

Topological SLAM - Simultaneous Localization and Mapping with Fingerprints of Places

THÈSE N° 3357 (2005)

PRÉSENTÉE A LA FACULTE SCIENCES ET TECHNIQUES DE L'INGÉNIEUR

Institut d'ingénierie des systèmes

SECTION DE MICROTECHNIQUE

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Adriana TAPUS

Master en Informatique : Systèmes et Communications de l'Université Joseph Fourier Grenoble, France

Ingénieur Diplômé en Informatique, Département Informatique et Sciences de l'Ingénieur, Université Polytechnique de Bucarest,

Roumanie

et de nationalité roumaine

acceptée sur proposition du jury :

Prof. Roland Siegwart, directeur de thèse
Prof. Pierre Bessière, rapporteur
Prof. Raja Chatila, rapporteur
Prof. Dario Floreano, rapporteur

Lausanne, EPFL
2005

Table of Contents

<i>List of Tables</i>	5
<i>List of Figures</i>	7
<i>Acknowledgments</i>	11
<i>Abstract</i>	13
<i>Résumé</i>	15
1 Introduction	19
1.1 Space Representation	20
1.2 Problem Statement and Contributions	22
1.3 The Platforms	23
1.4 Thesis Outline	25
2 Multi-Modal Perception	27
2.1 Introduction	27
2.2 Odometry	27
2.3 Laser Range Finder	30
2.4 Omnidirectional Camera	32
2.5 Summary	35
3 Feature Extraction and Environment Modeling	37
3.1 Introduction	37
3.2 Laser Scanner Features	38
3.2.1 Horizontal Lines.....	38
3.2.1.1 Line Model.....	39
3.2.1.2 Segmentation with Douglas-Peucker Algorithm.....	39
3.3 Omnidirectional Camera Features	41
3.3.1 Vertical Edges.....	42
3.3.1.1 Edge Enhancement.....	42
3.3.1.2 Image Histogram and Thresholding.....	45

3.3.1.3 Non-Maxima Suppression and Regrouping Filtering	45
3.3.2 Color Patches	46
3.3.2.1 Color Spaces	47
3.3.2.2 Saturation Thresholding	47
3.3.2.3 Hue Histogram	48
3.3.2.4 Color Fusion	49
3.4 Environment Modeling with Fingerprint of Places	50
3.4.1 Fingerprint of Place Definition	50
3.4.2 Fingerprint Encoding	51
3.4.3 Fingerprint Generation	51
3.4.4 Uncertainty Modeling in the Fingerprint	51
3.5 Summary	54
<i>4 Topological Localization with Fingerprints of Places</i>	<i>55</i>
4.1 Related Work	55
4.2 Environment Model	57
4.3 Localization with Fingerprints of Places	57
4.3.1 Fingerprint Matching using Bayesian Programming	58
4.3.1.1 Bayesian Programming (BP) Formalism	58
4.3.1.2 Bayesian Program for Localization	60
4.3.1.3 Mixture of Gaussians	62
4.3.1.4 Expectation Maximization	63
4.3.1.5 Example	64
4.3.1.6 Experimental Results	65
4.3.1.6.1 Simulation Experiments	65
4.3.1.6.2 Experiments with the robots	68
4.3.2 Fingerprint Matching with Global Alignment Algorithm	69
4.3.2.1 Localization with Global Alignment (GA)	70
4.3.2.2 Localization with Global Alignment with Uncertainty	72
4.3.2.3 Indoor Experimental Results	73
4.3.2.3.1 Results with Global Alignment	73
4.3.2.3.2 Results with Global Alignment with Uncertainty	73
4.3.2.4 Outdoor Experimental Results	74
4.3.3 Fusion of Bayesian Programming and Global Alignment	76
4.3.3.1 Bayesian Program	76
4.3.3.2 Experimental Results	78
4.4 Discussion and Comparison of the Place Recognition Experiments	78
4.5 Summary	79
<i>5 Indoor Exploration Tools</i>	<i>81</i>
5.1 Introduction	81
5.2 Wall Following	81
5.2.1 Wall Detection	82
5.2.2 Control Law	83
5.2.3 Kalman Filter (KF)	84
5.3 Mid-line Following	87

5.4	Center of Free Space	87
5.4.1	Method Description	88
5.4.2	Experimental Results	88
5.5	Indoor Structures and Door Detection using Bayesian Programming.....	89
5.5.1	Implementation Related Assumptions	90
5.5.2	Indoor Structures and Doors Corpus.....	90
5.5.3	Bayesian Program for Indoor Structures and Doors Detection	90
5.5.3.1	Bayesian Program	92
5.5.3.2	Discussions	94
5.5.4	Experimental Results	95
5.6	Summary	98
6	<i>Simultaneous Localization and Mapping (SLAM) with Fingerprints of Places.....</i>	99
6.1	Related Work	99
6.2	Environment Representation with Fingerprints of Places	102
6.3	Topological Localization and Map Building.....	102
6.3.1	Topological Mapping Technique	102
6.3.1.1	Exploration Strategy	102
6.3.1.2	Automatic Map Building	103
6.3.2	Mean Fingerprint	104
6.3.2.1	Clustering of Fingerprints	105
6.3.3	Indoor Topological Localization with POMDP.....	106
6.3.4	Control Strategy	108
6.3.5	Map Update.....	109
6.3.6	Closing the Loop.....	109
6.4	Experimental Results	110
6.4.1	Mapping Indoor Environments	111
6.4.1.1	Results.....	111
6.4.2	Mapping Outdoor Environments	114
6.4.2.1	Results.....	114
6.4.3	Indoor Topological Localization with POMDP.....	115
6.4.4	Closing the Loop.....	116
6.5	Discussion and Limitations.....	117
6.6	Summary	118
7	<i>Conclusions.....</i>	119
7.1	Review and Contributions	119
7.2	The Hippocampal Place Cells and the Fingerprints of Places: Spatial Representation Animals, Animats and Robots.....	121
7.3	Open Issues.....	122
Appendix A	<i>Bayesian Programming – Basic Concepts.....</i>	125
A.1	Fundamental Definitions.....	125
A.1.1	Proposition	125
A.1.2	Variable.....	126
A.1.3	The probability of a proposition	126

A.2 Inference rules and postulates	127
A.2.1 The conjunction postulate (Bayes theorem)	127
A.2.2 The normalization postulate.....	127
A.2.3 The disjunction rule	127
A.2.4 The marginalization rule.....	127
A.3 Bayesian Programming Formalism	128
A.3.1 Structure of a Bayesian Program	128
A.3.2 Description.....	129
A.3.3 Specification	129
A.3.4 Identification.....	130
A.3.5 Utilization	130
A.4 Simple Example: Sensor Fusion	131
A.4.1 Bayesian Program Description	132
A.4.2 Results and Discussion	136
A.5 Summary	137
<i>Appendix B Glossary</i>	<i>139</i>
<i>Bibliography</i>	<i>143</i>
<i>Personal Publications</i>	<i>157</i>
<i>Curriculum Vitae</i>	<i>157</i>

List of Tables

<i>TABLE 4.1: Simulation results for VE (vertical edges) classification</i>	<i>66</i>
<i>TABLE 4.2: Mean number of occurrences of features.....</i>	<i>69</i>
<i>TABLE 4.3: Experimental results for localization with Bayesian Programming.....</i>	<i>69</i>
<i>TABLE 4.4: Comaparison of the results obtained with three dynamic programming methods for fingerprint-matching.....</i>	<i>73</i>
<i>TABLE 4.5: Experimental results for outdoor localization with GA</i>	<i>75</i>
<i>TABLE 4.6: Localization results after fusion of Bayesian Programming and GA ...</i>	<i>78</i>
<i>TABLE 4.7: Comparison of the results obtained with the fingerprint-matching algorithms.....</i>	<i>79</i>
<i>TABLE 5.1: Results for indoor structures recognition</i>	<i>96</i>
<i>TABLE 5.2: Results for doors detection</i>	<i>96</i>
<i>TABLE 5.3: Results for global recognition of indoor structures and doors</i>	<i>98</i>
<i>TABLE 6.1: Results for indoor topological localization with POMDP.....</i>	<i>116</i>

List of Figures

<i>Figure 1.1: Hierarchy of levels of abstraction.....</i>	<i>21</i>
<i>Figure 1.2: Mobile platforms.....</i>	<i>23</i>
<i>Figure 2.1: The differential drive system of the BIBA robot</i>	<i>28</i>
<i>Figure 2.2: Growing position uncertainty with odometry.....</i>	<i>29</i>
<i>Figure 2.3: Laser range finder LMS200 by SICK AG</i>	<i>31</i>
<i>Figure 2.4: Example of a scan</i>	<i>32</i>
<i>Figure 2.5: Omnidirectional camera system mounted on our mobile platforms</i>	<i>33</i>
<i>Figure 2.6: Equiangular mirror.....</i>	<i>34</i>
<i>Figure 2.7: Example of panoramic images (raw and unwarped).....</i>	<i>35</i>
<i>Figure 3.1: Scan with the extracted lines with the Douglas-Peucker(DP) algo.....</i>	<i>38</i>
<i>Figure 3.2: The line model</i>	<i>39</i>
<i>Figure 3.3: Douglas-Peucker (DP) algorithm.....</i>	<i>41</i>
<i>Figure 3.4: Flow-chart for vertical edge extraction</i>	<i>43</i>
<i>Figure 3.5: Convolution mask principle</i>	<i>43</i>
<i>Figure 3.6: Sobel mask.....</i>	<i>44</i>
<i>Figure 3.7: Example of a gradient magnitude image.....</i>	<i>44</i>
<i>Figure 3.8: Histograms of the gradient magnitude in the vertical direction.....</i>	<i>45</i>
<i>Figure 3.9: Example of image with the extracted vertical edges</i>	<i>46</i>
<i>Figure 3.10: Example of image with the extracted vertical edges after filtering.....</i>	<i>46</i>
<i>Figure 3.11: Predetermined color set.....</i>	<i>47</i>
<i>Figure 3.12: Example of a saturation image after the thersholding process.....</i>	<i>48</i>
<i>Figure 3.13: Fuzzy voting process.....</i>	<i>48</i>
<i>Figure 3.14: Color histogram.....</i>	<i>49</i>
<i>Figure 3.15: Color fusion</i>	<i>50</i>
<i>Figure 3.16: Fingerprint concept overview</i>	<i>50</i>
<i>Figure 3.17: Fingerprint generation.....</i>	<i>52</i>

<i>Figure 4.1: General procedure for fingerprint based localization.....</i>	<i>57</i>
<i>Figure 4.2: Structure of a Bayesian Program</i>	<i>59</i>
<i>Figure 4.3: Bayesian Programming and other probabilistic approaches</i>	<i>60</i>
<i>Figure 4.4: Bayesian Program for robot localization with fingerprints.....</i>	<i>61</i>
<i>Figure 4.5: Evaluation of $P(f=VE loc \pi)$ for the original set.....</i>	<i>64</i>
<i>Figure 4.6: Evaluation of $P(f=VE loc \pi)$ for other data set</i>	<i>65</i>
<i>Figure 4.7: Comparison of simulation results for different MOG components.....</i>	<i>67</i>
<i>Figure 4.8: Test environment.....</i>	<i>68</i>
<i>Figure 4.9: Example of calculating the cost between two strings with GA.....</i>	<i>70</i>
<i>Figure 4.10: The main elements of the Global Alignment algorithm.</i>	<i>71</i>
<i>Figure 4.11: Example of a GA with uncertainty</i>	<i>72</i>
<i>Figure 4.12: The outdoor test environment (a part of the EPFL campus) with the trajectory of 1.65 km long traveled by the Smart vehicle.</i>	<i>74</i>
<i>Figure 4.13: Fingerprint matching – orientation independent</i>	<i>75</i>
<i>Figure 4.14: The fusion of Bayesian Programming and GA algorithm</i>	<i>77</i>
<i>Figure 5.1: Wall following diagram.....</i>	<i>82</i>
<i>Figure 5.2: Control law for wall-following behavior</i>	<i>83</i>
<i>Figure 5.3: Wall following technique description.....</i>	<i>86</i>
<i>Figure 5.4: Center of free space.....</i>	<i>88</i>
<i>Figure 5.5: Different views of the free space taken by the panoramic vision system, with their correspondent data given by the laser scanner.....</i>	<i>88</i>
<i>Figure 5.6: Indoor structures and Doors to learn.</i>	<i>91</i>
<i>Figure 5.7: The BP used for the Indoor structures and doors identification.....</i>	<i>92</i>
<i>Figure 5.8: The observation field of view of the robot.....</i>	<i>95</i>
<i>Figure 5.9: Example of a false positive door detection</i>	<i>97</i>
<i>Figure 6.1: Flow-chart of the new topological node detection algorithm.....</i>	<i>103</i>
<i>Figure 6.2: Adding a new node automatically to the topological map by moving in an unexplored environment.....</i>	<i>104</i>
<i>Figure 6.3: Mean fingerprint generation.....</i>	<i>105</i>
<i>Figure 6.4: Loop Closing Problem.....</i>	<i>110</i>
<i>Figure 6.5: The topological map of the first indoor test environment.....</i>	<i>112</i>
<i>Figure 6.6: The topological map of the second indoor test environment.</i>	<i>113</i>
<i>Figure 6.7: The outdoor test environment used for mapping</i>	<i>114</i>
<i>Figure 6.8: The low granularity outdoor topological map.....</i>	<i>115</i>

<i>Figure 6.9: Example of loop closing.....</i>	<i>117</i>
<i>Figure A.1: Structure of a Bayesian Program</i>	<i>128</i>
<i>Figure A.2: Example of a simple Bayesian Program: light sensors fusion</i>	<i>132</i>
<i>Figure A.3: The mean value of the light sensor in function of the distance D and the bearing θ_L of the light source</i>	<i>134</i>
<i>Figure A.4: Results of the sensor fusion.....</i>	<i>137</i>

Acknowledgments

This thesis is the result of my work as a research assistant and doctoral student at the Autonomous Systems Lab of Prof. Roland Siegwart at Ecole Polytechnique Fédérale de Lausanne (EPFL).

First of all, I would like to thank Prof. Roland Siegwart for being a great advisor, for believing in my ideas, supporting and motivating me all along. I am also grateful to Prof. Roland Siegwart for the many opportunities I was given to go present my work in conferences, meet other people, and see their work.

I would also like to thank Prof. Pierre Bessière, Prof. Raja Chatila and Prof. Dario Floreano for being my co-examiners.

I would like to thank Prof. Pierre Bessière not only for being one of the co-examiners of this work, but also for his valuable comments and help on Bayesian stuff, and for the great summer and winter BIBA (Bayesian Inspired Brain and Artifacts) schools.

Many thanks to Prof. Alain Berthoz and Francesco Battaglia from LPPA Collège de France, for helping me understanding more things about neurophysiology.

I wish to express my gratitude to several former students: Etienne Glauser, Stefan Heinzer, now Ph.D. student in the Institute of Biomedical Engineering, ETH Zürich, and Luc Dobler for their valuable collaboration on the navigation software.

Special thanks to Marie-José Pellaud for being a second woman in the lab and for solving all the administrative problems, Nicola Tomatis for putting me on the track in the area of mobile robots navigation, Agostino Martinelli for being Masssicio, Guy Ramel for the different collaborations and for all his software support when needed, Sascha Kolski for helping me with the outdoor experiments (driving the SMART vehicle is not an easy task), Frederic Pont for the GenoM discussions and Shrihari Vasudevan for being a valuable friend, for providing anytime help and for the nice cognitive mapping discussions. Thanks also to Samir Bouabdallah for creating a very nice atmosphere in our office. Distinguish thanks to all the ASL team. They provided a very funny, friendly, stimulating and supportive environment.

I would also like to thank my mom and dad for believing in me, for providing support, love and understanding. A great thank to my brother Cristian for all his help, nice

discussions and debates. I also thank my brother and his wife Diana for making me the nicest gift ever, my nephew Alexandru Ioan.

Last but not the least I would like to thank Yann for all his support, help and love during these last three years. He made each moment easier, more fun, more rememberable, and more unique. He and his family were great. Thank you!!! This dissertation is entirely dedicated to my family and my husband, that I love more than anything in the world.

Abstract

This thesis is about topological navigation, more precisely about space representation, perception, localization and mapping. All these elements are needed in order to obtain a robust and reliable framework for navigation. This is essential in order to move in an environment, manipulate objects in it and avoid collisions. The method proposed in this dissertation is suitable for fully autonomous mobile robots, operating in structured indoor and outdoor environments.

High robustness is necessary to obtain a distinctive and reliable representation of the environment. This can be obtained by combining the information acquired by several sensors with complementary characteristics. A multimodal perception system that includes the wheel encoders for odometry and two exteroceptive sensors composed of a laser range finder that gives a 360° view of the environment and an omnidirectional camera are used in this work. Significant, robust and stable features are extracted from sensory data. The different features extracted from the exteroceptive sensors (i.e. corners, vertical edges, color patches, etc.) are fused and combined into a single, circular, and distinctive space representation named the *fingerprint of a place*. This representation is adapted for topological navigation and is the foundation for the whole dissertation.

One of the major tasks of a mobile robot is localization. Different topological localization approaches based on the *fingerprint* concept are presented in this dissertation. Localization on a fingerprint-based representation is reduced to a problem of *fingerprint* matching. Two of these methods make use of the Bayesian Programming (BP) formalism and two others are based on dynamic programming. They also show how multimodal perception increases the reliability of topological localization for mobile robots.

In order to autonomously acquire and create maps, robots have to explore their environment. Several exploration tools for indoor environments are presented: wall following, mid-line following, center of free space of a room, door detection, and environment structure identification.

An automatic and incremental topological mapping system based on *fingerprints of places* and a global localizer using *Partially Observable Markov Decision Processes* (POMDP) are proposed. The construction of a topological mapping system is combined with localization, both relying on *fingerprints of places*, in order to perform *Simultaneous Localization and Mapping* (SLAM). This enables navigation of an autonomous mobile robot in a structured environment without relying on maps given a

priori, without using artificial landmarks and by employing a semantic spatial representation that allows a more natural interface between humans and robots. The fingerprint approach, combining the information from all sensors available to the robot, reduces perceptual aliasing and improves the distinctiveness of places. This fingerprint-based approach yields a consistent and distinctive representation of the environment and is extensible in that it permits spatial cognition beyond just pure navigation.

All these methodologies have been validated through experiments. Indoor and outdoor experiments have been conducted over a distance exceeding 2 km. The *fingerprints of places* proved to provide a compact and distinctive methodology for space representation and place recognition – they permit encoding of a huge amount of place-related information in a single circular sequence of features. The experiments have verified the efficacy and reliability of this approach.

Résumé

Cette thèse porte sur la navigation topologique, et plus précisément sur la représentation spatiale, la localisation et la construction de cartes. Tous ces éléments sont nécessaires afin d'obtenir un cadre de travail robuste et fiable pour la navigation. Ceci est essentiel pour se déplacer dans un environnement, manipuler des objets dans celui-ci et éviter des collisions. La méthode proposée dans cette dissertation convient aux robots mobiles autonomes, opérants dans des environnements intérieurs et extérieurs structurés.

Afin d'obtenir une représentation distincte et fidèle de l'environnement, une robustesse importante est nécessaire. Il est possible de l'obtenir en combinant les informations acquises par des différents capteurs avec des caractéristiques complémentaires. Un système de perception multimodale a été utilisé dans ce travail. Il inclut des encodeurs de roues pour l'odométrie et deux capteurs extéroceptifs composés d'un laser scanner capable de donner une vue à 360 degrés de l'environnement, et une caméra omnidirectionnelle. Des *features* robustes, précises et significatives sont extraites des données sensorielles. Les différentes *features* extraites des capteurs extéroceptifs (par exemple : les coins, les lignes verticales, les couleurs, etc.) sont fusionnées et combinées dans une représentation spatiale circulaire, unique et distincte appelée « signature d'un lieu » (*fingerprint of a place*). Cette représentation est adaptée pour la navigation topologique et il s'agit du fondement même de cette entière dissertation.

Une des tâches principales pour un robot mobile est la localisation. Différentes approches de localisation topologique basées sur le concept des *fingerprints* sont présentées tout au long de ce travail. La localisation basée sur la représentation à l'aide de *fingerprints* se réduit à un problème de correspondances entre *fingerprints*. Deux de ces méthodes utilisent un formalisme de Programmation Bayésienne (BP) et les deux autres sont basées sur la programmation dynamique. Elles décrivent également comment la perception multimodale augmente la fidélité de la localisation topologique pour des robots mobiles.

Afin d'obtenir et de créer de façon autonome des cartes, les robots doivent explorer leur environnement. Plusieurs outils d'exploration pour les environnements intérieurs sont présentés : le suivi de mur (wall following), le suivi de ligne du milieu (mid-line following), le centre d'une espace libre d'une pièce (center of free space), la détection de portes, et l'identification de structure de l'environnement.

Un système de construction de cartes topologiques automatique et incrémental basé sur les *fingerprints of places*, ainsi qu'une localisation globale utilisant les Processus

Décisionnels Partiellement Observables de Markov (POMDP) sont proposés. Le développement d'un système de construction de cartes topologiques est combiné avec la technique de localisation globale, toutes deux s'appuyant sur les *fingerprints of places*, afin de donner une solution au problème nommé Simultaneous Localization and Mapping (SLAM). Ce qui permet la navigation d'un robot mobile autonome dans un environnement structuré sans s'appuyer sur les cartes apprises auparavant et sans utiliser des points de repère artificiels. L'approche à base du concept de fingerprints, en combinant les informations provenant de tous les capteurs disponibles du robot, réduit le perceptual aliasing et améliore la distinction des lieux. Elle donne également une représentation consistante et distincte de l'environnement, et est extensible dans la mesure qu'elle permet la cognition spatiale au delà de la pure navigation.

Toutes ces méthodologies ont été validées à travers des expérimentations. Des expériences ont été conduites à l'intérieur et à l'extérieur sur une distance supérieure à 2 km. Les *fingerprints of places* permettent de produire une méthode compacte et de distinction pour la représentation spatiale et la reconnaissance de lieux – ils permettent l'encodage d'une énorme quantité d'informations propres aux lieux dans une simple et unique séquence circulaire de caractéristiques. Les expériences ont permis de vérifier l'efficacité et la validité de cette approche.

1

Introduction

“The important thing is not to stop questioning.”

Albert Einstein (1879-1955)

If asked to describe what mental picture the word "robot" evokes, most people would rather describe something from science fiction (e.g., Sonny of I, Robot; C3PO of Star Wars) than scientific reality. In the past, most robots were developed for use in relatively stable and structured environments (e.g., manufacturing, welding, spraying); however, in the past 20 years, robotics researchers have been continually advancing the state of robotics - moving our images of robots from science fiction closer to reality. One day, robots will have the same level of impact on our lives, as personal computers have today.

One of the most fundamental tasks that a robot should be able to accomplish is navigation in the environment. This implies the need for mobility and interaction with the environment (e.g. the movement within an environment, the manipulation of objects in it, and the avoidance of collisions). This thesis addresses the above mentioned broad objective, and focuses on space representation, perception, localization and mapping.

The introductory part contains a description of existing work related to the navigation framework, and focuses on the problems addressed in this work.

1.1 Space Representation

In all our daily activities, the natural surroundings that we inhabit play a crucial role. Many neurophysiologists have dedicated their efforts towards understanding how our brain can create internal representations of the physical space. Both neurobiologists and roboticists are interested in understanding the behavior of intelligent beings like us and their capacity to learn and use their knowledge of the spatial representation in order to navigate. The ability of intelligent beings to localize themselves and to find their way back home is linked to their internal “mapping system”. Most navigation approaches require learning and consequently need to memorize information. Stored information can be organized into *cognitive maps* – term introduced for the first time in [Tolman48]. Tolman’s model advocates that the animals (rats) don’t learn space as a sequence of movements; instead the animal’s spatial capabilities rest on the construction of maps, which represent the spatial relationships between features in the environment.

Various methods have been proposed to represent environments in the framework of autonomous navigation, from precise geometric maps based on raw data or lines to purely topological maps using symbolic descriptions. Each of these methods is optimal with respect to some characteristics but can be very disappointing with respect to others.

Most current approaches make a trade-off between precision and global distinctiveness. Precision and distinctiveness have a strong link with the level of abstraction of the features used for navigation (see Figure 1.1).

Raw data represents the lowest level in the hierarchy of abstraction. Localization and mapping with raw data can result in very high precision of the represented environment, but the required data volume scales very badly with the size of the environment and the distinctiveness of the individual data points is very low. An example of such an approach is Markov localization [Fox98], [Thrun01].

The second level of abstraction corresponds to geometric features (e.g. lines, edges). The stochastic map technique to SLAM [Castellanos99], [Dissanayake01], [Leonard92] and the multi-hypothesis localization [Arras03] are typical examples belonging to this level. These approaches still feature high precision with reduced memory requirements, but have shortcomings concerning global distinctiveness and non modeled events.

Partially geometric features correspond to the third level of the hierarchy. Significant progress has been made since the seminal papers by Kuipers [Kuipers78] and [Kuipers91], where an approach based on concepts derived from a theory on human cognitive mapping is described as the body of knowledge representing large scale space. Representations using partially geometric features are demonstrated using *fingerprints of places* (i.e. a circular sequence of low level features) described in this thesis, and the more bio-inspired approaches shown in [Arleo00], [Berthoz97], and [Hafner00] using neural networks.

