national and European research projects, as well as research collaborations with private companies. As a result, he has published more than 40 scientific papers and is co-inventor of three patents and one utility model. He is also an active supporter of Open Source initiatives, having participated in more than 10 software projects, most remarkably in the Mobile Robot Programming Toolkit (MRPT), which he started in 2005 and still actively maintains at present. Among his non-academic interests, he collaborates in popular science blogs aimed at the Spanish speaking community.