On the highest level of abstraction, the environment is represented by a purely symbolic description. This can be very compact and distinctive, but reliable tools for extraction of high level features are still missing.

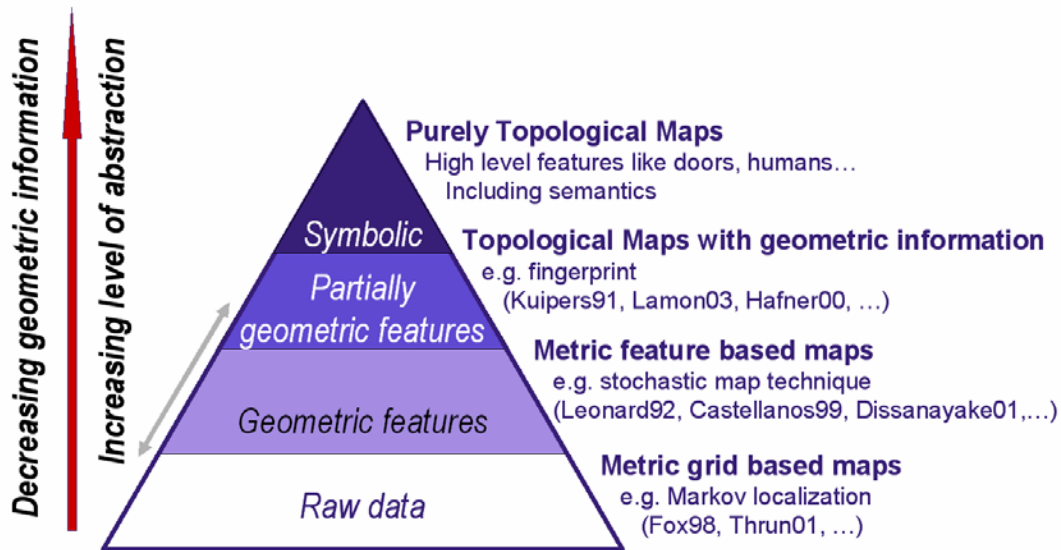


Figure 1.1: *Hierarchy of abstraction levels.*

These levels of abstraction are represented in a pyramidal form as depicted in Figure 1.1. It can be noticed that the more we go up the hierarchy, the more the geometric information is reduced and distinctiveness is increased. For global localization and mapping, high distinctiveness is of importance, whereas for local action, precise geometric relations with the environment are more critical.

Navigation strategies are based on two complementary sources of information (available on the mobile agent: animal, robot): idiothetic and allothetic. The idiothetic source yields internal information about the mobile agent's movements (e.g. speed, acceleration, etc.). Allothetic sources of information provide external information about the environment (e.g. the cues coming from the visual sensors, odor sensors, laser range finders, sonars, etc.). Idiothetic information provides a metric estimate of the agent's motion, but suffers from error accumulation, which makes the position estimation unreliable at long-term. In contrast, the allothetic (sensory) data is stationary over time, but is susceptible to perceptual aliasing (i.e. observations at multiple locations are similar) and requires non-trivial processing in order to extract spatial information.

The map-based navigation needs map-learning and localization. Map-learning is the process of constructing a map representing the environment explored by the mobile agent and localization is the phenomenon of finding the mobile agent's location (position) in the map. Localization and mapping are interdependent – to localize the robot, a map is necessary and to update a map the position of the mobile agent is needed. This is usually known as Simultaneous Localization and Mapping (SLAM). While navigating in the environment, the mobile agent first creates and then updates the map.

1.2 Problem Statement and Contributions

As previously mentioned, many methods for localization and mapping have been proposed. However, navigation approaches capable of coping with both structured indoor and outdoor environments and at the same time permitting a higher level, symbolic representation of space are very rare. The aim of this thesis is to contribute towards gaining a better understanding of these problems. The research work proposed in this dissertation is based on a multi-modal, feature-based representation of the environment, called *a fingerprint of a place*. It also proposes concrete solutions for improving the distinctiveness of space representation, localizing the robot and building compact and reliable topological maps.

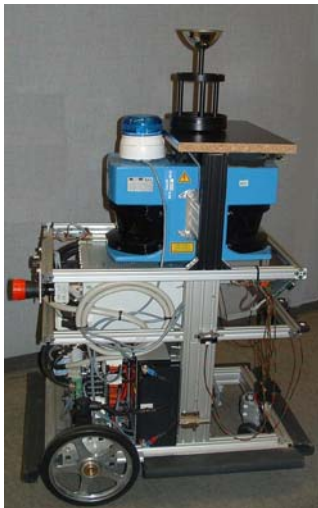
This work contributes to following fields:

- *Space Representation*: navigating in unknown environments requires the development of an adapted structure capable of representing space in a compact fashion, minimizing the ambiguities due to perceptual aliasing and increasing the distinctiveness of space. The *fingerprint of a place* is proposed as an efficient and distinctive methodology for representing space. This characterization of the environment is especially interesting when used within a topological and multiple modality framework.
- *Localization*: localization is one of the fundamental problems in mobile robotics. Different methods for fingerprint based localization have been developed, implemented and tested both for indoor and outdoor environments. The localization problem is reduced to one of matching sequences of local features representing the environment.
- *Exploration*: a mobile robot has to explore its environment in order to construct maps. Probabilistic tools are proposed for indoor exploration. A new method for doors and indoor structures recognition based on the Bayesian Programming formalism is also proposed in this work.
- *Mapping*: a robust navigation system requires a spatial model of physical environments. In this work, a topological representation of space has been chosen. The topological map can be viewed as a graph of places, where at each node, the information concerning the visible landmarks (i.e. the *fingerprints of places*) and the way to reach other places connected to it, is stored. The fingerprints of places permit the construction of maps that are compatible with the topology of the environment. The fingerprint-based approach for mapping, presented in this dissertation, yields a consistent and distinctive representation of the environment and is extensible in that it permits spatial cognition beyond just pure navigation.

All approaches introduced in this work have been validated on the two mobile platforms, the BIBA robot for indoor experiments and the "SMART" vehicle for outdoor experiments, respectively. These two platforms are described in more detail in the next section. The experiments have been conducted in several indoor and outdoor environments. The indoor environments used in this work are a part of our institute building, as depicted in Figure 4.8. They include scenarios found in typical office environments such as places that look a same (e.g. corridors) and people moving around. The outdoor environment is part of the EPFL campus.

1.3 The Platforms

For the work reported here, two mobile platforms have been used: the BIBA robot (see Figure 1.2(a)) for indoor experiments and the "SMART" vehicle (see Figure 1.2(b)) for outdoor experiments. The BIBA robot has been built for the BIBA (i.e. Bayesian Inspired Brain and Artefacts) European project and it was designed by BlueBotics SA, a spin-off of the Autonomous Systems Lab at the Ecole Polytechnique Fédérale de Lausanne (EPFL).



(a)



(b)

Figure 1.2: *Mobile Platforms: (a) Indoor Platform: The fully autonomous BIBA robot. (b) Outdoor platform: The "SMART" vehicle.*

The relevant characteristics of each platform are mentioned below.

A. BIBA Robot

- Computers:
 - PowerPC 750 400 MHz running XO/2 operating system
 - PC Intel Pentium III 700 MHz running Windows 2000
- Sensors:
 - Two SICK LMS200 laser range finders that give a panoramic view of the environment (except the two blind zones) (see Chapter 2)
 - Four ultrasound I2C modules
 - 3x5 Infrared sensors
 - Omnidirectional Vision System:
 - Camera: SONY DFW VL500
 - Mirror: "360 One VR" equiangular mirror
 - Motors with encoders and harmonic drives and EC motor amplifiers

- Bumpers – four tactile zones around the robot
- Mechanics:
 - Chassis
 - Two drive wheels in differential drive arrangement
 - Front castor wheel
- Communication:
 - Wireless Ethernet 802.11
- Autonomy:
 - Two batteries (12 V) that provide up to 2 hours of autonomy
- External Measurements:
 - Size: 42 cm width x 54 cm depth x 56 cm height
 - Weight: 30 kg

B. "SMART" vehicle equipped for autonomous driving

- Computers:
 - PC Intel Pentium IV 2.4 GHz running Unix
- Sensors:
 - Three SICK LMS200 laser range finders; two laser range finders mounted on top of the vehicle that give a panoramic view of the environment (except the two blind zones) (see Chapter 2) and one in front of the car for obstacle detection.
 - One Inertial Measurement Unit (IMU) delivering angular speed, and angular and translational accelerations of the car
 - CAN (Controller Area Network) bus delivering data of the actual steering angle and the actual speed of the vehicle
 - A stereo-vision system composed of two monocular cameras SONY DFW VL 500.
 - Omnidirectional Vision System:
 - Camera: SONY DFW VL500
 - Mirror: "360 One VR" equiangular mirror
- Mechanics:
 - Engine type: 3-cylinder in-line engine at rear with turbo charger, charger cooler
 - Capacity in cc: 698
- External Measurements:
 - Size: 2.5 m width x 1.515 m depth x 1.815 m height
 - Weight: 730 kg

1.4 Thesis Outline

The previously described sections briefly synopsise the motivations, problems and contributions of the thesis. This section outlines the structure of the thesis and summarizes each of the chapters.

Chapter 2 describes the sensors - laser range finder, odometry and omnidirectional camera - relevant to this work and shows the importance of having multi-modal perception. These sensors are useful for the main task of this work: localization and mapping.

Different methods for features extraction from range data and omnidirectional images are introduced and discussed in **Chapter 3**. These features are fused and combined into a single and distinctive space representation named the *fingerprint of a place*. This representation is adapted for topological navigation and is the foundation for the next chapters.

Chapter 4 presents several approaches for topological localization using *fingerprints of places*. These techniques for localization are tested both in indoor and outdoor environments and a comparison of all of them is reported.

In **Chapter 5**, exploration techniques are addressed. In order to have a complete navigation system, the robot has to have the ability to move through the environment autonomously. Hence, methods for wall and corridor following, door detection and indoor topological structures identification are presented. By combining the different behaviors, the robot easily navigates and applies exploration strategies

Chapter 6 presents a new automatic and incremental topological mapping system based on *fingerprints of places* and a global localizer using *Partially Observable Markov Decision Processes* (POMDP). The construction of a topological map is combined with localization, both relying on *fingerprints of places*, in order to perform *Simultaneous Localization and Mapping* (SLAM). The obtained maps are compact, consistent and distinctive. The method for topological SLAM has been validated in indoor and outdoor environments.

Finally, **Chapter 7** evokes the main points of this dissertation, raises some questions related to the presented work and points to further questions and work still to be done.

2

Multi-Modal Perception

2.1 Introduction

In order to be able to develop autonomous and intelligent robotic systems, sensors and data fusion techniques are needed. Unstructured environments involves unforeseen and unpredictable environment characteristics (including dynamic objects and human beings), thus making sensing very critical for mobile robots. These sensors are embedded in the robot system and they can be classified as proprioceptive and exteroceptive sensors. Proprioceptive sensors measure and monitor the internal state of a robot (e.g. motor speed, temperature, acceleration). In this work, only the wheel-encoders are used as proprioceptive sensors. The purpose of exteroceptive sensors is to measure environment features such as distance, color and luminosity. These sensors help the robot to detect the changes in the environment, to avoid obstacles and to correct the errors encountered in the world model. However, perfect sensors do not exist. Each sensor provides different kinds of information and has its positive and negative sides. In order to overcome the limitations of the individual sensors, in this work, a multi-sensory perception system is used. By adopting a multi-sensory approach, fusing sensory input from various sensors, a more accurate and robust perception system is realized. The resulting measurement (of the world state) is thus much more reliable. This chapter illustrates the multi-sensory perception system used in this thesis. It elaborates on the various sensors used and elicits how the internal / external states are measured. The sensors used in this work include wheel encoder, two laser range finders and an omnidirectional camera.

2.2 Odometry

As mentioned earlier, wheel-encoders are internal-state sensors and they measure the motion of a mobile robot. Odometry yields good short term accuracy, is inexpensive

and permits sampling rates that are very high. Identifying the odometry errors of a mobile robot is of high importance both in order to reduce them, and to

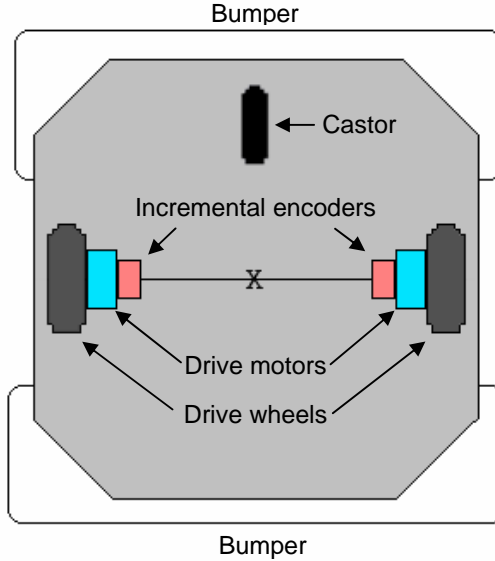


Figure 2.1: *The differential drive system of the BIBA robot with the two incremental encoders, drive wheels and motors.*

determine the accuracy of the state estimation based on encoder data. Odometry is characterized by two types of errors: systematic and non-systematic.

Systematic errors occur due to asymmetries and uncertainties in robot production and assembly (e.g. unequal wheel diameters produce curved trajectories; uncertainty about wheel base produces errors in turn angle). These errors can be identified and adjusted. Borenstein and Feng in [Borenstein94] propose a calibration technique called UMBmark test, developed to calibrate the systematic errors of a mobile robot equipped with a differential drive system.

Non-systematic errors are random errors. Situations in which these can appear include uneven and slippery floors. Non-systematic errors can basically be generated by wheel slippage and backlash, and non-continuous sampling of wheel increments [Chong97]. These errors can be modeled by describing the interaction between system's dynamic and the error sources.

In [Borenstein96], Borenstein et al. elicit the different possible sources of both these kinds of errors and a method for calibration and quantification of systematic and non-systematic errors.

The indoor robot used in this work has a differential drive kinematics (see Figure 2.1). In order to model the odometry error of such a system, the error in the displacement of each wheel (left and right) is calculated separately, and the uncertain control input is expressed as:

$$u(k+1) = [\delta\rho_R, \delta\rho_L]^T, \quad (2.1)$$

where $\delta\rho_R$ and $\delta\rho_L$ are the displacements of the right and left wheel respectively.

Chong and Kleeman in [Chong97] proposed a model for errors in the wheel space, starting from the uncertain input $u(k+1)$ and the diagonal input covariance matrix:

$$U(k+1) = \begin{bmatrix} k_R |\delta\rho_R| & 0 \\ 0 & k_L |\delta\rho_L| \end{bmatrix}, \quad (2.2)$$

where k_R and k_L are two constants (with unit meter) representing the non-systematic parameters of the motor drive and the wheel floor interaction.

Some assumptions were made when the covariance matrix shown in Equation (2.2) was calculated:

- the errors of the individually driven wheels are independent (the covariance matrix is diagonal)
- the variance of the errors is proportional to the absolute value of the traveled distances $\delta\rho_R$ and $\delta\rho_L$.

The main idea of odometry is the integration of incremental motion information over time, which leads inevitably to the accumulation of errors. The accumulation of orientation errors produce large position errors, which increase proportionally with the distance traveled by the robot. This process is depicted in Figure 2.2.

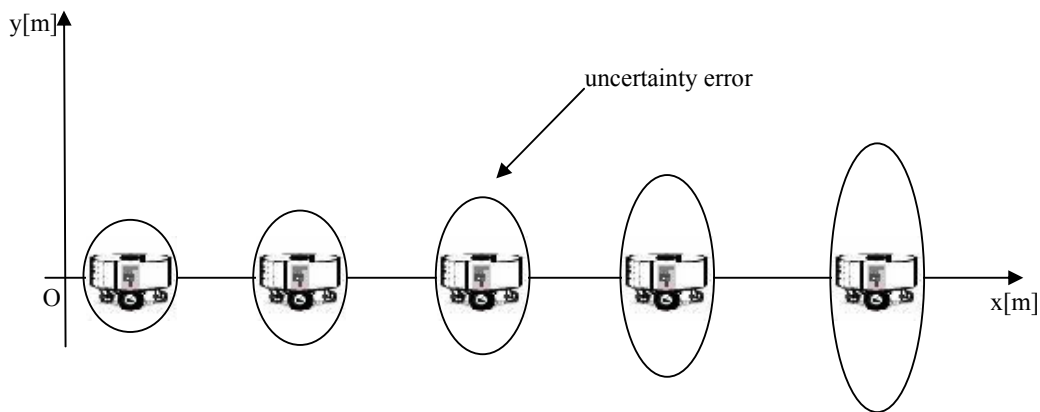


Figure 2.2: Growing position uncertainty with odometry for straight-line movement. The ellipses show the uncertainty error that grows with respect to the distance traveled. Note that the uncertainty in y grows much faster than in the direction of movement, due to the integration of the uncertainty about the robot's orientation.

Let $X = [x \ y \ \theta]^T$ be the state of the robot composed of the Cartesian coordinates of the robot (x, y) and θ its orientation with respect to a global reference frame. The kinematic model (i.e. the estimation of the position of the robot after performing the action) can be described as:

$$\begin{aligned}
x(k+1|k) &= f(x(k|k), u(k+1)) \\
&= x(k|k) + \begin{bmatrix} \delta\rho_k \cos(\theta_k + \frac{\delta\theta_k}{2}) \\ \delta\rho_k \sin(\theta_k + \frac{\delta\theta_k}{2}) \\ \delta\theta_k \end{bmatrix},
\end{aligned} \tag{2.3}$$

where $\delta\rho_k = \frac{\delta\rho_R + \delta\rho_L}{2}$, $\delta\theta_k = \frac{\delta\rho_R - \delta\rho_L}{d}$ and d is the distance between the left and the right wheel.

The associated uncertainty is given by the following expression:

$$P(k+1|k) = \nabla f_x P(k|k) \nabla f_x^T + \nabla f_u U(k+1) \nabla f_u^T \tag{2.4}$$

where ∇f_x and ∇f_u are the Jacobians taken with respect to the uncertain inputs $x(k+1|k)$ and $u(k+1)$.

A more detailed description is given in [Siegwart04]. A method for on-the-fly odometry error estimation is presented in [Martinelli03b].

2.3 Laser Range Finder

In this work, the laser range finder is used as one of the sources of information for feature extraction. The laser range finder is an active (i.e. an active sensor is a sensor that emit energy into the environment and measure the environmental reaction) time-of-flight sensor that scans its surroundings two-dimensionally. The time-of-flight principle measures the time taken by the emitted light to travel to an obstacle and return back. Thus, in order to compute the distance between objects and the measurement device, the elapsed time between emission and reception of the laser pulse is calculated.

The BIBA robot (see Figure 1.2(a)) and the "SMART" vehicle (see Figure 1.2(b)) are both equipped with two laser scanners of the type LMS200, by SICK AG. Each sensor covers 180° degrees. Mounted back to back, the two laser range finders construct an omnidirectional distance measurement system, except for a blind-zone on the left and on the right of the robot (see Figure 2.3).

The laser range finder LMS200 by SICK AG is an industrial device; it is widely used in mobile robotics. This device performs twenty five 180° degrees scans per second. Depending on its mode, the sensor has a measurement range of 8 to 80 meters. The angular resolution can be 0.25°/0.5°/ 1° degree (selectable).

The measurement range of the scanner depends on the reflectivity of the target object and the transmission strength of the scanner. The amount of light reflected

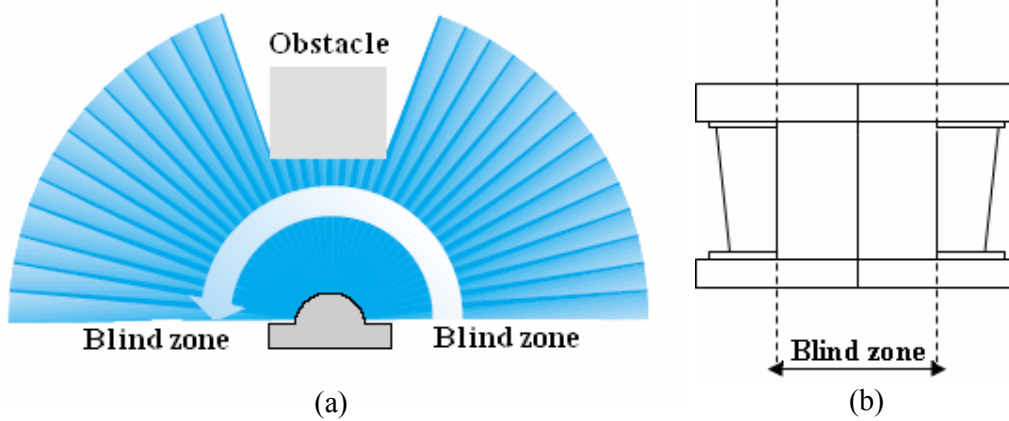


Figure 2.3: (a) The LMS200 by SICK AG with its scanning angle of 180° , its direction of transmission and its blind zones (image as depicted on the official website of SICK: <http://www.sick.com/saqqara/IM0012759.PDF>); (b) The two laser range finders mounted back to back, with their blind zone

depends on the nature of reflecting surface (composition, structure, density, color and so on), the texture of the reflecting surface (smooth or rough, regular or irregular, dull or polished, etc.), the wavelength and polarization of the light, the angle at which the light strikes the surface. While laser range finders are known for their accuracy, they also suffer from a severe limitation of being unable to detect / measure transparent objects/targets such as glass. This limitation may prove very impairing in certain scenarios such as the typical indoor environments found in museums.

Even if the laser scanner provides high accuracy, its measurements still have small errors on distance and angle. These measurement uncertainties can principally occur due to the timer counting the time-of-flight, the stability of the rotational frequency of the revolving unit, the surface quality and the frequency with which beams are sent out [Jensen04]. For the mode BIBA robot is running (8 meters and 0.5° of angular resolution) the documentation of the sensor [SICK00] specifies the standard deviation on the distance measured to 5 mm. The angular uncertainty is not specified. The angular values are constant and taken from a table, which makes the experimental determination of the angular uncertainty quite difficult.

The covariance matrix that models the sensor uncertainty is given as follows:

$$U_{\text{sensor}} = \begin{bmatrix} \sigma_{\text{distance,sensor}}^2 & 0 \\ 0 & \sigma_{\text{angle,sensor}}^2 \end{bmatrix}, \quad (2.5)$$

where $\sigma_{\text{distance,sensor}}^2$ is the uncertainty on the distance measured and $\sigma_{\text{angle,sensor}}^2$ is the angular uncertainty. The covariance matrix is diagonal, since the distance and angular uncertainty are assumed to be independent.

Figure 2.4 illustrates an example of a scan taken in a portion of our institute-building at EPFL with the two laser range finders LMS200 by SICK AG mounted back to back.

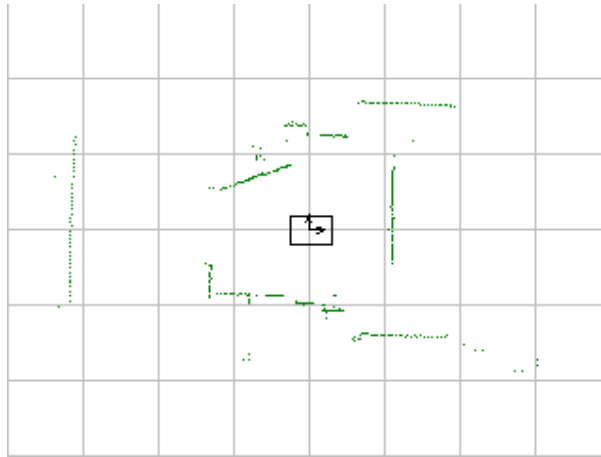


Figure 2.4: Example of a scan taken with two LMS200 by SICK AG mounted back to back. The rectangle with the two arrows in the middle of the image represents the robot and the green points correspond to the data given by the laser scanner.

2.4 Omnidirectional Camera

Conventional cameras have restricted fields of view, which make them unadapted or unusable for certain applications in computer vision. Panoramic image sensors are becoming very popular because they provide a wide field of view in a single image and thus are eliminating the need of multiple cameras or for physically rotating a camera with a smaller field of view. There are three different types of systems: catoptric, catadioptric and dioptric. Catoptric systems make only use of mirrors for image formation. They contrast with catadioptric systems which employ both mirrors and lenses in their optics and with dioptric systems which use only lenses. Catadioptric systems are usually used to provide a far wider field of view than using lenses or mirrors alone. Therefore, in this work, a catadioptric system is adopted so as to obtain an omnidirectional image. It consists of a conventional CCD camera and an equiangular mirror. The equiangular mirror is mounted on top of the camera lens, thus providing an omnidirectional view of the robot's surroundings.

In this thesis, the panoramic vision system (explained here and below) is combined with the omnidirectional distance measurement system (see Section 2.3), in order to extract significant features (see Section 3) to perform localization and map building.

A brief description of the equiangular mirror and of the global omnidirectional vision system is given below.

Many types of mirror designs for omni-vision exist: spherical mirrors, ellipsoidal mirrors, hyperbolic mirrors and equiangular mirrors are just some examples. In this work an equiangular mirror is used. The main idea of these mirrors is that each pixel spans an equal angle irrespective of its distance from the center of the image. More information about this kind of mirror can be found in [Chahl97] and [Ollis99]. Our



Figure 2.5: *Omnidirectional camera system mounted on our mobile platforms (i.e. BIBA robot and "SMART" vehicle)*

omnidirectional vision system uses the "360° One VR" mirror from EyeSee360-Kaidan. The system uses a patented OptiFOV™ mirror (i.e. an equiangular mirror) [Herman02]. The panoramic vision system depicted in Figure 2.5 covers 360° degrees of azimuth and up to 100° degrees in elevation.

The mirror-system is mounted on a structure such that when the camera is facing upwards, the mirror axis and the camera axis are identical. In other words, the camera points up through the center of the optic and captures the entire scene.

Given the geometry depicted in Figure 2.6(b), the equation relating the various parameters shown in it (and hence characterizing the shape / configuration of the mirror) is given by:

$$\cos\left(\theta \frac{1+\alpha}{2}\right) = \left(\frac{r}{r_0}\right)^{\frac{1+\alpha}{2}}, \quad (2.6)$$

where r is the ray between the camera and the mirror, r_0 is the distance between the center of the mirror axis and the center of the camera axis and θ is the angle enclosed between the rays r and r_0 . The angle of elevation is the angle of incoming light with respect to the vertical axis of the surface. α is a constant and represents the elevation gain. It is discussed in [Herman02] that for different values of α , mirrors can be produced with a high or a low degree of curvature, while still maintaining their equiangular properties. For this design, the values of the parameters r_0 and α are 14 cm and 11° degrees respectively.

By using an equiangular mirror exact uniformity is obtained, as each pixel on the image covers the same solid radial angle. Thus, if moving radially in the image, the shape of objects is less distorted than it would be if other mirror shapes would be used.

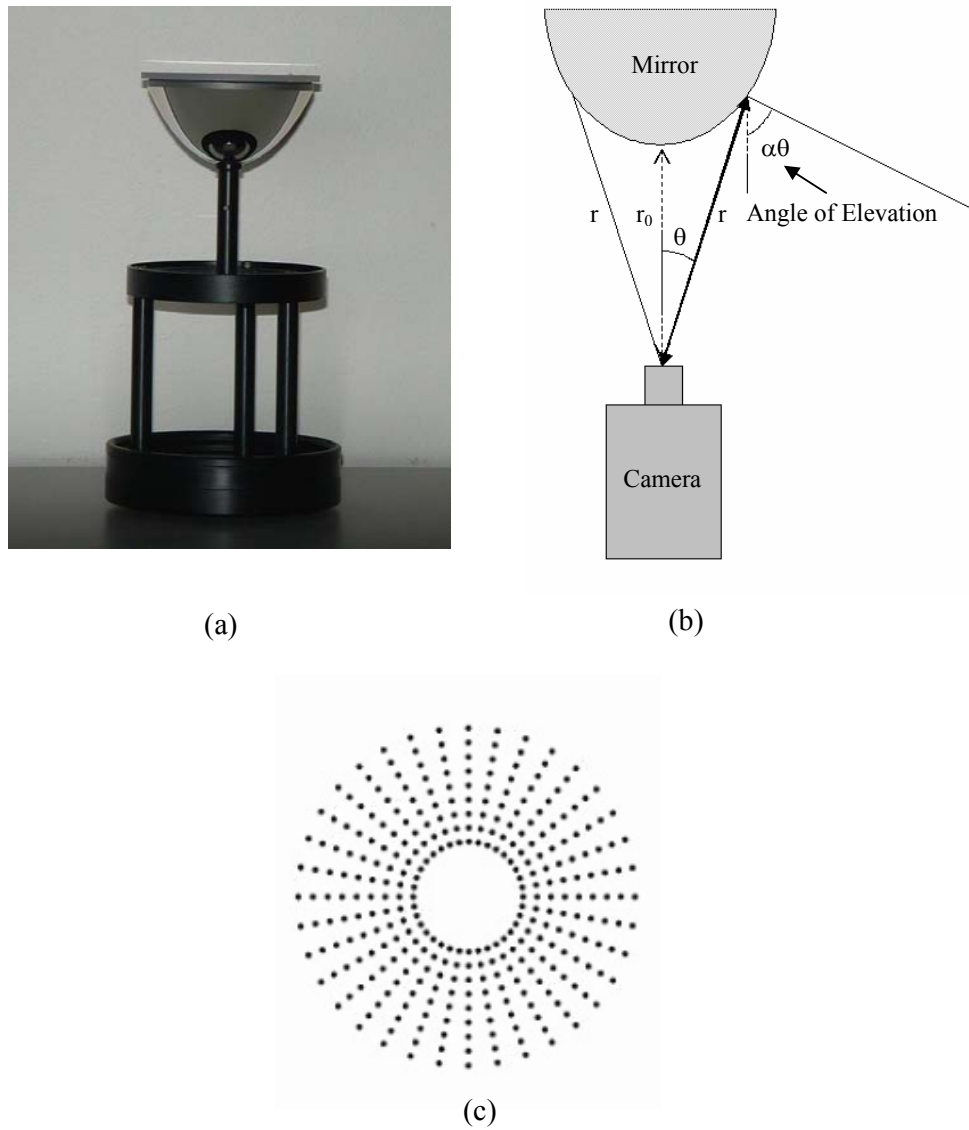
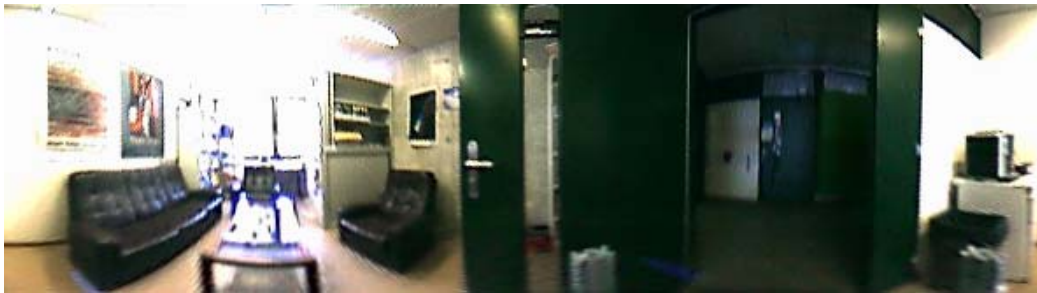


Figure 2.6: *Equiangular mirror: (a) The 360 One VR from EyeSee360-Kaidan. (b) Geometry of an equiangular mirror. (c) The camera's axis and the mirror's axis are identical and lines in the environment parallel to this axis appear as radial lines in the image [Hermann02].*

Figure 2.7(a) illustrates a raw 360° degree image taken with our omnidirectional vision system and Figure 2.7(b) shows the unwarped 360° panoramic image.



(a)



(b)

Figure 2.7: (a) 360° raw image. (b) Panoramic image (unwarped image).

2.5 Summary

This chapter summarizes the models of the exteroceptive and proprioceptive sensors that are used in this work. The quality of a perception system depends more on the quality of the perceived data than the quantity of data obtained. Our perceptual system is a multi-sensory system containing both proprioceptive (wheel encoders) and exteroceptive (laser range finders and omnidirectional vision system) sensors. The odometry for a differential drive system has been described and the non-systematic errors have been modeled. Both exteroceptive sensors, the laser range finders and the panoramic vision system, provide omnidirectional data, from both distance and vision point of view. A catadioptric panoramic-vision system (i.e. it consists of a conventional camera and an equiangular mirror) is introduced in this chapter and used in this work. When moving radially in the image the distortion in the shape of objects is significantly improved, since in an equiangular mirror each pixel spans an equal angle irrespective of its distance from the center of the image. The actual shape of "objects" is better conserved by this mirror due to its main property, than by other kinds of mirrors.

Chapter 3 will describe in more detail, how these sensors are used; what are the significant features that are extracted so as to perform mapping and localization. The main benefit of using a multi-sensory perception approach is that the environment model produced is highly distinctive.

3

Feature Extraction and Environment Modeling

3.1 Introduction

With the help of sensors, the robot is able to sense its environment and take appropriate actions. In order to solve some of the main aspects of the robot-navigation problem, such as localization (estimating the position of the mobile robot or vehicle), mapping (creating a representation of the environment) and SLAM (Simultaneous Localization and Mapping) problem, some form of environment modeling is required. Different sensors can be used, including cameras, global positioning systems (GPS), lasers, sonars, radars, etc, but the main problem that must be addressed is effective employment raw sensory data. Usually, sensors yield an enormous amount of raw data and it is difficult to process it as it is. In the context of navigation, extracting significant, robust and stable features from sensory data is very important. This limits the amount of information used, and consequently the computational time required for processing the given information. The features extracted can vary from low-level features such as circles and lines to relatively high-level ones such as doors and chairs. Thus, the features can be categorized as belonging to different levels of abstraction, depicted using the pyramid shown in Figure 1.1.

The laser range finder and the omnidirectional camera are the two main sensors used in this work. In the following sections, a description of the features that are extracted using them is presented. The feature extraction algorithms presented in Section 3.2 and in Section 3.3 are not novel as they are not meant to be contributions of this thesis. As mentioned before, this thesis aims at developing a new navigation system based on a

compact representation of the environment obtained using a methodology for encoding information from the environment - the *fingerprint of a place*. The following two Sections (3.2 and 3.3) contribute towards this aim. The "fingerprint of a place" concept is presented in detail in Section 3.4.

3.2 Laser Scanner Features

The features extracted from the laser range finder include points, line-segments and infinite-lines. They have been extensively used in the context of navigation [Castellanos96], [Jensfelt99], [Gutmann98]. These are low-level geometrical features, typically employed for pure metric navigation [Arras97], [Castellanos96]. In this thesis, the laser scanner features will be fused with the data extracted from the omnidirectional camera (see Section 3.3), resulting in a compact representation of space – the fingerprint of a place [Tapus04b]. Such a representation is very suitable for topological navigation and hence employed here for the same.

3.2.1 Horizontal Lines

Extraction of horizontal lines is done using a two step process, which involves segmentation and fitting. In the context of this work, segmentation is the process of clustering points belonging to the same line (thus extracting the required line) and fitting refers to the procedure used to find the best line-model parameters describing the points under consideration, as being part of that line. Thus, the quality of the horizontal-line extraction procedure depends on the quality of the segmentation and fitting processes used. An example of a raw scan and the corresponding output, depicting the extracted line segments obtained using the Douglas-Peucker algorithm [Douglas73], is shown in Figure 3.1.

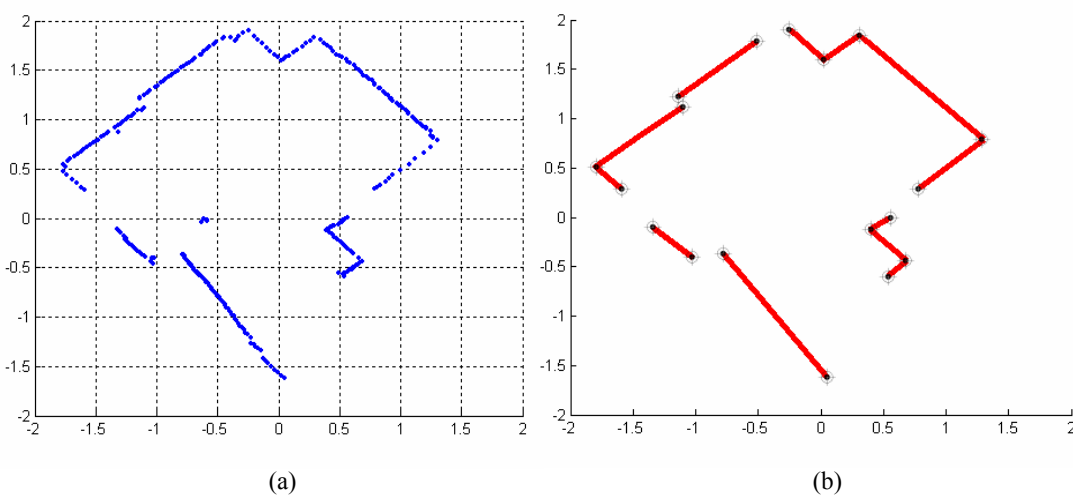


Figure 3.1: (a) 360° raw scan. (b) The extracted lines as obtained using Douglas-Peucker (DP) algorithm

3.2.1.1 Line Model

The line model in the polar coordinates is given as follows:

$$x_i \cos \alpha + y_i \sin \alpha - r = 0 \quad (3.1)$$

where r is the length of the perpendicular to the line passing through the origin (i.e. the origin corresponds to the center of the robot) and α is the angle enclosed by this line with the x -axis, having a range given by $\alpha \in [-\pi, \pi]$. Thus, the line model, in polar coordinates, may be expressed as the parameter set (r, α) . This is depicted in Figure 3.2.

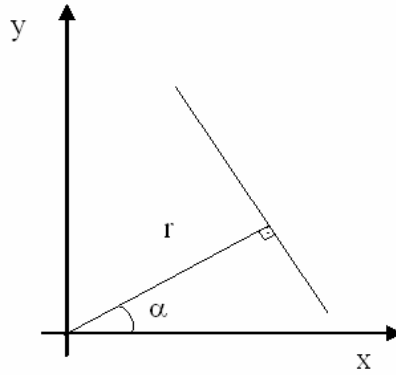


Figure 3.2: *The line model*

The polar representation is used in this work since in [Arras97] it was shown that the polar model (see Equation 3.1) can avoid singularity problems that are implicit in the Cartesian representation.

A given point X_i belongs to the line, if its Cartesian coordinates (x_i, y_i) satisfy the Equation (3.1). If the point X_i is given as polar coordinates (δ_i, θ_i) , knowing the relationship between Euclidean and polar coordinates, we have $x_i = \delta_i \cos \theta_i$ and $y_i = \delta_i \sin \theta_i$. Substituting this result in Equation (3.1) the following line model express in polar form is obtained:

$$\delta_i \cos \theta_i \cos \alpha + \delta_i \sin \theta_i \sin \alpha - r = \delta_i \cos(\theta_i - \alpha) - r = 0 \quad (3.2)$$

The covariance matrix of line parameters is:

$$C_{(r, \alpha)} = \begin{bmatrix} \sigma_r^2 & \sigma_{r\alpha} \\ \sigma_{\alpha r} & \sigma_\alpha^2 \end{bmatrix} \quad (3.3)$$

3.2.1.2 Segmentation with Douglas-Peucker Algorithm

Many algorithms in literature treat the problem of line extraction. They include the Hough Transform [Hough62], Douglas-Peucker (DP) algorithm (also known as split and merge algorithm) [Douglas73], [Castellanos96], RANSAC [Fischeler81],

Incremental [Siadat97], and Expectation Maximization [Pfister03]. A detailed description and comparison of these line extraction algorithms can be found in [Nguyen05]. Among these segmentation methods, the DP and Incremental methods are the best options for SLAM applications (one of the goals of this work), because of their speed and accuracy. For real-time applications, the DP algorithm is definitely the best algorithm as a result of its high computational speed. A description of DP algorithm is given below.

Given a set of point $P = \{p_0, p_1, \dots, p_n\}$, the Douglas-Peucker algorithm proceeds as follows:

A. Recursive Split Step

- a. Approximate the line segment passing through the 2 extreme points $\overline{p_0 p_n}$
- b. Determine the farthest point p_f from the line $\overline{p_0 p_n}$
- c.
 - i. If its distance $d(p_f, \overline{p_0 p_n}) \leq \epsilon$, with $\epsilon \geq 0$, accept $\overline{p_0 p_n}$ as a good line approximation for the set of point P
 - ii. Otherwise, split the set of points p_f into two groups $\{p_0, p_1, \dots, p_f\}$ and $\{p_f, \dots, p_n\}$ and recursively approximate (perform this step – i.e. step A) for each of the sub-groups.

B. Merge Step

- a. If two consecutive segments $\overline{p_i p_j}$ and $\overline{p_j p_k}$ are close enough, approximate the common line $\overline{p_i p_k}$ and determine the farthest point p_{cf} from the line $\overline{p_i p_k}$
- b. If its distance $d(p_{cf}, \overline{p_i p_k}) \leq \epsilon$, merge the segments into a single line segment

C. Prune Short Segments

D. Estimate Line Equations

These steps are described in Figure 3.3. The algorithm works in a recursive manner. The first and the last points from the set of points P are connected by a straight line segment and perpendicular distances from this segment to all the other points are calculated. If none of these distances exceed a previously specified tolerance, then the straight line segment is deemed suitable to represent the set of points P. If this condition is not met, then the point with the greatest perpendicular offset from the straight line segment is selected, and the set of points P is split into two new sub-sets, containing respectively, all points between the first and the selected point and those between the selected and the last point. The procedure (known as the `split` step) is then repeated. The complexity of the recursive split-step is given by $O(N \log N)$, where N is the number of points under consideration. When the split step is completed, there will be many segments. Arbitrary division of the set of points into line segments might result in an `excessive` split operation. The merge step, then, tries to fix this by considering consecutive segments that were not compared with each other during the

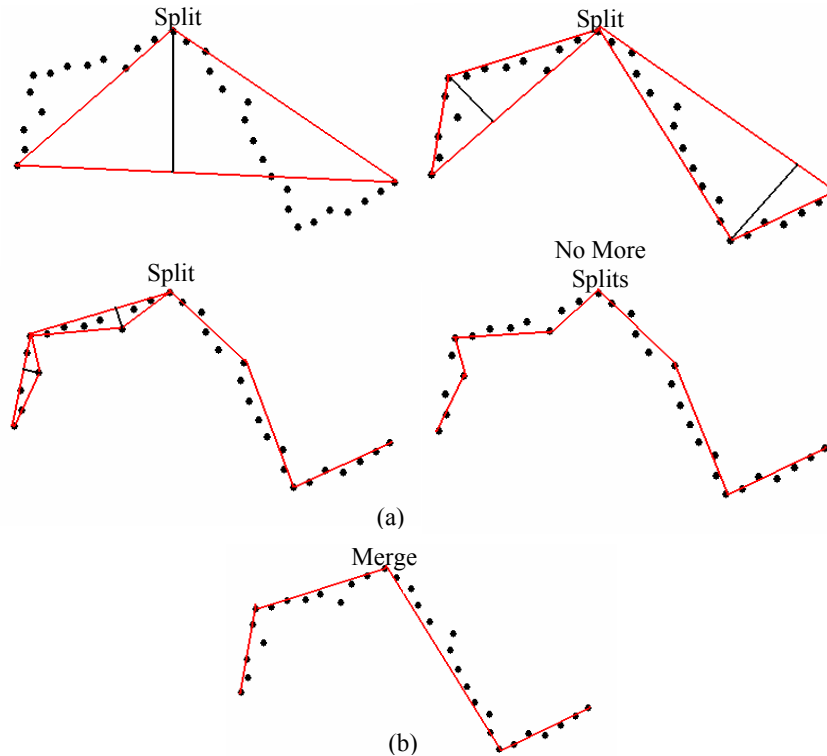


Figure 3.3: Douglas-Peucker (DP) algorithm (Split and Merge): (a) Recursive Split. (b) Merging

split phase, and merging them if they are close enough. In order to obtain good repeatability and correctness, a compulsory demand that must be met when trying to do SLAM is that the short segments (i.e. the segments composed of few points) must be eliminated. To estimate the line parameters, the *total-least-square* method is used. This approach is a well known and widely used one. More details can be found in [Arras97], and [Jensfelt99].

3.3 Omnidirectional Camera Features

For many reasons, vision is an interesting and attractive choice of sensory input, in the context of robot navigation. In addition to vision being the primary functionality used by humans and animals to navigate, intuitively, vision seems to be the most appropriate sensory input for natural everyday environments. Vision based navigation involves acquisition and processing of visual information. Even though it is more difficult to reliably extract geometric information from images (for SLAM), as compared to other popular sensors, vision can play a crucial role in the context of autonomous navigation. Visual features contain significant information (i.e. verticality, horizontality, color, texture, shape etc.) that can be very useful for navigation. In this thesis, the omnidirectional camera is used to extract vertical edges and color patches. Vertical edges are low-level geometric features. In contrast, color patches are more complex features that are intrinsically more distinctive for the environment modeling.

In the following two sub-sections, a description of the methods used to extract vertical edges and color patches is presented.

3.3.1 Vertical Edges

Edges characterize boundaries and are therefore of fundamental importance in real world environments (e.g. indoor office buildings, outdoor urban environments). Due to the necessity for robustness and stability when working with autonomous mobile robots, only features which are invariant to the effects of perspective projection must be used for environment modeling and later for SLAM. This will enable view-independent feature recognition. If the camera is mounted on a mobile robot or on an outdoor vehicle with its optical axis parallel to the ground plane (i.e. floor – used in this work), the vertical edges have the view invariance property, contrary to the horizontal ones. The vertical edges, extracted with the camera, permit a low-level geometrical description of the environment and they have been extensively used in prior work in vision-based localization [Ulrich00], [Lamon01]. However, vertical edges in real world environments, especially indoor environments, are numerous. This can result in many noisy ones being detected, thus leading to the formation of a faulty environment model.

For this reason, the approach for edge detection adopted in this work is composed of the following steps:

1. **Filtering:** used to improve the performance of an edge detector with respect to the noise. A mean filtering is used since it's a simple, intuitive and easy to implement method of *smoothing* images.
2. **Enhancement:** used to facilitate the detection of edges, by determining changes in the intensity in the neighborhood of a point.
3. **Detection:** used to detect the edges by applying non-maxima suppression to a histogram representation.

The flow chart showing the process used for vertical edge detection is illustrated in Figure 3.4 and a more detailed description of the different steps is given below.

3.3.1.1 Edge Enhancement

Many methods for edge enhancement can be found in literature. One of the most optimal approaches is Canny edge detector [Canny86]. The Canny algorithm is based on a set of criteria which include finding the most edges by minimizing the signal-to-noise ratio, selecting edges as closely as possible to the actual edges to maximize detection and extracting edges only once, when a single edge exists for minimal response. According to Canny, the optimal filter that meets the three criteria mentioned above can be efficiently approximated using the first derivative of a Gaussian function.

One of the main problems in autonomous robotic systems is the computational time required for processing the algorithms performing different tasks. Image processing is one of the most computationally expensive processes. In order to avoid calculating the gradient for an interpolated point between pixels, a 3x3 convolution mask is used for

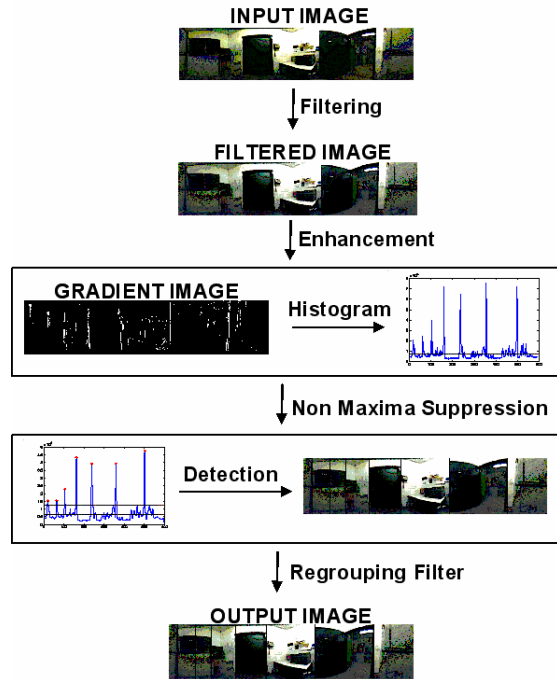


Figure 3.4: Flow-chart showing the different steps used for vertical edge extraction

the gradient calculations in the x-direction (columns) and y-direction (rows). For this operation, the Sobel operator is employed (see Figure 3.5). A convolution mask is usually much smaller than the actual image. The mask is slid over the image, manipulating a grid of 9 pixels at a time.

The example below (see Figure 3.5) shows the mask being slid over the top left portion of the input image represented by the red outline. The Equation (3.4) shows how a particular pixel in the output image would be calculated.

$$b_{22} = (a_{11} \times m_{11}) + (a_{12} \times m_{12}) + K + (a_{32} \times m_{32}) + (a_{33} \times m_{33}) \quad (3.4)$$

The center of the mask is placed over the pixel to be manipulated in the image.

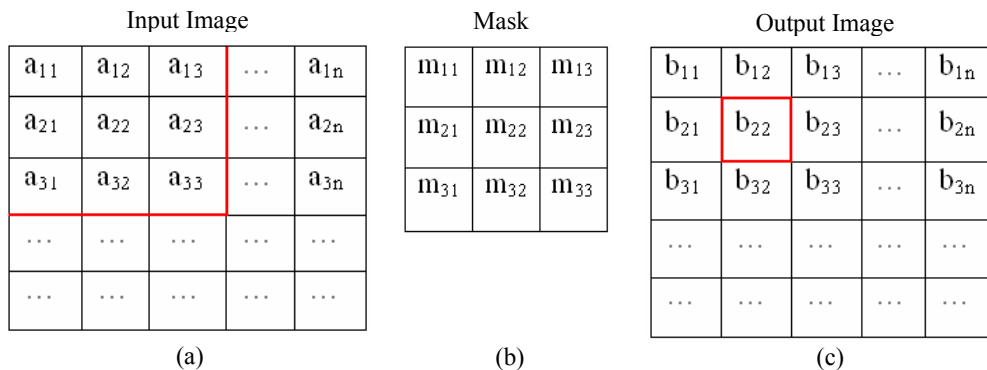


Figure 3.5: (a) The input image. (b) The mask that will be applied. (c) The output image.

The actual Sobel masks, in x-direction and y-direction, respectively, are shown below:

-1	0	1
-2	0	2
-1	0	1

G_x

1	2	1
0	0	0
-1	-2	-1

G_y

Figure 3.6: Sobel Operator in x-direction (G_x) and in y-direction (G_y)

and the magnitude of the gradient is calculated as follows:

$$|\vec{G}| = \sqrt{G_x^2 + G_y^2} \quad (3.5)$$

Since gradient computation is based on the calculation of intensity values sensitive to noise (due to the diode light sensitivity in the digital camera's sensor chip), a process of channel mixing (i.e. red and green channels) is performed. This permits the amelioration of the edge detection algorithm.

For vertical edges, only the mask along the x-direction (columns), G_x , is taken into account. The gradient magnitude for the vertical direction of the image described in Figure 3.7(a) is illustrated in Figure 3.7(b).



(a)



(b)

Figure 3.7: (a) A CCD color image depicting a part of our lab. (b) The corresponding gradient magnitude image after applying the Sobel filter on the x-direction (columns).

3.3.1.2 Image Histogram and Thresholding

This step consists of regrouping the resulting edge fragments together so as to obtain segments of vertical edges. A vertical histogram of the image is constructed by adding the gradient magnitude of the pixels for each column. Figure 3.8(a) describes the histogram obtained for the image depicted in Figure 3.7(a). Significant peaks in this histogram correspond to meaningful clusters of vertical edges. However, the histogram depicted in Figure 3.8(a) is noisy.

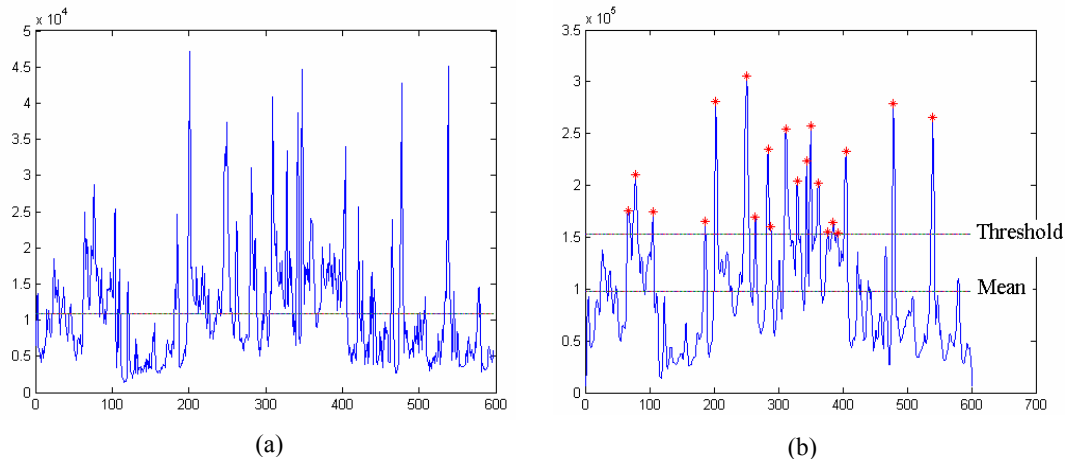


Figure 3.8: (a) The histogram of the gradient magnitude of pixels in the vertical direction. (b) The smoothed vertical histogram, with the mean, the threshold and the local maxima (red stars).

A triangle filter, $T = [1 \ 2 \ 3 \ 2 \ 1]$, is used to smoothen the histogram. It also ensures that the high values of the spikes are retained (see Figure 3.8(b)).

In order to reduce the number of false edge fragments a threshold is applied to the histogram gradient magnitude values. All values below the threshold are ignored. The threshold is adaptable and is calculated by using a statistical approach as detailed in [Lamon01]. This method uses the addition between the standard deviation of the values of the histogram and the mean of the histogram so as to fix the threshold. Figure 3.8(a) contains also the mean and the threshold values.

3.3.1.3 Non-Maxima Suppression and Regrouping Filtering

The histogram contains the gradient magnitude of pixels having the vertical direction. In order to identify the most probable edges, only the local maxima above the threshold, fixed in Section 3.3.1.2, are selected. Figure 3.9 illustrates this process and depicts all the extracted edges.



Figure 3.9: *The original image with the extracted vertical edges, corresponding to the local maxima*

Since the extraction of vertical edges must be robust and stable for a later use in Section 3.4, the sensitivity to occlusion and noise must be limited. A *regrouping filter* is applied on the histogram obtained after the thresholding process. This enables the grouping of edges closer than a fixed *constant* value which represents the minimum distance between the edges. The final set of extracted vertical edges is shown in Figure 3.10.



Figure 3.10: *The original image with the extracted vertical edges after applying the regrouping filter*

3.3.2 Color Patches

Another important feature that can be extracted from the camera sensor is color. It may be one of the most natural features used by humans for visual recognition and discrimination and it is context dependent. It is an intuitive feature that can give an effective and compact representation by itself. Computer vision performs color extraction without the benefit of a context. Lack of knowledge also makes it difficult to distinguish true color information from color distortion. The appearance of the color of real world objects is generally altered by surface texture, lighting, shading effects, and viewing conditions.

In our work, we are interested in identifying the regions within images that contain colors from a predetermined color set. The fixed color set is illustrated in Figure 3.11.

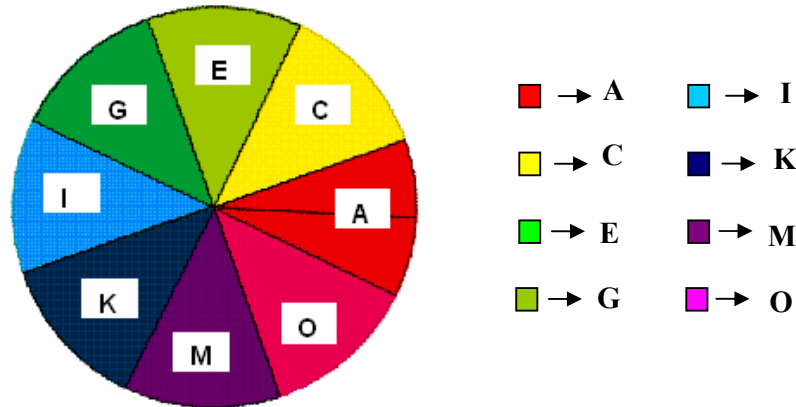


Figure 3.11: The predetermined color set - 8 colors ($hue \in [0,255]$). The characters *A* to *O* indicate the hue of colors (*A*=red, *C*=yellow, *E*=light green, *G*=dark green, *I*=light blue, *K*=dark blue, *M*=violet, *O*=magenta)

We choose to work with a set of eight color bins, which is a good tradeoff (i.e. a number too small would not have exploited all the colorimetric information and a number too big of color patches would have resulted in a high computational process time). A similar approach for color patch extraction is also described in [Lamon01].

3.3.2.1 Color Spaces

The RGB (i.e. a superposition of red, green and blue colors) color format is the most common color format for digital images, because it is compatible with computer displays. However, the RGB space has a major shortcoming in that it is not perceptually uniform (i.e. there is a low correlation between the perceived difference of two colors and the Euclidian distance in the RGB space). In contrast, other color spaces that offer improved perceptual uniformity, such as CIE-LAB and CIE-LUV can be used. In general, they represent with equal emphasis, the three color variants that characterize color - hue, lightness, and saturation. The hue channel calculates the apparent color of the light, as determined by the dominant wavelengths, the saturation is a term used to specify the purity of the light and the intensity represents the total light across all frequencies. This separation is attractive because color image processing performed independently on the color channels does not introduce false colors and it is easier to compensate color distortions. For example, lighting and shading distortions will typically be isolated to the lightness channel. Therefore, for color extraction we use the HSI (Hue, Saturation and Intensity) color space because it has the above mentioned characteristics. More details on RGB to HSI color space conversion can be found in [Jain95].

3.3.2.2 Saturation Thresholding

Saturation is an important element in the HSI color space and it defines the purity of color. As saturation increases, colors appear more "pure." As saturation decreases,

colors appear more "washed-out." The high values of the saturation are usually obtained after the color space conversion (RGB to HSI) when there is a great difference between RGB values and not only because the color is truly saturated. For this reason the pixels with very high and very low saturation values are ignored and will not be taken into account for the rest of the color bin detection algorithm (see Figure 3.12)

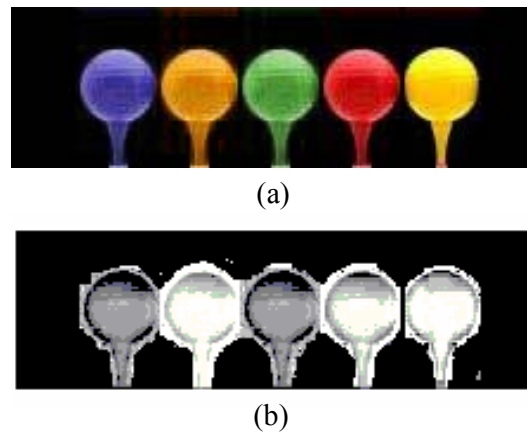


Figure 3.12: (a) The input image composed of 5 golf balls of different colors. (b) Saturation image after the thresholding process (the saturation values are in between 100 and 250)

3.3.2.3 Hue Histogram

The color distribution in the scene is in general not known in advance and can cover the entire color space. One common method for characterizing image color is to use a hue histogram. The hue histogram for each base color depicted in Figure 3.11 is constructed by counting the number of pixels of each color. Only the pixels that passed the thresholding saturation process are taken into account. Each of these pixels will vote in one or two histograms depending on the value of the hue (e.g. a pixel with hue=0 will add 100 in the corresponding column of the red histogram, and a pixel with hue=7 will add a bigger column in the red histogram than in the yellow one). A description of this voting process can be viewed in Figure 3.13.

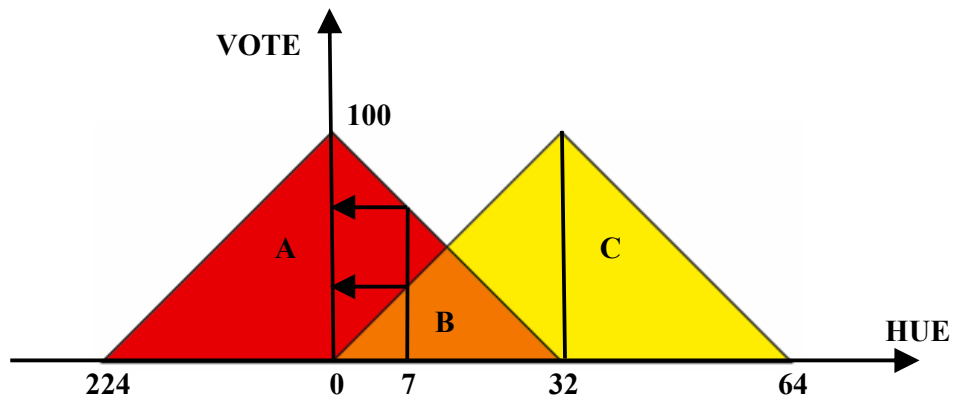


Figure 3.13: Fuzzy voting process

For histogram peak detection, the same methodology as the one described in Section 3.3.1.2 for vertical edge detection is also used here. Some parameters are adapted for the case of color patches. Therefore, instead of using a triangular filter, a trapezoidal filter $T2=[1 \ 2 \ 2 \ 2 \ 1]$ will be employed, because we want to smoothen out thin peaks in this case.

The Figure 3.14 shows the histograms generated with our method for the image shown in Figure 3.12(a).

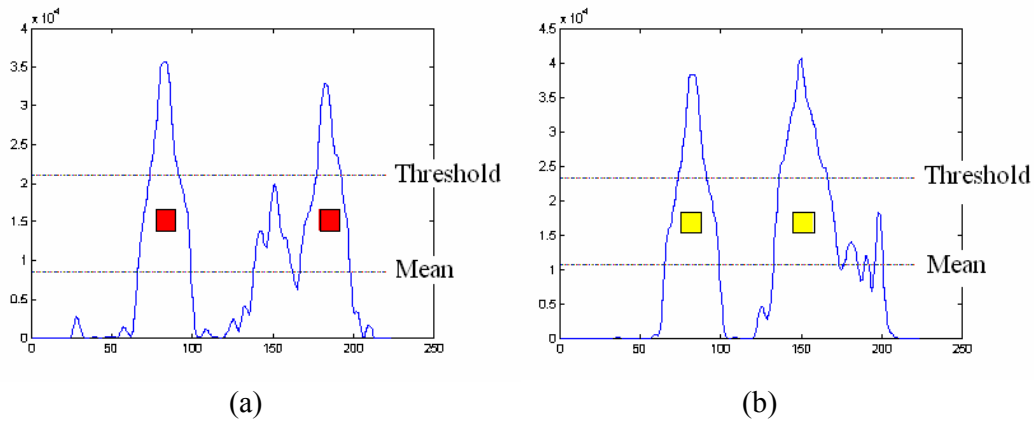


Figure 3.14: *Color Histogram. (a) Red color histogram ('A'); (b) Yellow color histogram ('C')*

3.3.2.4 Color Fusion

Figure 3.15(a) shows that the number of color patches detected in Figure 3.12(a) is higher than the number of color patches expected. This means that some golf balls voted in two different color histograms. This phenomenon can be noticed in Figure 3.14, when the second golf ball voted both in the red and in the yellow histogram. The process of keeping both detected color patches is very critical for environmental modeling (i.e. an inversion between patches can occur, and hence the representation of the environment can appear different). For this reason, we choose to fuse two consecutive color patches that have a difference between their patch coordinates smaller than a user predefined threshold. In this example (see Figure 3.14), the red patch and the yellow patch are fused and a new patch is thus obtained, the orange patch, denoted by the character 'B'.

Figure 3.15(b) illustrates the sequence of color patches detected after the color fusion performed Figure 3.15(a). It can be seen that the sequence of colors obtained in Figure 3.15(b) matches exactly to the input image as depicted in Figure 3.12(a).



Figure 3.15: (a) Color patches detected; (b) Color fusion

3.4 Environment Modeling with Fingerprint of Places

Most of the time, with the exception of reactive-behavior based navigation a space representation of the environment is needed in order to localize the robot. The notion of the *fingerprint of a place* is introduced here. This concept was first proposed in [Arras97] and [Lamon01]. The fingerprint approach, by combining the information from all sensors available to the robot, reduces perceptual aliasing and improves the distinctiveness of places. This manner to represent the environment is defined and described in the followings sub-sections.

3.4.1 Fingerprint of a Place Definition

Just as each person has a unique fingerprint, each location in the environment also has a unique set of characteristics associated with it. Of course, when relying on the limited perceptual capabilities of a machine, it is difficult to guarantee the unique distinction between two similar places. Our system assumes that a *fingerprint* of the current location can be created and that the sequence generation methods can be made insensitive to small changes in robot position. However, this characterization of the environment is especially interesting when used within a topological framework. In this case the distinctiveness of the observed location plays an important role for reliable localization and consistent mapping.

A fingerprint of a place is a circular list of features, where the ordering of the set matches the relative ordering of the features around the robot. We denote the fingerprint sequence using a list of characters, where each character represents an instance of a specific feature type. Figure 3.16 depicts an example of a fingerprint of a place.

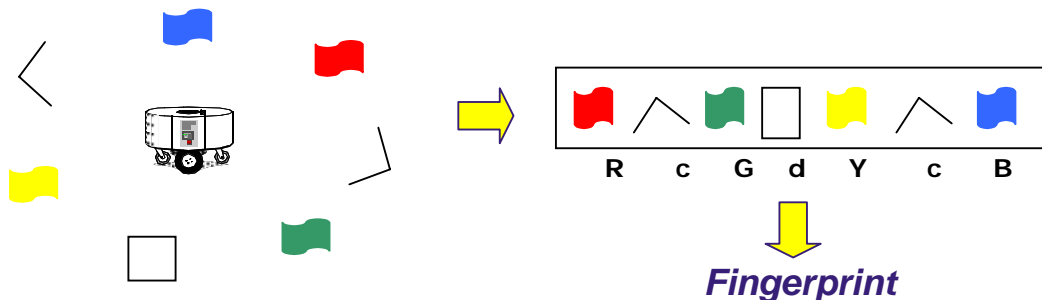


Figure 3.16: Fingerprint concept overview

3.4.2 Fingerprint Encoding

As previously mentioned, a fingerprint of a place is a circular list of features that the robot can perceive around it. In this thesis a fingerprint is created by assuming that a set of feature extractors (as described in the previous sub-sections) can identify significant features in the environment around the robot. Omnidirectional sensors are preferred because the orientation as well as the position of the robot may not be known a priori. In this work, we choose to extract color patches and vertical edges from visual information and corners (i.e. extremity of line-segments) from laser scanner. The letter 'v' is used to characterize an edge, the letters 'A','B','C',..., 'P' to represent hue bins and the letter 'c' to characterize a corner feature (i.e. in this work, a corner feature is defined as the extremity of a line-segment extracted with the Douglas-Peucker algorithm). Details about the extraction of visual features can be found in Section 3.3 and that of features extracted using laser scanner in Section 3.2.

3.4.3 Fingerprint Generation

Fingerprint generation is performed in three steps, as shown in Figure 3.17. The extraction of the different features (e.g. vertical edges, corners, color patches) from the sensors is the first step of the fingerprint generation process. The extracted features are ordered in a sequence depending on their angular position ($0 \dots 360^\circ$). In the second step, a new type of feature, the virtual feature 'f' is introduced. This reflects the correspondence between a corner (detected with the laser scanner) and an edge (detected in the unwrapped omnidirectional image). In order to represent large ($> 20^\circ$ degrees, in our case) angular distances between successive fingerprint elements, the notion of an 'empty space' feature is added. This is denoted in the fingerprint sequence by the character 'n'. In this way, the ordering of the features in a fingerprint sequence becomes highly informative, thereby increasing distinctiveness of fingerprints. This insertion is the last step of the fingerprint generation process.

3.4.4 Uncertainty Modeling in the Fingerprint

The interaction between a mobile robot and its surroundings is performed by means of exteroceptive sensor data. Sensors are imperfect devices, and thus the measurements always contain errors. This can be modeled by associating uncertainty to their data. For this reason, probabilities will be used to model the uncertainty of the geometric features extracted from the environment. We define the uncertainty as the probability of a feature of being present in the environment when the robot perceives it. In our fingerprint approach, this idea is incorporated by associating every observed feature (for each of the different types of features mentioned above) with an uncertainty measure. These uncertainty measures are modeled by experience, for each type of feature presented in Figure 3.17: vertical edges, colors, corners (extremities of the segments), 'f' feature and respectively 'n' feature.

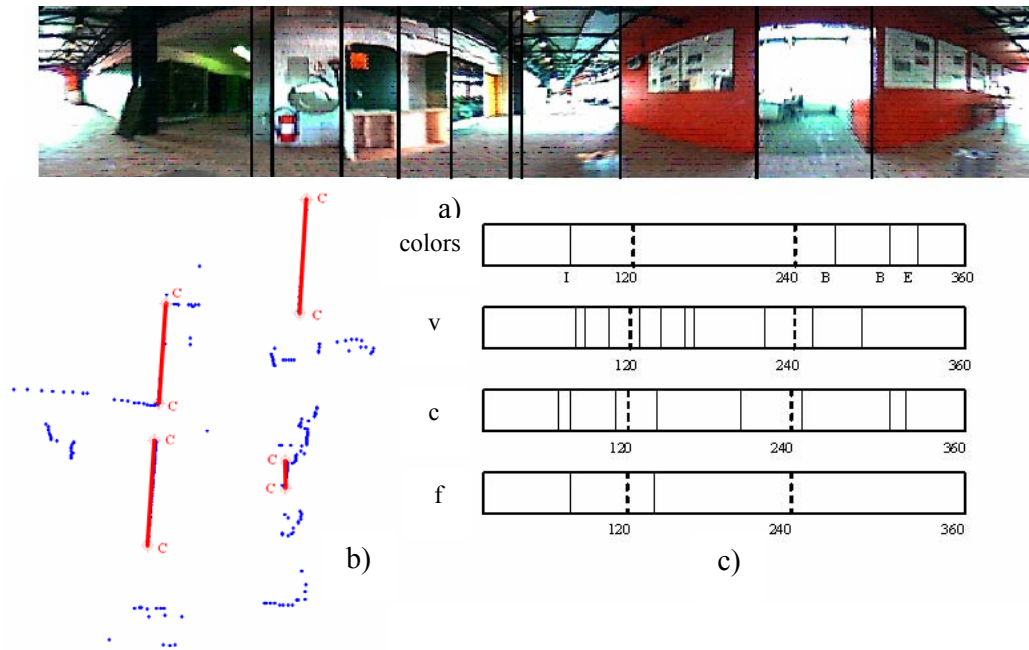


Figure 3.17: Fingerprint generation. (a) panoramic image with the vertical edges and color patches detected, denoted by 'v' and 'A'...'P', respectively ; (b) laser scan with extracted corners 'c'; (c) the first three images depict the position (0 to 360°) of the colors (I-light blue, B- orange and E-light green), vertical edges and corners, respectively. The fourth image describes the correspondence between the vertical edge features and the corner features. By regrouping all these results together and by adding the empty space features, the final fingerprint is: *clfvnvcyfnvvnvcvnnvcvBnvBccE*

The method used to associate each feature with an uncertainty measure is explained below:

- For the first three types of features (vertical edges, colors and corners) the uncertainty is calculated by using the following scheme.

$$u = \begin{cases} \min, & \text{if } extraction_value \leq low_bound \\ linear_interpolation, & \text{if } low_bound \leq extraction_value \leq high_bound \\ \max, & \text{if } extraction_value \geq high_bound \end{cases} \quad (3.6)$$

The *extraction_value* variable changes in function of the type of the feature. For the vertical edges, *extraction_value* corresponds to the gradient value. For the colors, the *extraction_value* is represented by the hue value of the color. In the case of the corner features, the *extraction_value* is identified as the distance between the robot and the extremities of the segments. The values of the *low_bound* and *high_bound* are experimentally determined for each type of feature. The value of *low_bound* represents the bound below which the feature has a low probability of existence so that the robot may not see it when it passes it again. Another important element is the *high_bound*, above which a feature has a high certainty to exist and to be in the place where it was found (extracted). The *low_bound* and the *high_bound* are determined for each feature at the extraction level. The extraction of the vertical edges consists of the application of a threshold function on the gradient values. Since all edges below the threshold are ignored, *low_bound* is used as the threshold value. The value *high_bound* must be high, but not *max_gradient*, so it has been fixed experimentally at $(threshold + mean_gradient)$. To extract the colors, a threshold function on the hue values has been applied and a similar method to that applied for the vertical edges has been chosen. Two other important elements in the formulation described before are the values of *min* and *max*, which are fixed to 0.6, respectively 0.99. Values between the *min* and *max* values may be computed by linear interpolation, as given below:

$$u = \min + \frac{extraction_value - low_bound}{high_bound - low_bound} \times (max - \min) \quad (3.7)$$

- As the 'f' feature reflects the correspondence between a corner and an edge, its uncertainty is defined as the mean value between the uncertainty of the corner and the uncertainty of the vertical edge feature.
- The last feature is the 'n' feature (i.e. the empty space feature that represents the angular distance between the features). The uncertainty of this feature is defined as being proportional to the distance between the features.

In this way, the uncertainty of the features used in the fingerprints is calculated. The limitation of this method resides in the models, which are difficult to define, especially for our definition of uncertainty, which cannot be directly derived from the physical characteristics of the sensors.

3.5 Summary

This chapter presented different methods for feature extraction and the way these features are combined into a compact representation, the fingerprint of a place. This environment model is the basis of our navigation system and will be used in the next chapters for localization, mapping and SLAM. Features from both the laser range finder (i.e. the horizontal edges) and the omnidirectional camera (i.e. the vertical edges and the color patches) have been extracted and will be used for space representation. The fingerprint provides a compact and distinctive methodology for space representation – it permits encoding of a huge amount of place-related information in a single circular sequence of features. In this work, low-level features (such as vertical edges, horizontal lines) are used. Other modalities and sensory cues can also be added to the fingerprint framework (e.g. auditory, smell, or higher level features such as doors, table, fridge) and thereby improve the reliability and accuracy of the method. This representation is suitable for both indoor and outdoor environments. The fingerprint-based environment modeling is extensible in that it permits spatial cognition beyond just pure navigation.

4

Topological Localization with Fingerprints of Places

Navigation poses several challenges - one of the key issues when talking about navigation is answering the question “Where am I?” (also known as the localization problem). The robot localization issue is a very important problem in the development of truly autonomous robots. Localization is the task of determining robot’s position with respect to some underlying representation. This representation is map situation-based and may comprise of anything - constellations of stars, a network of GPS satellites or even a simple map of an indoor (typical household/office) environment. If a robot does not know where it is relative to its environment, it is difficult for it to decide what to do next. For many roboticists, the problem of robot localization has been the most fundamental problem in the development of truly autonomous robots. There are many methods to answer this question, each having a different precision (i.e. metric and topological, more details are given in Section 1.2). This chapter looks at how the question “Where am I?” can be answered, given that a topological representation of the environment is used in conjunction with probabilistic techniques (to model the uncertainty in the environment).

4.1 Related Work

Finding an efficient solution to the robot localization problem will have a tremendous impact on the manner in which robots are integrated into our daily lives. Most tasks for which robots are well suited demand a high degree of robustness in their localizing capabilities before they are actually applied in real-life scenarios.

Since localization is a fundamental problem in mobile robotics, many methods have been developed and discussed in literature. These approaches can be broadly classified into three major types: metric, topological and hybrid. Approaches using metric maps are useful when it is necessary for the robot to know its location accurately in terms of metric coordinates (i.e. Cartesian coordinates). However, the state of the robot can also be represented in a more qualitative manner, by using a topological map (i.e. adjacency graph representation). Thus, the problem of localizing a robot given a topological map is equivalent to determining the node in the graph that corresponds to the robot's current location. Recently, researchers have integrated both the metric and topological paradigms, thereby obtaining a hybrid representation.

Most of the mobile robot localization approaches focus on metric localization. Metric localization approaches can be based on either model-matching or on landmarks. Most of the model-matching based methods use an Extended Kalman Filter (EKF) that combines the exteroceptive and proprioceptive information from sensors so as to determine the current position of the robot [Arras00], [Jensfelt99], [Perez99]. These systems need a good statistical sensor model; the uncertainties (of the recorded sensor data) they yield, will be provided to the Kalman filter. Recent advances in Kalman filtering permit for non-Gaussian multimodal probability distributions through multiple hypothesis tracking. Markov localization and Monte Carlo localization are two of them. Markov localization [Fox98] is well suited for globally estimating the position of a mobile robot based on its observations and its actions and has the ability to relocalize a robot if its position is lost. Monte Carlo localization (localization using particle filters) yields a method of representing multimodal distributions for position estimation. This concept has been demonstrated in [Dellaert99] and [Thrun01]. These works also take advantage of the fact that the computational time required, is independent of the area within which localization has to be performed.

Landmark based localization can use two types of landmarks: "artificial" and "natural". Natural landmarks are features that are already present in the environment and may/may not have other totally distinct roles, as against their utility for robot navigation. Artificial landmarks are features (such as beacons, color signs, RFID's, etc) placed in the environment specifically for the purpose of robot navigation. Therefore, artificial landmarks are much easier to reliably detect (they can be chosen appropriately) than natural landmarks. The usage of artificial landmarks requires the modification of the environment within which the robot will move and hence the natural landmarks are preferred. Typically used examples of natural landmarks include: corners [Tomatis03], doors [Althaus03], ceilings [Abe99]. One of the main drawbacks of the existing landmark-based localization approaches is the fact that they are made for specific environments, they are thus not generic and cannot be used as is for different types of environments (i.e. both indoor and outdoor).

Topological approaches attempt to overcome the drawbacks of geometric methods by modeling space using graphs. Significant progress has been made since the seminal paper by Kuipers [Kuipers78]. Next, a brief description of several topological methods to perform place recognition is presented. Kortenkamp and Weymouth in [Kortenkamp94] have proposed an approach based on concepts derived from a theory of human cognitive mapping that also involved topological navigation. They have used the data from the sonars combined with vision information in order to achieve a rich sensory place characterization. Their work is an amelioration of Mataric's approach in [Mataric90]. The main goal of their work was the reduction of the perceptual aliasing problem, the improvement obtained by introducing more sensory information for place

representation. A model by Franz, Schölkopf and Mallot [Franz98] was designed to explore open environments within a maze-like structure and to build graph-like representations. Their method has been tested on a real robot equipped with an omnidirectional camera. Place recognition was done by comparing the current observation to the stored omni-directional snapshots. The model described in [Choset01] represents the environment with the help of a generalized Voronoi graph (GVG) and localizes the robot via a graph matching process. In [Ulrich00], Ulrich and Nourbakhsh present an appearance-based system for topological localization. An omnidirectional camera was used. Furthermore, Tomatis et al. have conceived a hybrid representation of the environment [Tomatis03]. This comprises local metric maps interconnected via topological relations. Their system uses a camera and two laser range finders to perform the recognition of doors and corners.

This chapter presents several topological localization techniques based on the fingerprint approach. In contrast to most of the previously presented methods, our fingerprint-based methods combine multimodal perceptual information and perform well both in indoor and outdoor environments. The algorithms presented in this chapter operate on a graph-like representation of the world. These approaches have also been illustrated in depth in the works [Tapus04a] and [Tapus04b].

4.2 Environment Model

The environmental model used, plays a very important role for determining the precision and for performing the localization. In this work, it was chosen to represent the environment in a topological fashion. The topological map can be viewed as a graph of places, where at each node the information concerning the visible landmarks and the way to reach other places connected to it, is stored. The *fingerprint of a place*, a circular list of features around the robot, described in Section 3.4, can be easily used to represent places and therefore the nodes in the topological framework. Thus, the topological representation is compact and allows high level symbolic reasoning for map building and navigation.

4.3 Localization with Fingerprints of Places

Localization is the task by which, given a map, an active agent determines its current location with respect to the map. As mentioned earlier, in this work, *fingerprints of places* are used (as representations of the environment) to build an environment model. This model is subsequently used to localize the robot. Localization on a fingerprint-based representation is reduced to a problem of fingerprint matching.

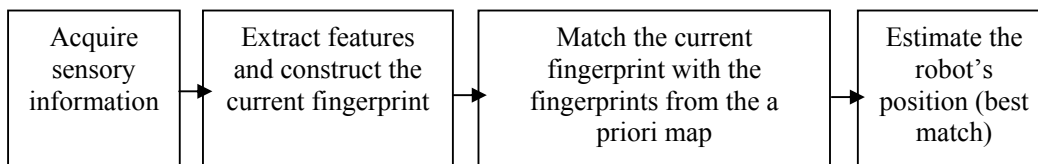


Figure 4.1: General procedure for fingerprint-based localization

The next sections illustrate different localization methods using fingerprints of places, the results obtained and a comparison between them. Figure 4.1 depicts the general procedure used for fingerprint-based localization.

4.3.1 Fingerprint Matching using Bayesian Programming

This section presents a Bayesian localization algorithm for a topological (fingerprint based) environment model. The approach also describes how multimodal perception increases the reliability of topological localization (using the Bayesian Programming formalism) for mobile robots.

4.3.1.1 Bayesian Programming (BP) Formalism

When programming a robot, the programmer constructs an abstract representation of the robot's environment. This representation may be geometrical, analytical or symbolic in nature. In a way, the programmer imposes upon the robot, his or her own abstract conception of the environment. Difficulties appear when the robot needs to link this representation with the sensory input and use the same to send output signals to its actuators. The underlying cause for all of these problems lies in the irreducible incompleteness of the model. Controlling the environment is the usual answer to these problems. However, this is not a desirable solution as the robot may be expected to function in environments not specifically designed for it, populated environments or even under situations with unexpected events occurring.

Probabilistic methodologies and techniques offer possible solutions to the incompleteness and uncertainty problems encountered when programming a robot. The basic programming resources are *probability distributions*.

The Bayesian Programming (BP) approach was originally proposed as a tool for robot programming (see [Lebeltel04]), but nowadays it is used in a wider range of applications ([Mekhnacha00] shows some examples).

In this approach, a probability distribution is associated with the uncertainty of a logical proposition value. The usual notion of a *Logical Proposition* (true or false) and its operators (conjunction, disjunction and negation) are used to define a *Discrete Variable*. A *Discrete Variable* X is a set of logical propositions x_i , such that these propositions are mutually exclusive (i.e. for all i, j with $i \neq j$, $x_i \wedge x_j$ is false) and exhaustive (at least one of these propositions x_i is true).

The probability distributions assigned to logical propositions are always defined according to some preliminary knowledge, denoted by π . The probability $P(x_i | \pi)$ gives the probability distribution of the variable X having the value x_i , knowing π . Probabilities will be manipulated using the Bayes rule. More details about the inference postulates and rules for carrying out probabilistic reasoning in this context can be found in [Bessière03] and [Bellot03].

The Bayesian Programming formalism enables the usage of a uniform notation and provides a structure to describe probabilistic knowledge and its use. The elements of a Bayesian Program are illustrated in Figure 4.2. A BP is divided in two parts: a description and a question.

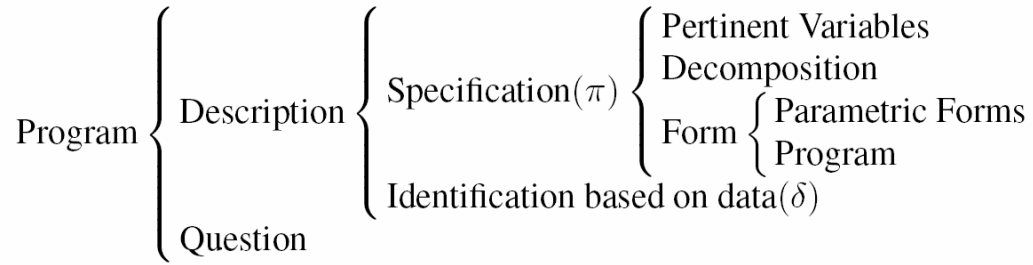


Figure 4.2: Structure of a Bayesian Program (see [Bessière03])

A. Description

The purpose of a description is to specify an effective method to compute a joint distribution on a set of relevant variables $\{X_1, X_2, \dots, X_n\}$, given a set of experimental data δ and a priori knowledge π .

In the specification phase of the description, it is necessary to:

- Define a set of relevant variables $\{X_1, X_2, \dots, X_n\}$, on which the joint distribution shall be defined;
- Decompose the joint distribution into simpler terms, using the conjunction rule. The conditional independence rule can allow for further simplification, and such a simplified decomposition of the joint distribution is called *decomposition*.
- Define the forms for each term in the decomposition; i.e. each term is associated with either a parametric form (a function) or another Bayesian Program.

B. Question

Given a description $P(X_1, X_2, \dots, X_n \mid \delta \pi)$, a question is obtained by partitioning the variables $\{X_1, X_2, \dots, X_n\}$ into three sets: *Searched*, *Known* and *Unknown* variables. A question is defined as the distribution $P(\text{Searched} \mid \text{Known} \delta \pi)$. In order to answer this question, the following general inference is used:

$$P(\text{Searched} \mid \text{Known} \delta \pi) = \frac{\sum_{\text{Unknown}} P(\text{Searched} \text{Unknown} \text{Known})}{\sum_{\text{Unknown}, \text{Searched}} P(\text{Searched} \text{Unknown} \text{Known})} \quad (4.1)$$

Depending on the number of variables (and its discreteness) and the decomposition choice, this calculation may need a lot of computational time and turn out to be infeasible. Numerous techniques have already been proposed to achieve an admissible computation time. A brief summary of the approximative approaches used for reducing calculation time can be found in [Mekhnacha00]. In [Bessière03], one of these approximative methods is described in detail.

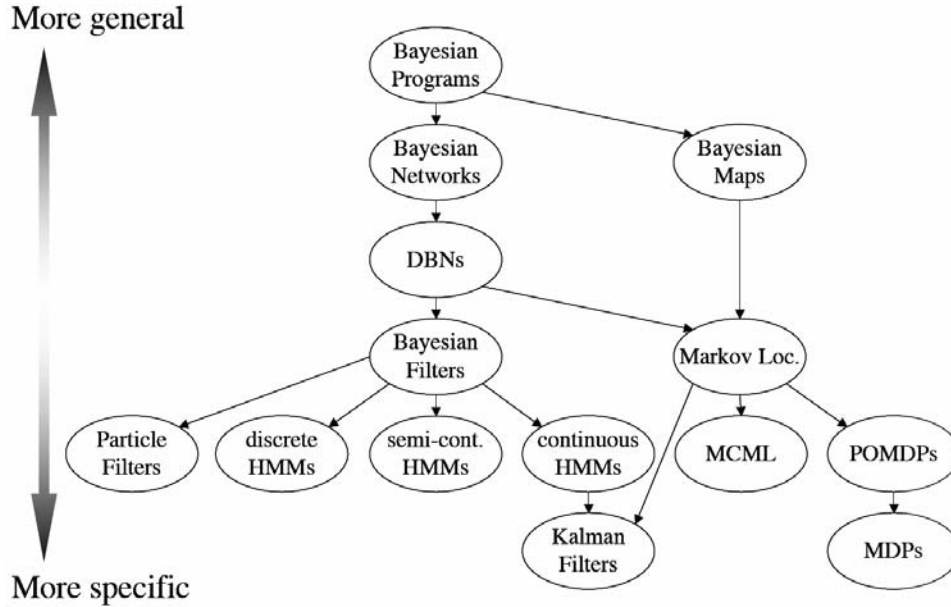


Figure 4.3: *Bayesian Programming and other probabilistic approaches (see [Diard03])*

C. Bayesian Programs and Other Probabilistic Approaches

Bayesian Programs have been shown to be a generalization of most of the other probabilistic approaches [Diard03], as shown in Figure 4.3. It means that all these probabilistic approaches may be reformulated using the Bayesian Programming formalism and thus easily compared with one another. For instance, Bayesian Networks correspond to a description where one and only one variable may appear to the left of each probability distribution appearing in the decomposition. This restriction enables optimized inference algorithms for certain class of questions.

A more detailed description can be found in Appendix A.

4.3.1.2 Bayesian Program for Localization

The Bayesian approach to localization with the fingerprints of places, presented here, is composed of two steps. The first step is the phase of supervised learning where the robot inspects several locations, denoted by Loc . From each location $loc \in Loc$ the robot extracts the fingerprint data, as explained in Section 3.4 and stores it along with the name of the location in a database, denoted by the symbol δ . The second step is the phase of application, when the robot localizes itself in the environment. To answer the question “Where am I?” the robot will extract the features composing the *fingerprint of a place* of its surroundings: the set of vertical edges VE , the set of color patches CP , and the set of corners C , and solve the question corresponding to probabilistic robot localization given as:

$$loc^* = \arg \max_{loc \in Loc} P(loc | C VE CP \pi \delta) \quad (4.2)$$

This means that if *fingerprints of places* are associated to each location, then the actual location of the robot may be recovered by comparing the features composing the *fingerprint of a place*: the set of vertical edges VE , the set of color patches CP , and the set of corners C with the database of known locations. The location loc^* which maximizes the probability measure is chosen. The preliminary knowledge is summed up by π . In the following it is shown how $P(loc | C VE CP \pi \delta)$ can be solved by the Bayesian Programming technique.

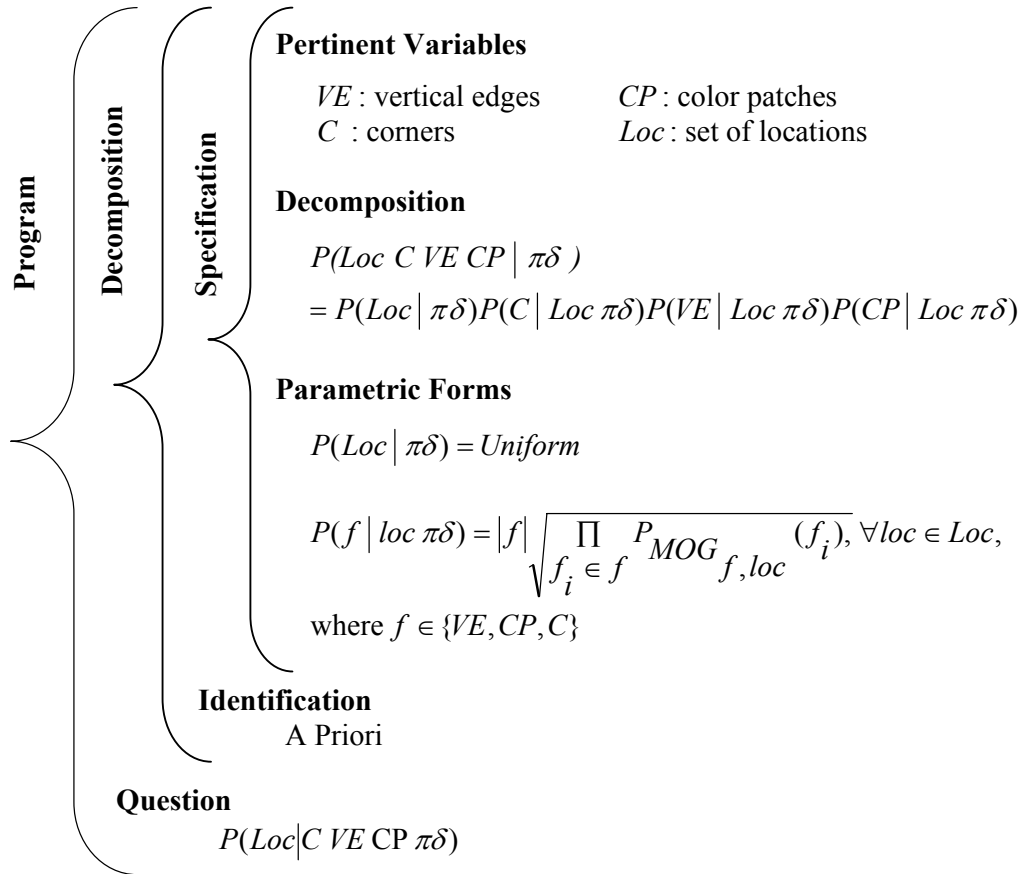


Figure 4.4: Bayesian Program for robot localization given a fingerprint based topological map

Figure 4.4 illustrates the Bayesian Program used for the Bayesian fingerprint matching. Several features are used in the fingerprints of places. These are denoted by: VE the set of vertical edges and CP the set of color patches extracted by the omnidirectional camera and C the set of corners (i.e. the line-segment extremities) extracted from the data given by the laser scanner. The variables VE , CP and C are independent of one

another, but they are location-dependent and it is these dependencies that give rise to the Bayesian Program formulation shown in Figure 4.4.

The decomposition (see Bayesian Program) described above involves three different kinds of probability distributions:

- Since no a priori information about locations is available, it is considered that each location is equally probable and consequently the probability of a location given all the prior knowledge is expressed as a uniform distribution.
- To determine the probability of one feature f , where $f \in \{VE, CP, C\}$, given the location and all the a priori knowledge, this probability is given as the likelihood of the new feature data f with respect to the distribution of the same feature as encountered at the given location during the learning phase. This distribution is calculated as a mixture of Gaussians (MOG) in angle space, optimizing the mixture parameters by making use of the Expectation Maximization (EM) algorithm. More details about these two concepts are described in the next three sub-sections.

The two equations from the Parametric Forms will solve the basic question described in the Bayesian Program (see Figure 4.4).

4.3.1.3 Mixture of Gaussians

Mixture of Gaussians (MOG) is a widely used approach when estimating the distribution of data. A MOG in the parameters θ is a probability density function, which results from combining k Gaussian probability density functions in a weighted sum:

$$P_{MOG}(\theta)(x) = \sum_{i=1}^k w_i P(x|\theta_i) \quad (4.3)$$

$$\theta_i = \{w_i, \mu_i, \sigma_i\} \quad (4.4)$$

where w_i is the weight, μ_i and σ_i the mean and the standard deviation of the i^{th} mixture component, which itself is a Gaussian probability density function given by the formula:

$$p(x|\theta_i) = N(x, \mu_i, \sigma_i) = \eta \exp\left(-\frac{(\mu_i - x)^2}{2\sigma_i^2}\right) \quad (4.5)$$

The normalization factor η turns the Gaussian function into a probability distribution function by guaranteeing that the integral over the function evaluates to 1:

$$\int P(x|\theta_i) = 1 \quad (4.6)$$

In angle space, η is the inverse of the integral from $-\pi$ to π over the un-normalized Gaussian function of:

$$\eta = \frac{1}{\text{erf}\left(\frac{\sqrt{2\pi}}{2\sigma_i}\right)\sqrt{2\pi}\sigma_i} \quad (4.7)$$

where $\text{erf}(x)$ is the error function

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \quad (4.8)$$

Since p_{MOG} is also a probability density function, the weights w_i must sum to 1, such that the integral over the distribution is 1:

$$\sum w_i = 1 \quad (4.9)$$

$$\int P_{MOG}(x|\theta_i) = 1 \quad (4.10)$$

The parameters of the complete MOG are then

$$\theta_{MOG} = \{\theta_1 \dots \theta_n\} = \{w_1 \dots w_n, \mu_1 \dots \mu_n, \sigma_1 \dots \sigma_n\} \quad (4.11)$$

The MOG is a compromise between the efficient but parametric models on one side, and the flexible but expensive non-parametric methods like histograms or kernel methods on the other side.

4.3.1.4 Expectation Maximization

Finding the optimal parameters θ_{MOG} of a Mixture of Gaussians (MOG) over a set of data points X is not trivial. A widely used approach to solve this problem is the Expectation Maximization (EM) algorithm [Bilmes97].

This algorithm starts with an initial estimate of the parameters θ and improves upon them iteratively. The algorithm proceeds in two steps:

E- step: Calculates the complete data likelihood given the known data X and the current parameters θ .

M- step: Calculates the new parameters θ^{ew} which maximize the joint probability $P(X, Y|\theta)$, where Y is the hidden data, which in our case is the knowledge about the probability that the i^{th} data point x_i was generated by the k^{th} mixture component.

The ‘improvement’ is defined in the sense that the log-likelihood of the data X increases with respect to the new parameters θ^{ew} . In the case of mixtures of Gaussians, it is

possible to derive the new parameters θ^{new} analytically. The resulting formulas merge the E- and M- step and are given by:

$$w_k^{new} = \frac{1}{N} \sum_{i=1}^N p(k|x_i, \theta) \quad (4.12)$$

$$\mu_k^{new} = \mu_k^{old} + \frac{\sum_{i=1}^N dist_{AS}(\mu_k^{old}, x_i) p(k|x_i, \theta)}{\sum_{i=1}^N p(k|x_i, \theta)} \quad (4.13)$$

$$\sigma_k^{new} = \sigma_k^{old} + \frac{\sum_{i=1}^N p(k|x_i, \theta) dist_{AS}(\mu_k^{old}, x_i)^2}{\sum_{i=1}^N p(k|x_i, \theta)} \quad (4.14)$$

where N is the number of data points: $N = |X|$ and $dist_{AS}(a, b)$ the distance function in angle space. It takes two angles a, b and returns the shortest way to go from a to b . The sign of the distance is positive if going clockwise, and negative if going counter clockwise. The iteration is typically terminated when the increase of the log-likelihood falls below some threshold value ε .

4.3.1.5 Example

This part shows a simple example of localization, when only the vertical edges are used and thus the probability $P(f = VE | loc \pi\delta)$ is illustrated through an example. A set of 13 occurrences of vertical edges is chosen and the MOG for it are calculated. A second set of data is generated, this time with 18 occurrences, and the probability

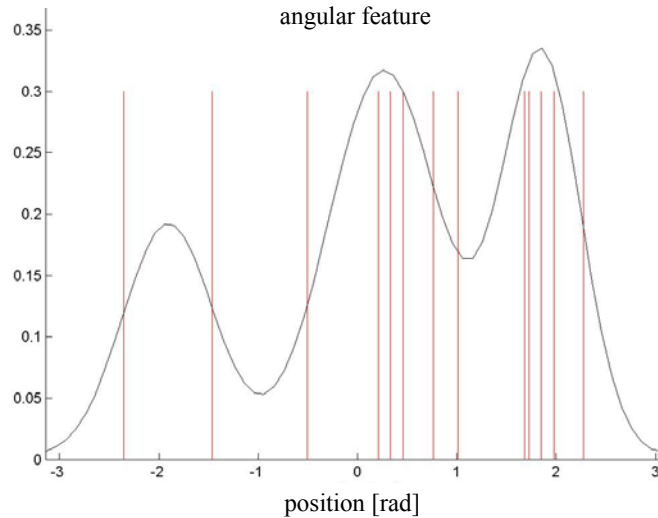


Figure 4.5: Evaluation of $P(f=VE | loc \pi\delta)$ for the original data set

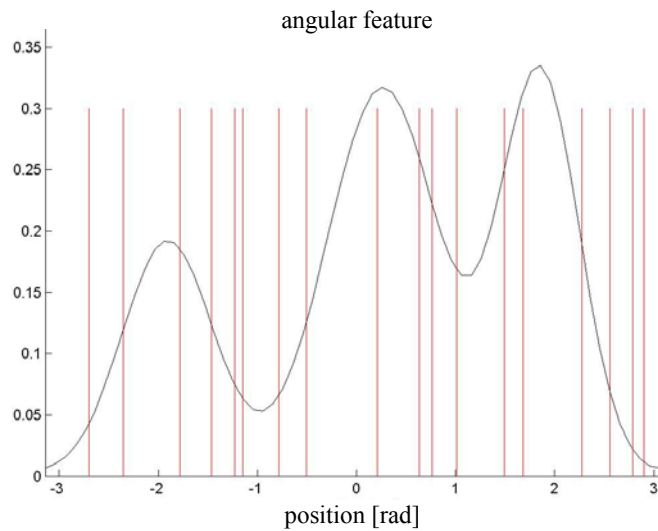


Figure 4.6: Evaluation of $P(f=VE | loc \pi\delta)$ for the some other data set, resulting in a smaller value above, since the MOG is not optimal for this data.

$P(f = VE | loc \pi\delta)$ for both datasets with the same MOG parameters (see Figure 4.5 and Figure 4.6) is evaluated. As expected, the resulting value is for the first data set significantly higher than for the second, since the parameters of the MOG were chosen to maximize the first set.

Note how flexible this method is with respect to the number of features per set: A MOG can be generated from a set of any number of features, and it can be evaluated later for samples of arbitrary length.

4.3.1.6 Experimental Results

This approach is firstly tested in simulation on a variety of synthetic office environments, where it is easier to run extensive experiments. These results are thereafter confirmed by running experiments in a real office environment with a robot (see Figure 4.8).

4.3.1.6.1 Simulation Experiments

For ten synthetic office environments, four observations of each feature $f \in \{VE, CP, C\}$ were randomly generated, such that the four generations were similar but not identical. The four observations of a feature in a certain place were disturbed by the addition of a small random distance noise, with a standard deviation of five degrees. After the generation of the dataset, a MOG of k components was calculated for the combination of the four observations of each place. With these ten MOGs the 40 observations were classified, by calculating the likelihood of the angular feature according to the MOG. For

a given observation, the classification is successful, if the highest value of the MOG corresponds to the correct place.

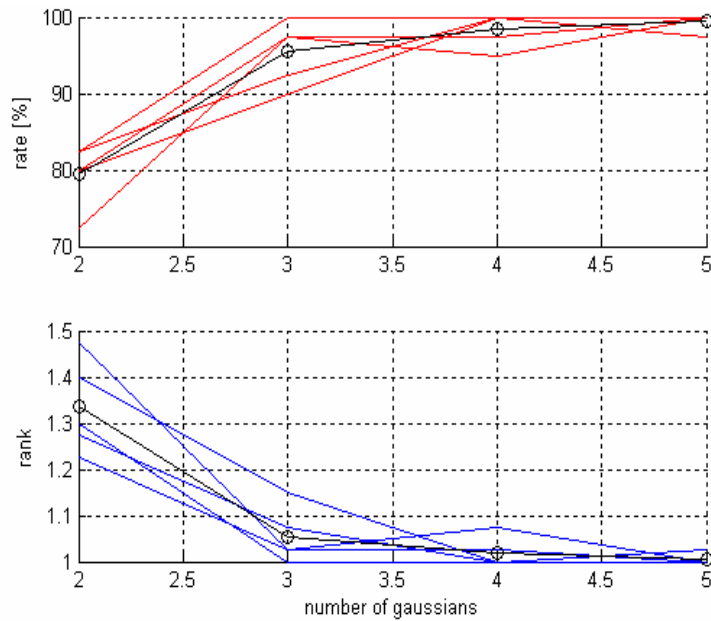
The simulation results for all the combination of k , with $k \in \{3, 4, 5\}$ and N (the number of occurrences of a feature f , in our case $f = VE$) with $N \in \{5, 10, 15, 20, 25, 30, 35\}$, is described in the following table (see Table 4.1).

TABLE 4.1: *The table shows the results of a simulation with different parameters for the number k MOG components and N , the number of observations of a feature.*

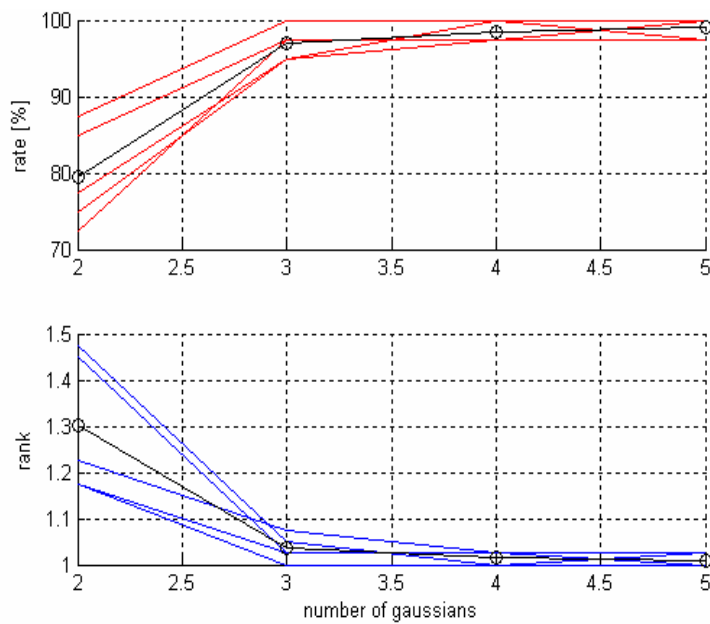
	N = 5	= 10	= 15	= 20	= 25	= 30	= 35
k = 3	55.5%	81.5%	93.0%	93.5%	96.0%	97.5%	99.5%
k = 4	66.5%	93.0%	98.0%	95.0%	98.0%	99.5%	99.0%
k = 5	80.0%	98.5%	98.0%	99.5%	100%	99.5%	100%

A graphical description of these results (i.e. for the vertical edge feature; for 25 and 30 occurrences of the same) for all the possible combination of k (number of mixture components) is illustrated in Figure 4.7(a) and (b) respectively. The graphs depict the rate and the rank obtained. The rate signifies the percentage of successful classifications as a function of the different number of mixture components. In the case of a correct match, the rank is set to 1. If the evaluation with other MOG's parameters than the expected one gave a higher value, the rank of the sample is 2, 3, 4, ..., etc, depending on how many values were better than the expected one.

Figure 4.7(a) and (b) shows that for $k_{MOG} = 3$ very good results are obtained with a percentage of successful classification in between 81.5% and 99.5%. By increasing the number of MOG components the results are improved. For $k_{MOG} = 5$, the results give a percentage comprise in between 98.5% and 100%. It is clearly visible, that with more feature occurrences per sample, both rank and rate yield better results.



(a)



(b)

Figure 4.7: (a) Results of the experiments with $N_{VE} = 25$ for different numbers (k) of mixture of Gaussians (MOG) components. $k_{MOG} = 3$ gives already very good results. The best values are obtained for $k_{MOG} = 5$. (b) Results of the experiments with $N_{VE} = 30$ for different numbers (k) of mixture of Gaussians (MOG) components. $k_{MOG} = 3$ gives already very good results. The best values are obtained for $k_{MOG} = 5$.

For the $f \in \{CP, C\}$ the process was similar to the shown one. The simulation results are significant and after the combination of all elements it was noted that:

- a. With the increase of the MOG components one sees an improvement of the results.
- b. With the increase of the number of observations N of a feature one sees an improvement of the results

4.3.1.6.2 Experiments with the robots

In order to verify the simulation results, these experiments were repeated on a real robot. The approach has been tested in a $50 \times 25 \text{ m}^2$ portion of our institute shown in Figure 4.8.

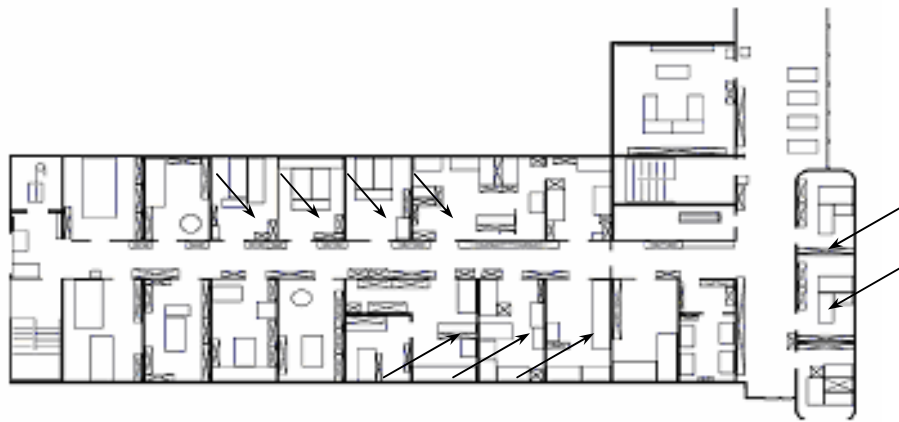


Figure 4.8: *The test environment. The arrows indicate the rooms in which the experimentation has been done. The tenth room is not represented in the image.*

The test setup was the following: The robot extracted the three features (i.e. vertical edges, color patches and corners) in ten offices ten times. Eight times it was placed on a circle of 40 cm to 70 cm of radius (so that the places will be slightly different), yielding the training data, and two times inside the same circle, yielding the test data. The mean number of feature occurrences in each measurement is summarized in the following table:

TABLE 4.2: *The table shows the mean number of occurrences of instances of the same feature type during the measurements.*

	mean occurrences
vertical edge	5.6
corner	5.6
color patch	2.7

During all measurements, the orientation of the robot was approximately the same. This simplification could be omitted by letting the robot estimate its orientation by considering all rotations of the fingerprint string. In order to complete the training, for each location and each feature type the mixture of Gaussian was calculated. In the application phase, the classification of the 20 test samples was used to answer the following questions:

1. What is the significance of each feature type using the MOG approach?
2. How does the combination of several feature types improve the MOG approach?

Since the number of occurrences of the color patch feature was too small to give significant results, it was omitted for the MOG calculations. k_{MOG} signifies the number of mixture components used for the mixture (Table 4.3).

TABLE 4.3: *Results using single features only (vertical edge, corners) and the combination of these features.*

	$k_{MOG} = 3$		$k_{MOG} = 4$		$k_{MOG} = 5$	
vertical edge	42.1%	2.58	44.1%	2.47	55.6%	1.82
corners	55.6%	1.65	51.1%	1.71	47.1%	1.71
ve & c	58.8%	2.00	66.7%	1.82	66.7%	1.67

The success of a room classification is defined as the detection of the highest probability. Successful classification was produced 66.7% of the time, as observed from Table 4.3. The method seems promising, but the percentage of successful matches (i.e. 66.7%) can't be considered sufficient for robot localization.

4.3.2 Fingerprint Matching with Global Alignment Algorithm

Since the results of the previously presented method were not very satisfying (only 66.7% of successful classification), another alternative approach is proposed here.

Fingerprints of places are represented as a sequence of characters and therefore the localization is reduced to a string-matching problem. String matching is not a trivial problem. Usually strings do not match exactly because the robot may not be exactly located on a map point and/or some changes in the environment or perceptual errors may have occurred. Many string-matching algorithms can be found in the literature but they generally require the strings to have the same length. Some of them allow a level of mismatch, such as the "k-mismatch" matching algorithms and string matching with k

differences [Aho90], [Baeza-Yates99]. The first allows matches where up to k characters in the pattern do not match the text and the second requires that the pattern have an edit distance[†] of the text of k or less elements. One of the main problems of the above methods is that they do not consider the nature of the features and the specific nature of the mismatch that occurred.

The likelihood of specific types of mismatch errors has to be taken into account. For instance confusing a red patch with a blue patch is more egregious than confusing the red patch with a yellow patch. The standard algorithms are quite sensitive to insertion and deletion errors, which cause the string lengths to vary significantly. The method adopted previously in the fingerprint approach for sequence matching is the minimum energy algorithm usually used in stereovision [Lamon03], [Kanade85].

4.3.2.1 Localization with Global Alignment (GA)

The global alignment algorithm used in this work is an adapted version of the basic algorithm used in the context of DNA string matching [Needleman70]. It uses a cost-function as a means of estimating the cost of aligning the characters of the two strings under consideration. String matching is performed by minimizing this cost function. The following section attempts to first draw out the concepts (of alignment / computing cost/ matching using this algorithm) before going into more precise technical details about the same. Consider the example shown in Figure 4.9. This example uses two sample strings and computes the cost of aligning them.

string1 := « abcd »
string2 := « bbc »

$$\text{cost}(x, y) := \begin{cases} x = \varepsilon | y = \varepsilon & 0.6 \\ x = y & 0.0 \\ \text{else} & 1.0 \end{cases}$$

The cost of alignment $\begin{bmatrix} a b c d \\ b b c \varepsilon \end{bmatrix}$ is calculated as:

$$= \text{cost}('a', 'b') + \text{cost}('b', 'b') + \text{cost}('c', 'c') + \text{cost}('d', '\varepsilon')$$

$$1 \quad + \quad 0 \quad + \quad 0 \quad + \quad 0.6 \quad = \quad 1.6$$

Figure 4.9: An example of calculating the cost between two strings.

More formally, five elements, which constitute the elements used in the global alignment algorithm (see Figure 4.10) can be distinguished. The first element is an alphabet denoted by A , typically a set of letters, which is not empty. The second element corresponds to the two strings which are to be aligned: the first is composed of

[†] The *edit distance* of two strings, s_1 and s_2 , is defined as the *minimum* number of *point mutations* required to change s_1 into s_2 , where a point mutation is one of: change a letter, insert a letter, or delete a letter

m , the second of n letters of the alphabet. The occlusion symbol is used to represent a space inserted into the string. The cost function gives the cost for the match between two symbols of the alphabet, including the occlusion symbol. Finally, the cost matrix is used to keep the minimal cost of a match between the first i letters of the first string with the first j letters of the second string, keeping this value in the element (i,j) of matrix V .

Alphabet	$A, A \neq \{ \}$
Strings	$S1 \in A^m, S2 \in A^n, m, n \in \mathbb{N}$
Occlusion symbol	$\varepsilon, \varepsilon \notin A$
Cost function	$f_{cost} : a \in A \cup \varepsilon, b \in A \cup \varepsilon \rightarrow \mathfrak{R}$

Figure 4.10: *The main elements of the Global Alignment algorithm.*

The values of the cost function $f_{cost}(a, b)$, are calculated experimentally as a function of the similarity between characters a and b . In other words the more similar the characters are, the lower is the penalty for a mismatch. The last step involves the calculation of the elements of the cost matrix. This is constructed using a technique named "dynamic programming". Dynamic programming methods ensure the optimal global alignment by exploring all possible alignments and choosing the best. Initially, the edges of the matrix are initialized with the cumulative cost of occlusions. This reflects the fact that the number of letters that must be jumped in one or the other string in order to obtain the best solution is not known a priori.

The base conditions of the algorithm are:

$$\bullet \quad V(0, j) = \sum_{1 \leq k \leq j} f_{cost}(\varepsilon, S2(k)) \quad (4.15)$$

$$\bullet \quad V(i, 0) = \sum_{1 \leq k \leq i} f_{cost}(S1(k), \varepsilon) \quad (4.16)$$

For i and j both strictly positive, the recurrence relation is:

$$V(i, j) = \min \begin{cases} V(i-1, j-1) + f_{cost}(S1(i), S2(j)) \\ V(i-1, j) + f_{cost}(S1(i), \varepsilon) \\ V(i, j-1) + f_{cost}(\varepsilon, S2(j)) \end{cases} \quad (4.17)$$

The three cases that can be distinguished from the above relation are:

- **Aligning $S1(i)$ with $S2(j)$:** The score in this case is the score of the operation $f_{cost}(S1(i), S2(j))$ plus the score of aligning $i-1$ elements of $S1$ with $j-1$ elements of $S2$, namely, $V(i-1, j-1) + f_{cost}(S1(i), S2(j))$
- **Aligning $S1(i)$ with an occlusion symbol in string $S2$:** The score in this case is the score of the operation $f_{cost}(S1(i), \epsilon)$ plus the score of aligning the previous $i-1$ elements of $S1$ with j elements of $S2$ (Since the occlusion is not an original character of $S2$), $V(i-1, j) + f_{cost}(S1(i), \epsilon)$
- **Aligning $S2(j)$ with an occlusion symbol in string $S1$:** Similar to the previous case, the score will be $V(i, j-1) + f_{cost}(\epsilon, S2(j))$.

If strings $S1$ and $S2$ are of length n and respectively m , then the value of their optimal alignment with the global alignment technique is the value of the cell (n, m) .

4.3.2.2 Localization with Global Alignment with Uncertainty

The global alignment with uncertainty changes only the cost function described earlier. The cost function is adapted in order to take into account the corresponding uncertainty of features. The goal of adding the uncertainty in the string matching algorithm is to improve the distinctiveness of places. Next, a small example of global alignment algorithm with uncertainty will show the improvement in the matching (see Figure 4.11).

The example depicted in Figure 4.11 shows the improvement obtained by the new fingerprint matching with uncertainty algorithm. Even if the two fingerprints of places from the map are similar (i.e. string1 and string2), the uncertainty of the features will determine the map fingerprint of place that best matches the observed fingerprint of place (i.e. stringObs).

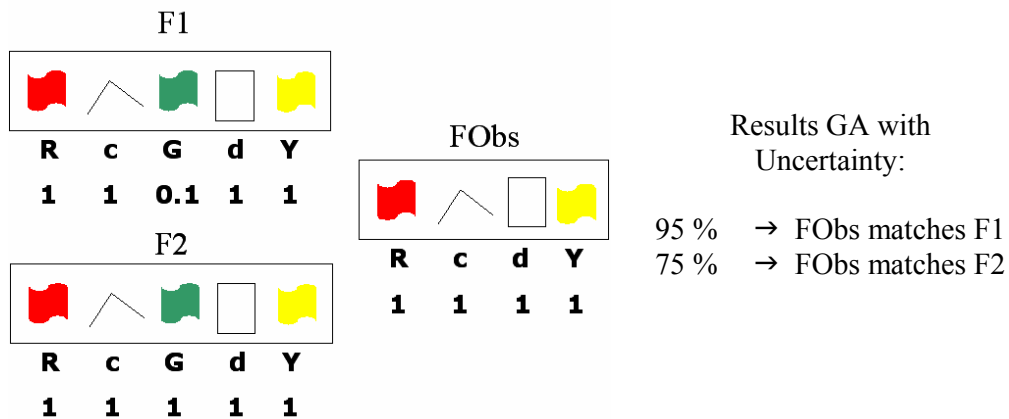


Figure 4.11: An example of the Global Alignment algorithm with the uncertainty: $F1$ and $F2$ are the two fingerprints of places stored in the database of known locations. $FObs$ is the observed fingerprint of place.

4.3.2.3 Indoor Experimental Results

The test setup was the following: The robot extracted the three features (i.e. vertical edges, colors, and corners) in seven offices (see Figure 4.8) at 11 different places. For the new matching approach, the uncertainties of different features have been modeled. One fingerprint of place per room has been included in a database as reference (map initialization) for the localization approach. The other 70 fingerprints of places have been matched to the database for testing the localization.

During all measurements, the orientation of the robot was approximately the same. This simplification could be omitted by letting the robot estimate its orientation by considering all rotations of the fingerprint string.

For a given observation (fingerprint of place) a match is successful if the best match with the database corresponds to the correct room. Table 4.4 illustrates the percentage of successful matching and the mean rank for three string matching algorithms: minimum energy, global alignment and global alignment with uncertainty. The rank calculates the position of the correct room, with respect to the others, in the classification (e.g. if the match is successful than the rank is 1, if the correct room is detected with the second highest probability the rank is fixed at 2, etc.).

4.3.2.3.1 Results with Global Alignment

In order to explain and validate the choice of the global alignment algorithm, a comparison between an adaptation of the minimum energy algorithm and the global alignment methodology has been performed. The minimum energy algorithm is also a dynamic-programming based algorithm and it is usually used in stereo-vision. An adaptation of this algorithm has been developed in [Lamon01]. Both algorithms (i.e. global alignment and minimum energy) have been tested in the same environment and under the same conditions. The test setup has been described above. In Table 4.4, one can see the improvement obtained using the global alignment technique in comparison to the minimum energy algorithm. Both the percentage of good classifications and the mean rank are improved. The classification accuracy with the global alignment algorithm is of 75 % and the mean rank of 1.32.

4.3.2.3.2 Results with Global Alignment with Uncertainty

The global alignment with uncertainty is just an adaptation of the global alignment algorithm, wherein, the uncertainty of features was added.

TABLE 4.4: *Classification using string matching, comparing minimum energy, global alignment and global alignment with uncertainty algorithms.*

	right classifications	mean rank
minimum energy	58.82%	1.85
global alignment	75%	1.32
global alignment with uncertainty	83.82%	1.23

In Table 4.4, one can see the amelioration obtained, from using global alignment with uncertainty instead of the global alignment or minimum energy algorithm. The results with global alignment with uncertainty algorithm have 83.82% of successful matches, which corresponds to a clear improvement of 8.82% with respect to the standard global alignment. However, false-classified rooms delivering high probability (second or third highest probability), which are typical results in the experiments, entail important information, which can be used in combination with a localization approach such as a *Partial Observable Markov Decision Process* (POMDP). More details are given in Chapter 6.

4.3.2.4 Outdoor Experimental Results

For outdoor experiments, the test setup was the following: data was acquired from both the lasers and the omnidirectional camera, using the system mounted on the “SMART” vehicle. The approach has been tested in a part of the EPFL campus (highly structured environment), shown in Figure 4.12, on a 1.65 km of trajectory.



Figure 4.12: *The outdoor test environment (a part of the EPFL campus) with the trajectory of 1.65 km long traveled by the “SMART” vehicle.*

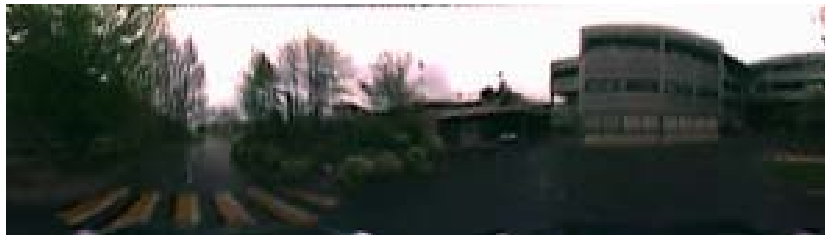
The reference map (the database) contains 15 different locations within the campus. The test set for localization is composed of 10 different fingerprints of places for each location. These test fingerprints of places have been extracted in a zone of 4 meters long and 2 meters large with respect to the reference location.

The vehicle orientation was not fixed a priori. During all measurements, the vehicle estimated its orientation by considering all rotations of the fingerprint string. Figure 4.13 depicts this phenomenon. Figure 4.13(a) represents a part of the environment within the EPFL campus, at a crossing. Figure 4.13(b) represents the same scene as the one illustrated in Figure 4.13(a), after turning left at the crossing and traveling 4 meters further. The scenes change less in outdoor environments than in indoor environments. In indoor environments, traveling 4 meters results the fingerprints of places change drastically. Contrary to this, in outdoor environments the features change less and thus the fingerprints of places change less. In Figure 4.13 the fingerprints of places appear to be totally different at a first glance, but they are almost

the same with a shift of 110° degrees. This problem is easily solved by our algorithm that takes into account all possible rotations.



(a)



(b)

Figure 4.13: *Fingerprint-matching is orientation independent: (a) Illustrates one location in an outdoor environment - at a crossing, just before turning. The corresponding fingerprint of place obtained is: `ncfccccccccffnccvncvccccncncccc` (b) The same scene after turning left and traversing 4 meters. The fingerprint of place obtained from the scene illustrated in (b) is: `nccccncfcfccccccffnccvncvcccc`.*

For a given observation (fingerprint of a place) a match is successful if the best match with the database corresponds to the location in the outdoor environment. Table 4.5 illustrates the percentage of successful matching and the mean rank for the two string matching algorithms described above: global alignment and global alignment with uncertainty. The rank calculates the position of the correct outdoor location, with respect to the others, in the classification (e.g. if the match is successful than the rank is 1, if the correct outdoor location is detected with the second highest probability the rank is fixed at 2, etc.).

TABLE 4.5: *Classification using string matching, comparing global alignment and global alignment with uncertainty algorithms, in outdoor environments.*

	right classifications	Mean rank
global alignment	72.66%	1.31
global alignment with uncertainty	80.67%	1.24

The results are as good as those obtained with the same algorithms in indoor environments. The “SMART“ vehicle was correctly localized, with the global alignment algorithm, with a percentage of 72.66%. The mean rank obtained is 1.31. As the table clearly shows, even in the case of outdoor environments, the global alignment with uncertainty algorithm for the fingerprint-based localization outperformed the basic

global alignment algorithm. The percentage of correct classification was 80.67%, with a mean rank of 1.24. The mobile vehicle was correctly localized most of the time. In those cases when the localization was erred, the vehicle was classified with the 2nd or the 3rd highest probability, which still gives significant information.

4.3.3 Fusion of Bayesian Programming and Global Alignment

In Section 4.3.1.6.1, it was shown through simulation that if sufficient information (i.e. the number of occurrences of the landmarks) was available, the robot was able to recognize locations without any difficulty. The real world experiments performed in Section 4.3.1.6.2 took into account only the vertical edges and the corners (i.e. extremities of the segments). In order to achieve results as good as the ones shown in simulation, all the information contained in the fingerprint of a place should be used. The idea of the new approach follows the fusing philosophy, and hence the use of redundant information. A drawback of this approach is that the resulting scheme after the fusion of the Bayesian Program and the global alignment algorithm is more expensive in terms of computational load (i.e. resources and computational time required by the algorithm). The new approach that combines the first two methods described in Section 4.3.1 and Section 4.3.2.1 respectively is illustrated in Figure 4.11.

4.3.3.1 Bayesian Program

The Bayesian Program depicted here is the same with the one illustrated in Figure 4.11, with some additions. One more variable has been added – Fp , which represents the fingerprint of a place.

Although the fingerprint string Fp , constructed over all the features (see Section 3.4), adds some redundancy to the system, it simultaneously introduces valuable information about the relative order of the features. This will serve to improve the results. It is assumed that the variables VE , CP , C and Fp are independent from one another. The introduction of a new variable in the Bayesian Program changes the decomposition and consequently a new parametric form appears. The decomposition is described as follows:

$$\begin{aligned} P(Loc\ C\ VE\ CPFp\ | \ \pi\ \delta) \\ = P(Loc\ | \ \pi\delta)P(C\ | \ Loc\ \pi\delta)P(VE\ | \ Loc\ \pi\delta)P(CP\ | \ Loc\ \pi\delta)P(Fp\ | \ Loc\ \pi\delta) \end{aligned} \quad (4.18)$$

Thus, to calculate the probability of the fingerprint sequence given the location and all prior knowledge $P(Fp\ | \ loc\ \pi\delta)$, the global alignment algorithm (see Section 4.3.2.1) is used. Let $GlobalAlignment(Fp, fp_{loc})$ be a function yielding the minimal cost of the global alignment algorithm for two fingerprint strings. The new Bayesian Program describing the fusion between the previous Bayesian Program approach for fingerprint matching (see Figure 4.4) and the global alignment algorithm, is depicted in Figure 4.14.

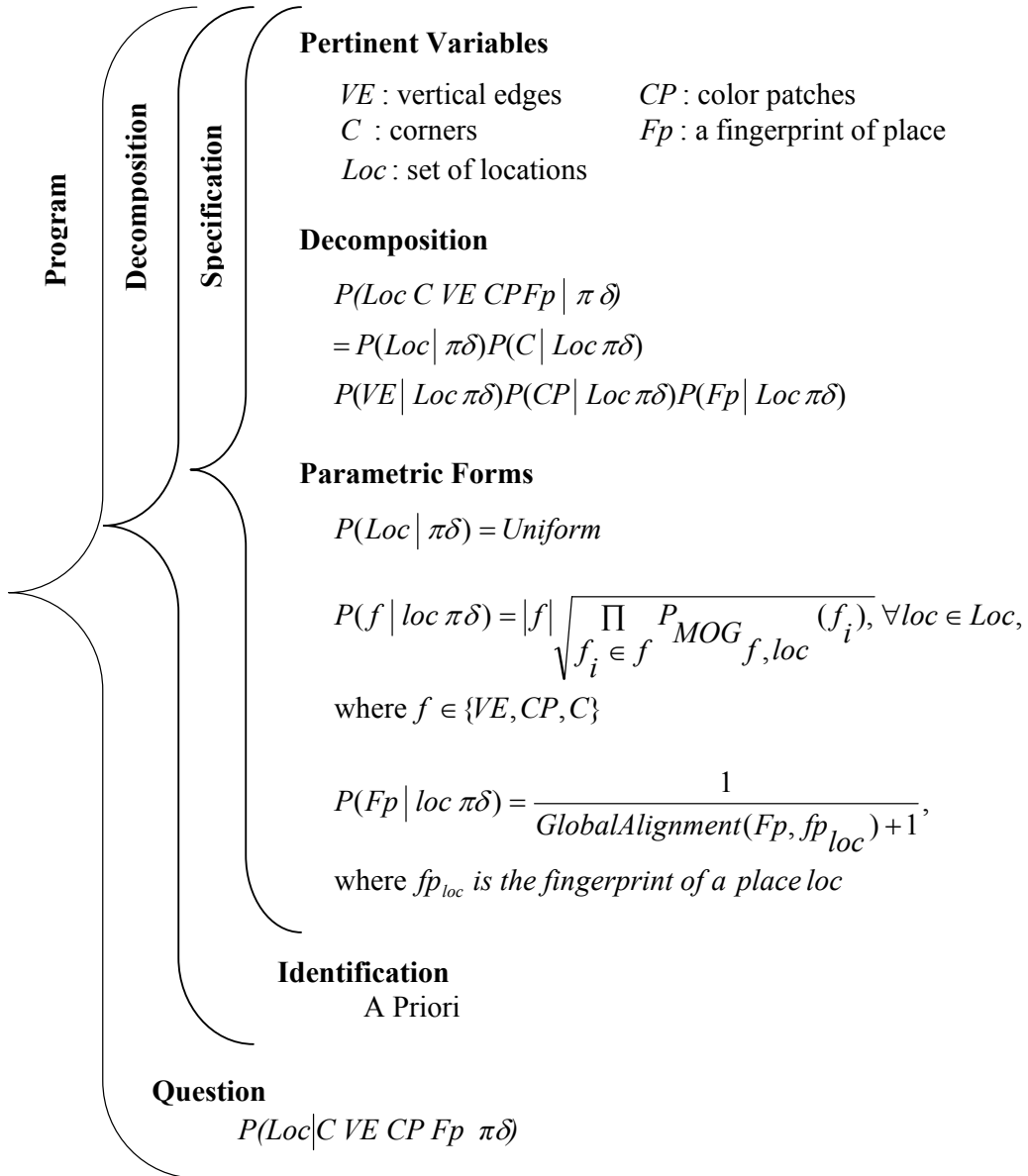


Figure 4.14: The fusion of Bayesian Programming and Global Alignment algorithm written in BP formalism

4.3.3.2 Experimental Results

The same test setup as the one described in Section 4.3.1.2.6 is used. By making the fusion between the Bayesian Program approach and the Global Alignment algorithm, we want to know “What is the overall classification capability of the system using both string matching and MOG approach combined in a Bayesian Program?.” Because the number of color patches is too small to give significant results, they were omitted for the MOG calculations. Nevertheless they were used for the fingerprint strings.

TABLE 4.6: *Results of the classification using all the features (vertical edges and corners) and the fingerprint string matching.*

	$k_{\text{MOG}} = 3$		$k_{\text{MOG}} = 4$		$k_{\text{MOG}} = 5$	
ve & c & Fps	61.1%	1.78	70.6%	1.58	82.4%	1.23

The success of room classification is defined as the detection of the correct room with the highest probability. The results have given a percentage of successful matches of 82.4%, when the number of MOG components $k_{\text{MOG}} = 5$ (see Table 4.6).

4.4 Discussion and Comparison of the Place Recognition Experiments

In the four sets of experiments, the BIBA robot has acquired the ability to recognize priorly visited places within the environment. The indoor experiments were similar in that they all used fingerprints of places (i.e. circular list of features) and they were all conducted under the same conditions. However, they differed in the choice of the algorithm used to perform the topological localization.

The first set of experiments used Bayesian Programming (i.e. a generic formalism) combined with mixture of Gaussians (MOG) and EM algorithm. The results showed that the number of Gaussians to be used depends on the number of occurrences of the features. Only vertical edges and corners (i.e. extremities of line-segments) have been used. By increasing the number of MOG components, an improvement of the results was noticed (e.g. the results obtained by using the vertical edges and corners as features, with $k_{\text{MOG}} = 3$, a percentage of 58.8% of right classification was detected; with $k_{\text{MOG}} = 4$, a percentage of 66.7% correct classification was obtained and finally with $k_{\text{MOG}} = 5$, the same percentage as previously (i.e. of 66.7%) was found with a better mean rank) (see Table 4.3).

The second set of experiments used a new approach, named the global alignment (GA) algorithm. All the features present in the fingerprint of a place were taken into account. A string-matching process was used to distinguish between the locations and hence localize the robot.

The third set of experiments used an adapted version of the global alignment algorithm, wherein the uncertainty of the features was also incorporated. This uncertainty is an important element that has to be taken into account when performing localization. Since the features extracted with the omnidirectional camera are less reliable than those extracted with the laser scanner, when the fingerprint matching is done, a bigger

importance factor (i.e. depending on the uncertainty of feature extraction) was given to laser scanner features than to omnidirectional camera features. The performance (percentage of correct classification) of the adapted global alignment algorithm was found to be better than the basic global alignment algorithm by as much as 8.82% (see Table 4.7). The outdoor experiments were performed using the "SMART" vehicle, in 1.65 km portion of the EPFL campus. Both the global alignment and the global alignment with uncertainty algorithms were used to localize the vehicle in these experiments. Their performance was compared. Global alignment algorithm produced correct classifications in 72.66% of the test cases. The second algorithm incorporated the uncertainty in the features themselves and produced significantly better results. It correctly classified 80.67% of the test cases thereby improving over the basic global alignment algorithm approach by about 8%.

The last set of experiments combined aspects of both the previous approaches: it used a Bayesian Program, like in the first experiment, to which it added the global alignment algorithm, as in the second experiment. A clear amelioration with respect to the results obtained with both methods alone was noticed. The percentage of successful matches by using the vertical edges, the extremities for MOG and the fingerprint of place containing all the ordered features was of 82.4%. The rest of the mismatched locations were found on the second or third positions.

All these results are summarized in the following table (see Table 4.7):

TABLE 4.7: *Comparison of the results obtained with the different fingerprint-matching algorithms.*

	right classifications	mean rank	observations
minimum energy	58.82%	1.85	-
BP(ve & ex)	66.7%	1.67	$k_{MOG} = 5$
global alignment	75%	1.32	-
BP(ve & ex & Fp)	82.40%	1.23	$k_{MOG} = 5$
global alignment with uncertainty	83.82%	1.23	-

Thus, the adapted global alignment algorithm (with uncertainties of features) was deemed to be the most appropriate method for the rest of this thesis, owing to its simplicity, robustness and yet high degree of accuracy. With this in mind, future usages of the term "fingerprint-matching" are meant to imply that the underlying method being used is the adapted global alignment algorithm (with uncertainties of features incorporated).

4.5 Summary

This chapter presented different algorithms for topological localization (i.e. place recognition) and showed through its results how mobile robots recognize locations they have visited before. In the place recognition experiments, the robot recognized the different locations by looking at the circular ordered sequence of features (i.e. the fingerprint of a place) and by comparing it with those of the known locations. Four algorithms for fingerprint-matching were developed: two probabilistic approaches (i.e. the Bayesian Program and the fusion between the Bayesian Program and the global

alignment algorithm) and two other methods, dynamic programming based. A discussion of the different algorithms has been concluded with a comparison of the results obtained using each of them. Even if false-classified rooms delivered high probabilities (second or third highest probability) this can entail important information if used in combination with a localization approach such as a *Partial Observable Markov Decision Process* (POMDP) (see Chapter 6).

5

Indoor Exploration Tools

5.1 Introduction

Autonomous robot map generation warrants suitable exploration strategies so as to enable the robot to discover its environment. In order to autonomously acquire and create maps, robots have to explore their environment. This chapter describes the different behaviors and methods used by our mobile platform to explore indoor environments: wall following, mid-line following, center of free space of a room, door detection, and environment structure identification.

5.2 Wall Following

Wall following behavior is an important behavior for mobile robot navigation. Many methods have been investigated, developed and discussed in literature for wall following. The methods described in [Remolina97] and [vanTurenout92] are just two examples from among the many existing ones.

The technique used to follow walls adopted in this work is described in the block diagram depicted in Figure 5.1. Following a wall consists of finding a wall in the environment and moving along-side it (i.e. moving parallel to the detected wall while maintaining a safe distance from it). Thus, in order to be able to track a wall, several processing steps are involved.

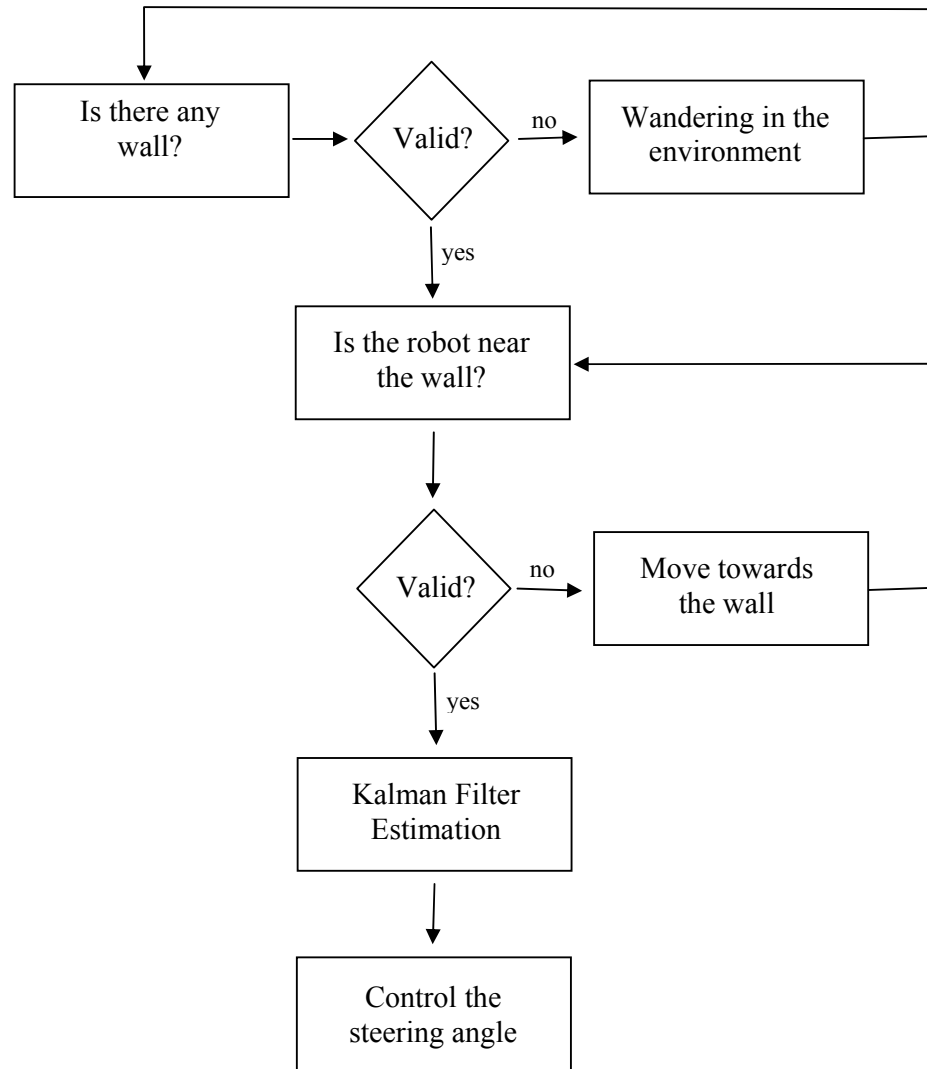


Figure 5.1: *Wall Following Diagram.*

5.2.1 Wall Detection

The first part is the detection of the wall to follow. In this work, it is assumed that a wall exists if the length of the line segment characterizing the wall is greater than 1 m.

In order to find a wall to follow, several steps are required. Firstly, the Douglas-Peucker algorithm (described in Section 3.2.1.2) is used to perform the segmentation of the data given by the two laser scanners of the robot, so as to find all the line-segments. From the set of all line-segments detected, the longest line-segment is selected. Since the hallways are typically made up of doorways and walls, they consist of several collinear line segments. In order to identify the longest wall to follow, all the collinear line-segments are connected.

If no wall is detected by the robot, a wandering behavior in its environment is adopted until one is found. Wandering is the simplest reactive behavior that an autonomous

mobile robot can have. The robot doesn't follow a pre-defined trajectory. The behavior relies on the move-straight-forward action. If an obstacle is detected, the robot avoids it and continues moving in a straight line until a wall is detected.

Once a wall is detected, a control law is used to control the steering angle. For example, if the distance from the robot to the wall is greater or smaller than a predefined threshold, the robot would change its steering angle so that this distance is close to the predefined threshold. This is described in the next section.

5.2.2 Control Law

In this part, the wall following control law that maintains the robot at a predefined distance from the wall while moving is described. The indoor robot used in this work, the BIBA robot (see Figure 1.2(a)), has a differential drive kinematics (see Figure 2.1).

In [Remolina97] and [vanTurenout92], a control law for boundary following, that specifies the value of the controlled variable ω (the robot's steering angle), is given as:

$$\omega = \frac{1}{v}[-k_{\theta}v\theta - k_e e] \quad (5.1)$$

where e and θ are the observed variables. The difference between the observed and expected (predefined) distances of separation between the robot and the object (in this case, a wall) is represented by e and the orientation of the robot with respect to the boundary is designed by θ . The parameter v indicates the forward velocity of the robot. This control law is similar to a PD controller and is depicted in Figure 5.2.

The system is critically damped if:

$$k_{\theta} = \sqrt{4k_e}. \quad (5.2)$$

This controller is illustrated below:

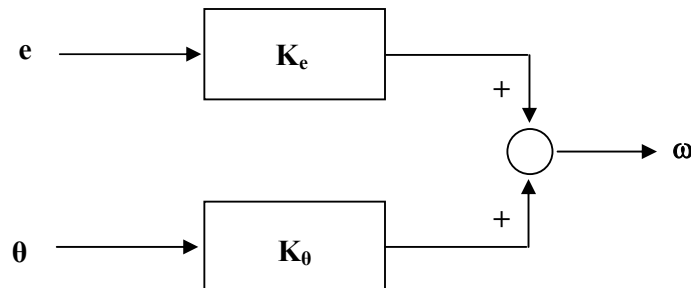


Figure 5.2: Control law for wall-following behavior, where

$$K_e = -\frac{k_e}{v} \text{ and } K_{\theta} = -k_{\theta}.$$

In order to estimate the values of the two variables e and θ , a Kalman Filter is used.

5.2.3 Kalman Filter (KF)

A Kalman filter [Kalman60] is employed to estimate the state of the system represented by $X = [e \ \theta]^T$, where e and θ are defined as above (see Figure 5.3). The odometry, as measured by the encoders (proprioceptive sensors) is used to perform the “prediction step“. The predicted state is corrected (“update step“) using the state estimate from the laser range data (exteroceptive sensor).

- **Prediction Step**

As mentioned earlier, the state of the system is represented by $X = [e \ \theta]^T$. This vector comprises of the quantities that are required to be estimated. In order to model the odometry error (see more details Section 2.2) of such a system, the error in the displacement of each wheel (left and right) is calculated separately, and the uncertain input is expressed as:

$$u(k+1) = [\delta\rho_R, \delta\rho_L]^T, \quad (5.3)$$

where $\delta\rho_R$ and $\delta\rho_L$ are the displacements of the right and left wheel respectively.

The corresponding diagonal input covariance matrix is given as:

$$U(k+1) = \begin{bmatrix} k_R |\delta\rho_R| & 0 \\ 0 & k_L |\delta\rho_L| \end{bmatrix}, \quad (5.4)$$

where k_R and k_L are two constants (with unit meter) representing the non-systematic parameters of the motor drive and the wheel floor interaction.

The Gaussian error model for the odometry is given by:

$$N(u^e(k+1), U(k+1)), \quad (5.5)$$

where $u^e(k+1)$ is the mean odometry.

The “*prediction step*“ of the state estimation process is given by the following equations:

$$x(k+1|k) = f(x(k|k), u(k+1)) \quad (5.6)$$

and

$$P(k+1|k) = \nabla f_x P(k|k) \nabla f_x^T + \nabla f_u U(k+1) \nabla f_u^T, \quad (5.7)$$

where P is the covariance matrix of the estimated state, and ∇f_x and ∇f_u are the Jacobians taken with respect to the uncertain inputs $x(k+1|k)$ and $u(k+1)$.

Equation (5.6) may be written as follows:

$$e_{k+1} = e_k + \delta\rho_k \sin\left(\theta_k + \frac{\delta\theta_k}{2}\right) \quad (5.8)$$

$$\theta_{k+1} = \theta_k + \delta\theta_k \quad (5.9)$$

where $\delta\rho_k = \frac{\delta\rho_R + \delta\rho_L}{2}$, $\delta\theta_k = \frac{\delta\rho_R - \delta\rho_L}{d}$, and d is the distance between the left and the right wheel.

The Jacobians of the associated uncertainty are given by the following expressions:

$$\nabla f_x = \begin{bmatrix} 1 & -\delta\rho_k \cos\left(\theta_k + \frac{\delta\theta_k}{2}\right) \\ 0 & 1 \end{bmatrix} \quad (5.10)$$

and

$$\nabla f_u = \begin{bmatrix} -\frac{1}{2} \sin \alpha_k - \frac{\delta\rho_k}{2d} \cos \alpha_k & \frac{1}{2} \sin \alpha_k - \frac{\delta\rho_k}{2d} \cos \alpha_k \\ \frac{1}{d} & -\frac{1}{d} \end{bmatrix} \quad (5.11)$$

where $\alpha_k = \theta_k + \frac{\delta\theta_k}{2}$.

- **Update Step**

The “*update step*” uses the lines extracted from the data of the laser scanner (see Section 3.2.1 for the line-extraction algorithm) to estimate the state of the system. This estimate is used to update the predicted estimate and thus arrive at a posterior state estimate of the system as shown below.

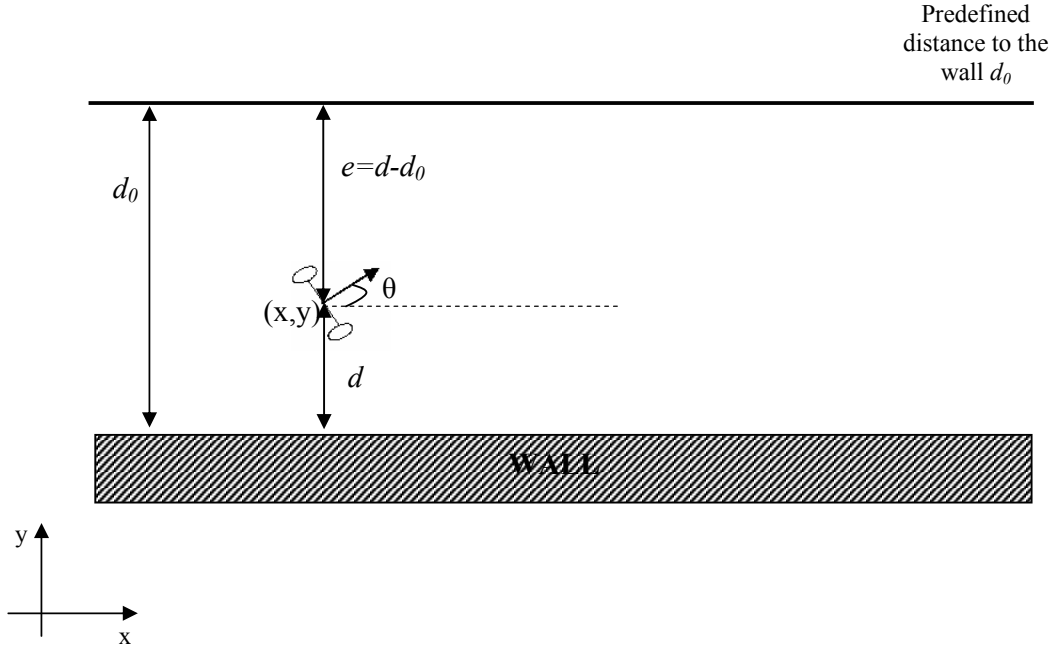


Figure 5.3: Wall following technique description

Let $z(k+1)$ be the extracted line parameters that constitute the vector of observations and $R(k+1)$ be its associated observation covariance matrix.

The observation process has the following form:

$$z(k+1) = h(x(k+1|k), w(k+1)), \quad (5.12)$$

where $h(\cdot)$ is the function that links the state of the system to the measurement and w represents the measurement noise.

Thus, the equation below is obtained:

$$h(x(k+1|k)) = \begin{bmatrix} y_{\text{predefined}} - e_{k+1} \\ \theta_{k+1} \end{bmatrix}. \quad (5.13)$$

Based on these assumptions, the correction step is described by the following equations:

$$G(k+1) = P(k+1|k) \nabla h^T (\nabla h P(k+1|k) \nabla h^T + R(k+1))^{-1} \quad (5.14)$$

$$X(k+1|k+1) = X(k+1|k) + G(k+1) \text{Innov}(k+1) \quad (5.15)$$

$$P(k+1|k+1) = P(k+1|k) - G(k+1) \nabla h P(k+1|k), \quad (5.16)$$

where $G(k+1)$ is the Kalman gain, ∇h is the Jacobian with respect to the system's state prediction $X(k+1|k)$, and $Innov(k+1)$ is the innovation.

The innovation $Innov(k+1)$ reflects the difference between the predicted measurement and the actual measurement. This is expressed as:

$$Innov(k+1) = z(k+1) - h(x(k+1|k)). \quad (5.17)$$

Thus, the *a posteriori* state estimate $X(k+1|k+1)$ is expressed as a linear combination of the *a priori* state estimate $X(k+1|k)$ and a weighted difference between the actual measurement and a measurement prediction as shown in Equation (5.15).

5.3 Mid-line Following

Mid-line following (MLF) behavior works in a very similar manner as wall following behavior. The same control law is also used here. This behavior needs the detection of two parallel walls. In comparison with the wall following behavior where only one main wall was used, for mid-line following behavior another wall parallel to the main wall must be found. The mid point of the corridor is found by averaging the two distances from the robot to the two walls.

Mid-line following presents an important advantage with respect to wall following in that it is flexible in its usage for different types of indoor environments. Typically, indoor environments consist of corridors having different widths. Performing mid-line following allows the mobile platform to maintain its mid-line trajectory in the corridor irrespective of its width. Wall following is not as flexible as MLF in the context of handling varying width corridors/spaces. This is due to the fact that the mobile platform follows the walls by maintaining a predefined distance with respect to it. When it changes from the right to the left wall (or vice-versa), if the predefined distance is not in the middle the mobile platform traverses the corridor in zig-zag fashion.

5.4 Center of Free Space

There are many alternatives for the positioning of a robot in a room. The robot could detect doors and stop at the entrance or it could enter a room until a pre-defined distance has been reached. The drawback of these methods is that they don't consider the dimensions and the shape of the room. This can lead to non-optimal positions such as places close to obstacles obtruding most of the field of view of the room. Therefore, we assume that the position in the room with the maximum free space around it is the one with the highest probability of extracting numerous and characteristic features. This ensures high distinctiveness of the recorded fingerprints in the mapping process.

5.4.1 Method Description

The laser scanner data is used to compute the center of gravity of the free space around the robot. Because the angle between two scan points is constant, the density of points is inversely proportional to the distance. To account for this, the points must be weighted, giving the following equation for computing the gravity center:

$$\vec{x}_G = \frac{1}{n} \left[\sum_i \omega_i x_i, \sum_i \omega_i y_i \right]^T \quad (5.18)$$

where ω_i the corresponding weights which are set equal to the distances from the scan points to the robot [Lamon03] and n the sum of the weights.

This method is depicted with a simple example, composed only of six points, in Figure 5.4.

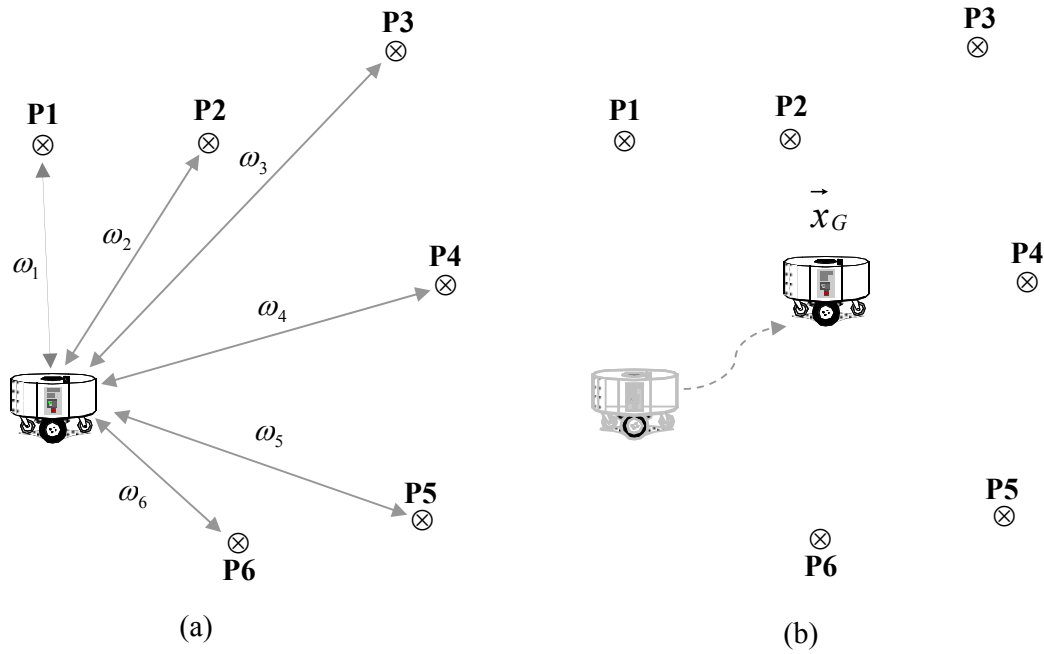


Figure 5.4: *Center of Free Space: (a) Shows the robot, the six points observed P1 to P6 and the corresponding weights set to the distance between the points to the robot; (b) Illustrates the calculated center of free space*

5.4.2 Experimental Results

In order to validate this method, the positioning capabilities of the robot have been tested and analyzed. For the experiments, the robot visited ten rooms of the environment depicted in Figure 4.8 and tested the method four times in each room. The results are

conclusive: the robot consistently reached the center of the free space in every room it was taken to (see Figure 5.5).

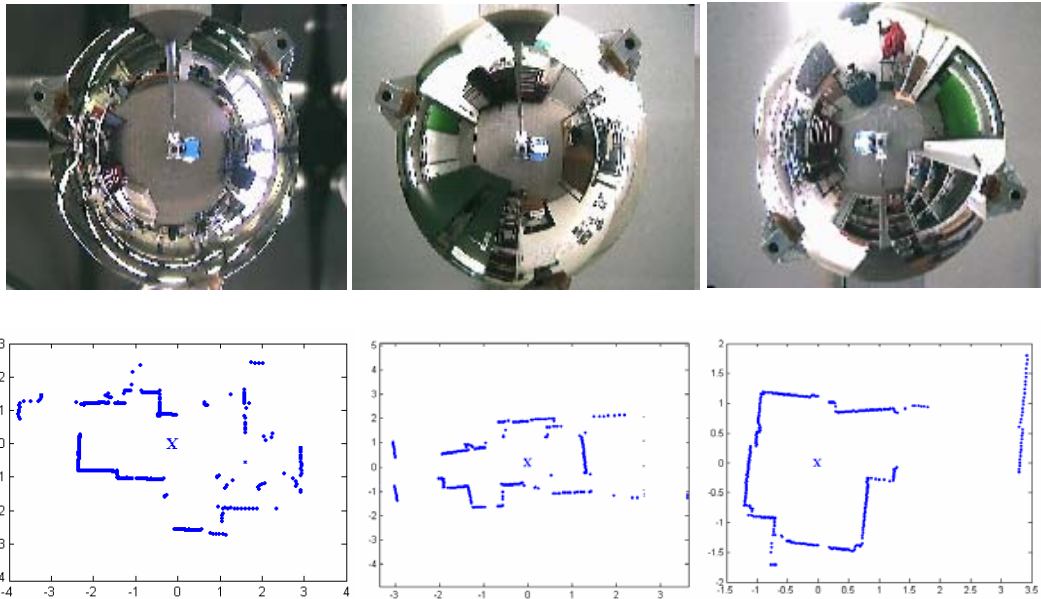


Figure 5.5: *Different views of the free space taken by the panoramic vision system, with their correspondent data given by the laser scanner. The position of the robot corresponds to the center of the free space measured with the laser scanner, represented by a 'X'.*

The repeatability was also tested: the robot reached an area of radius of 15 cm around the center of the free space.

5.5 Indoor Structures and Door Detection using Bayesian Programming

Since indoor office environments are usually orthogonal and highly structured and consist of offices, doors and corridors, a suitable method for the detection of these indoor structures and doors is useful for navigation. This section presents a model for learning the structure of the environment (e.g. corridor, door, x-crossing) and recognizing the learned situations based on a probabilistic approach, when only the laser scanner is used. The approach described here has been done in collaboration with Luc Dobler and Guy Ramel, and has been published at the IEEE International Conference on Intelligent Robots and Systems (IROS) in 2004 [Tapus04c].

Different techniques have been described in literature to identify the structure of the environment [Kortenkamp94], [Franz98], [Limketkai05]. The work most closely related to ours is one by Aycard in [Aycard97], in which places are learned by using the second-order Hidden Markov Models (HMM2). The maximum likelihood estimation criterion was used to determine the best model parameters given the corpus of observations, in order to perform the learning process. The recognition is carried out

using the Viterbi algorithm. For the experiments, ultrasonic and infrared sensors were employed. Unfortunately these sensors are very sensitive to ambient light, object color, object orientation and the surface of reflection.

The method employed here is probabilistic in that it takes into account the uncertainty, impreciseness and incompleteness of knowledge specific to the environment. The approach is based on the Bayesian Programming formalism (see Appendix A for a detailed description).

5.5.1 Implementation Related Assumptions

A simple navigation system has been implemented on the robot, so that the robot stays in the middle of the corridor (i.e. mid-line following), parallel to the two walls constituting the corridor. This navigation methodology has been detailed in Section 5.3.

5.5.2 Indoor Structures and Doors Corpus

A corpus with the main indoor structures situations and doors that the robot must detect is constructed as shown in Figures 5.6(a) and Figure 5.6(b). In Figure 5.6(a) the indoor structures are illustrated. These are: corridor, X-crossing, T-crossing and L-Intersection. For the T-crossing we have chosen three cases. This decision is justified by the fact that the recognition is only made with probability distributions and if we would have had only one state for the T-crossing, the distribution would not have been sufficient for this state. Figure 5.6(b) depicts different types of doors: closed door, right partially-opened door, left partially-opened door, opened door and no door.

The environment is assumed to be orthogonal i.e. the case for most office buildings, including the institute building where our robot was operating. The above mentioned limitation is not an inherent loss of generality because it is only a simplification for the current implementation and not a general requirement for the algorithm.

5.5.3 Bayesian Program for Indoor Structures and Doors Detection

The main goal of this work is to determine the state in which the robot may be, for instance that the robot is in a corridor and has a partially-opened door on its right.

First, in a learning phase, different indoor situations and doors are presented to the robot. During this phase, the robot collects the values of its sensory variables that correspond to a certain state. This data set is then used to identify the free parameters of the parametric forms.

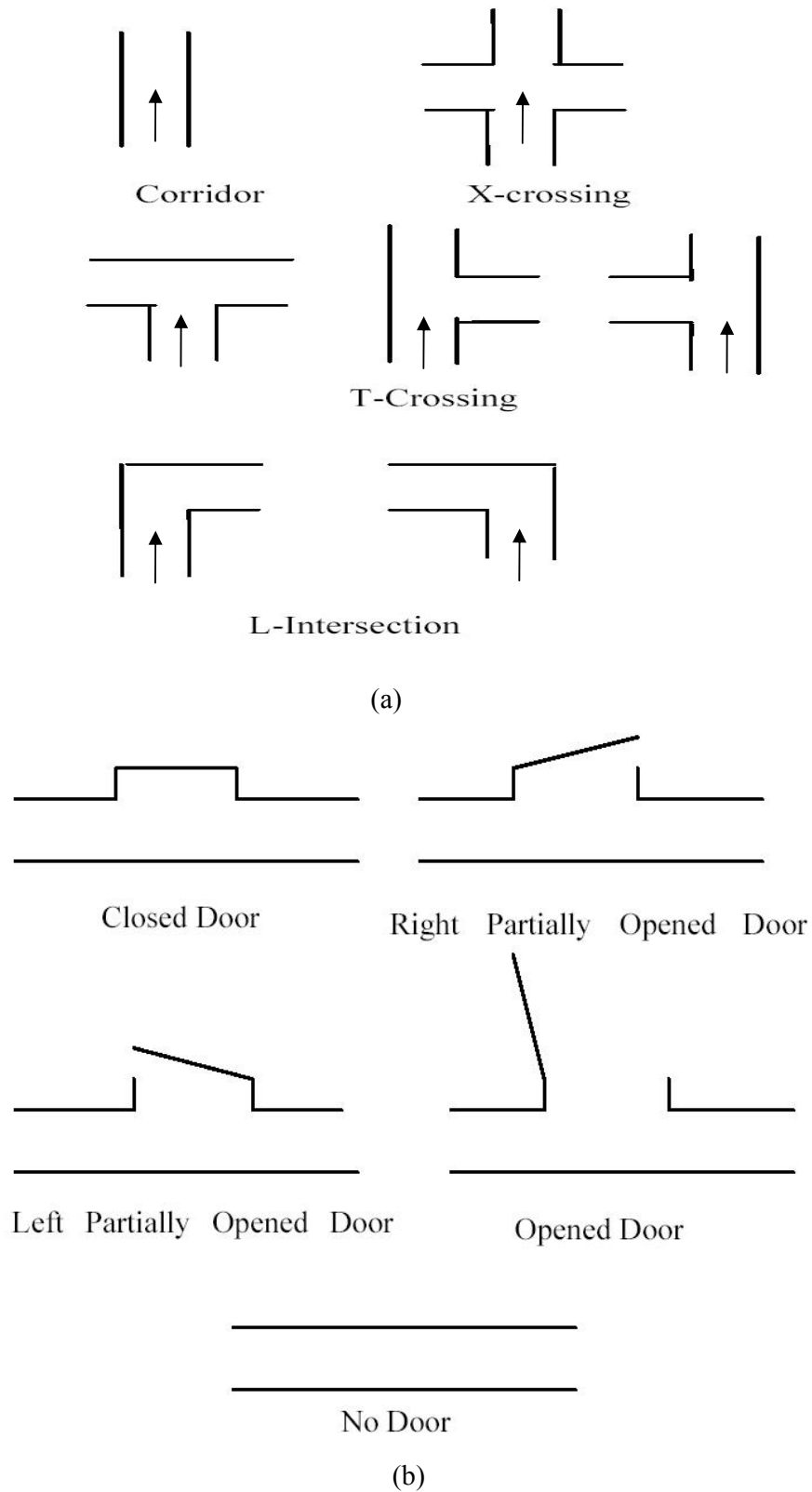


Figure 5.6: (a) The indoor structures to learn: Corridor, X-Intersection, T Intersection, L-Intersection. (b) The types of door to learn: closed door, right partially-opened door, left partially-opened door, opened door and no door.

The second step of the approach is the phase of application, when the robot has to recognize the situation in which it is. To solve this problem, the robot will extract the actual observation and answer the following probabilistic question:

$$sit^* = \arg \max_{sit \in State} P(sit | obs \pi \delta) \quad (5.19)$$

The actual state of the robot may be recovered by comparing the actual observation with the database of known situations and choosing the situation sit^* with the highest probability.

5.5.3.1 Bayesian Program

Figure 5.7 illustrates the Bayesian Program used for the indoor structures and door detection. A Bayesian Program, as shown in Figure A.1 (see Appendix A), is divided in two main parts: the decomposition and the question. The decomposition is also composed of two parts: the specification and the identification. All these are described in detail in the next sub-sections.

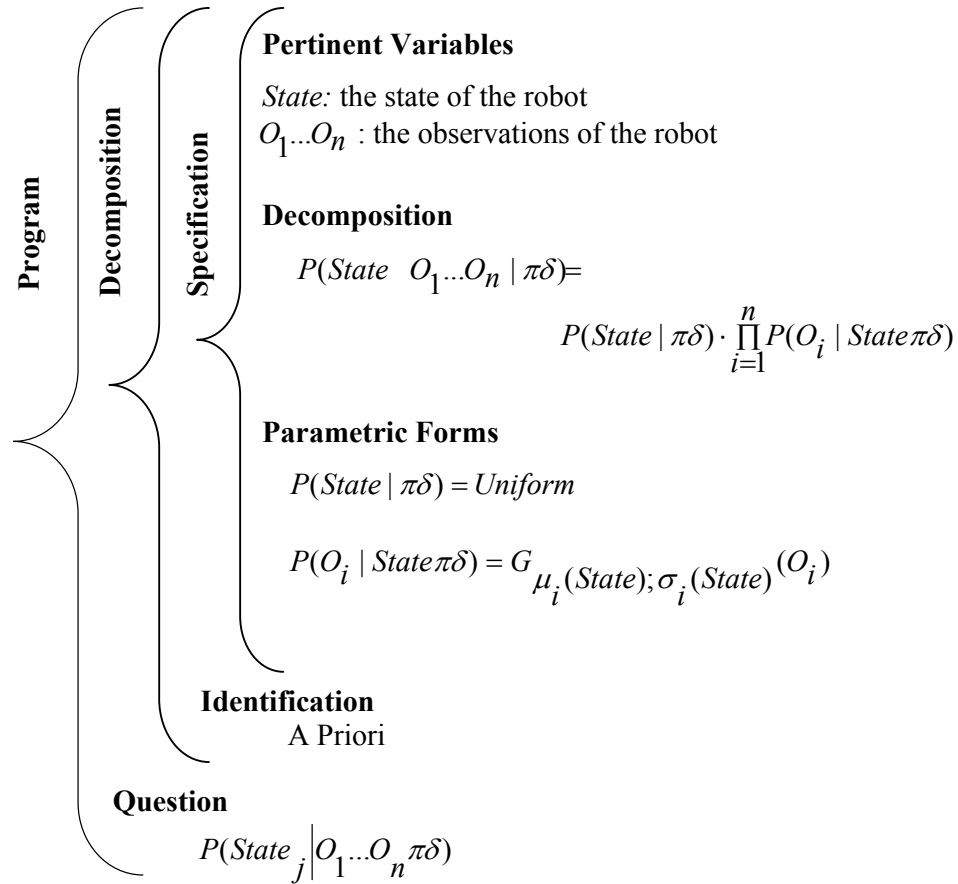


Figure 5.7: The Bayesian Program used for the indoor structures recognition and door detection, following the unique notation and structure described in Appendix A.

A Specification

This part defines the preliminary knowledge.

- **Pertinent Variables**

First of all, the variables that are pertinent for this task are defined as follows: O_1, K, O_n the set of observations and *State* the situation in which the robot can be.

In our case, the observation O_i is equal to the maximum distance found with the laser scanner in the corresponding section i (see Figure 5.8(a)). This choice reinforces the robustness of the observation O_i with respect to the robot's orientation and to the noise in the environment. For example, if the robot's orientation changes with an angle smaller than $360^\circ/n$ degrees, the robot finds similar values; a person near the robot does not influence the observation since the robot takes into account only the maximum distances.

- **Decomposition**

The decomposition consists in decomposing the joint distribution $P(\text{State } O_1 \dots O_n | \pi)$ into a product of simpler terms. This step also permits to express the independence and dependence relationships between the variables. This distribution is conditioned by both π the preliminary knowledge we are defining, and δ a data set that will be provided during the learning phase.

Since it is considered that the observations O_1, K, O_n are dependent on the location the following decomposition results:

$$P(\text{State } O_1 \dots O_n | \pi \delta) = P(\text{State} | \pi \delta) \cdot \prod_{i=1}^n P(O_i | \text{State} \pi \delta) \quad (5.20)$$

- **Parametric Forms**

From the result of the decomposition formula (see Figure 5.7) two different kinds of probability distributions can be distinguished:

- Since no a priori information about the different indoor structures situations or about the doors is available, each situation is considered to be equally probable and consequently the probability of a state given all the a priori knowledge is expressed as a uniform distribution.

$$P(\text{State} | \pi \delta) = \text{Uniform} \quad (5.21)$$

- To determine the probability of each observation O_i given the indoor structure or door situation and all the a priori knowledge, a Gaussian parametric form is assigned:

$$P(O_i | \text{State} \pi \delta) = G_{\mu_i(\text{State}); \sigma_i(\text{State})}(O_i) \quad (5.22)$$

B Identification

Different indoor situations and doors are presented to the robot. During this phase, the robot collects the values of its sensory variables that correspond to a certain state. This data set δ is then used to identify the free parameters of the parametric forms.

This phase corresponds to the identification of the free parameters appearing in the parametric forms. In our case, the parametric forms contain free parameters, the mean and the standard deviation in the Gaussian distribution. It is necessary to give numerical values to these parameters so as to finish the description. These numerical values can be obtained through a learning process. A supervised learning is used here. The robot visits different situations denoted by *State* and in each situation $sit \in State$ for each observation O_i the robot determines the values of the $\mu_i(State)$ (i.e. mean) and $\sigma_i(State)$ (i.e. standard deviation). A method similar to the one depicted in [Lebeltel99] is adopted. These two values (i.e. mean and standard deviation) are identified as follows:

$$\mu_i(State) = \frac{\sum_{i=1}^n O_i}{n} \quad (5.23)$$

$$\sigma_i(State) = \sqrt{\frac{\sum_{i=1}^n O_i^2}{n} - \mu_i^2(State)}, \quad (5.24)$$

where n is the number of times the robot was in a situation *State* during the learning process.

The mean and standard deviation are updated at each new measurement, thus permitting an incremental learning process.

C Question

Once the two previous steps are identified, the robot can answer any question concerning this joint distribution (see Equation (5.20)). Therefore, the question described in the Bayesian Program shown in Figure 5.7, $P(State_j | O_1 \dots O_n \pi \delta)$ can also be answered.

5.5.3.2 Discussions

The interesting point is that the same Bayesian Program is used for both indoor structures recognition and door detection. The only difference is the domain of the variable *State* and the database used for the learning. For the case of indoor structures identification, the variable *State* contains the following values: corridor, X-crossing, T-crossing and L-Intersection. The 360° view of the robot is divided in n equal parts, as shown in the Figure 5.7(a). In the case of door detection the variable *State* takes the values: closed door, right partially-opened door, left partially-opened door, opened door and no door. The observation field is portioned into four Zones. The method of

splitting the field of view of the robot in zones permits also the detection of the direction of doors. The Bayesian Program gives an answer for each of these zones in order to detect a door in front, behind, to the left and to the right of the robot. Each of these zones is split in n equal slices, as illustrated in the Figure 5.8(b). For both cases: indoor structures identification and door detection, the number of divisions n is fixed to 8.

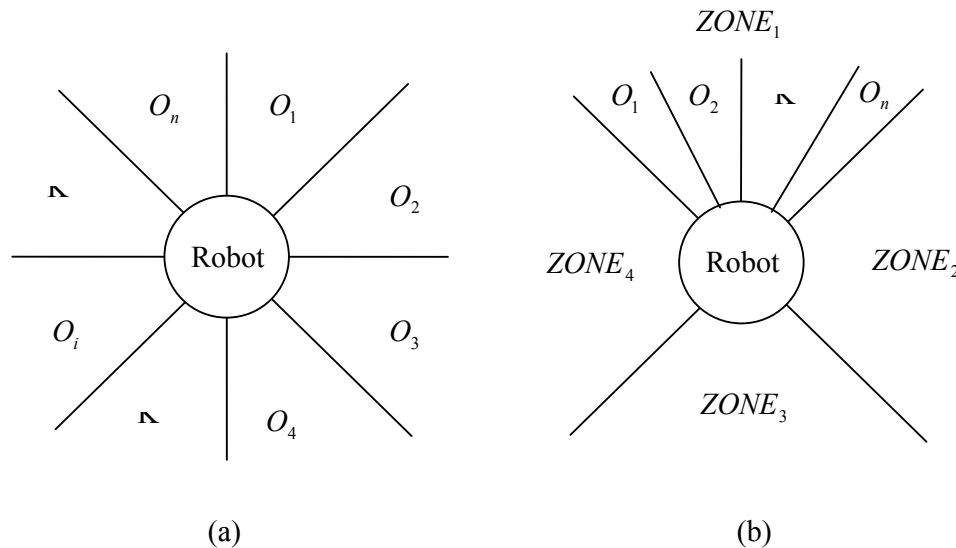


Figure 5.8: *The observation field of view of the robot: a) for the topology recognition the 360° view of the robot is portioned in n parts; b) for doors detection the view of the robot is divided in four Zones and each zone is portioned in n parts. In our implementation, for both cases a) and b) n is equal to 8.*

This method is very flexible with respect to the utilization of the same program for two different tasks: indoor structure recognition and door detection.

5.5.4 Experimental Results

The approach has been tested in a 50 m x 25 m portion (see Figure 4.8) of our institute building. For the experiments, the BIBA robot (see Figure 1.2 (a)) has been used.

A training data has been collected and a Gaussian model for each of the seven indoor structures and for each of the five door types have been constructed. The robot was placed 25 times in each situation in order to construct a robust training corpus. In order to complete the training, for each situation and each observation the Gaussian parameters (the mean and the standard deviation) have been calculated, as shown in Section 5.5.3.1.

To test our approach for indoor structures and doors recognition, 50 tests for each situation have been performed.

The results are summarized in Table 5.1 and Table 5.2. In these two tables the results of indoor structure identification and door detection are presented. Each row corresponds to a situation that the robot observed and each column corresponds to a situation that the robot recognized.

In Table 5.1, it can be seen that for instance, corridor recognition is very reliable, with a recognition rate of 98%. For the indoor structures identification, the percentage of successfully recognition is between 82% and 98%, with an average of 92.2%. It is important to notice that the falsely recognized situations are always similar to the real indoor structures situations. It would have been more compromising to recognize a situation like \lrcorner or \llcorner where the situation were \lrcorner and \llcorner respectively, because these are opposite indoor structure situations.

TABLE 5.1: The table shows the results of indoor structure recognition. The following notations has been used: \parallel (corridor), \lrcorner (left L-Intersection), \lrcorner (right L-Intersection), \lrcorner (left T-crossing), \llcorner (right T-crossing), \lrcorner (middle T-crossing), \lrcorner (X-crossing).

	\parallel	\lrcorner	\lrcorner	\lrcorner	\llcorner	\lrcorner	\lrcorner
\parallel	98%			2%			
\lrcorner		82%				18%	
\lrcorner			96%			4%	
\lrcorner	8%			86%			6%
\llcorner	6%				90%		4%
\lrcorner						98%	2%
\lrcorner	4%						96%

In Table 5.2, it can be noticed that in the case of the "left partially opened door" situation, 36% of responses were false. Instead of detecting the "left partially opened door" situation, the "opened door" has been detected (e.g. see Figure 5.8). However, this is not very important if the context of recognition is the detection of doors without considering its aperture. A similar false detection can be observed in the case of a "closed door" situation, where there are 20% of "no door" detections. However, most of the false detections can still be considered good results knowing that to determine a "closed door" situation, a jump near the frame of the door must be found. Another false detection is identified in our experiments where 6% of "opened door" situations are detected as "no door" situations. The "opened door" situations are separated in two zones (see Figure 5.7(b)), due to the wrong (i.e. not parallel to the walls) orientation of the robot. Thus, the robot's observation doesn't match correctly with the learning situation.

The percentage of successful door detection is between 60% and 90%, and at an average of 80.4%.

TABLE 5.2: The table shows the results of door detection. The following notation has been used: *nd* (no door), *cd* (closed door), *od* (opened door), *lpod* (left partially opened door) and *rpod* (right partially opened door).

	nd	cd	od	lpod	rpod
Nd	90%		4%	6%	
Cd	20%	76%			4%
Od	6%		90%	4%	
Lpod			36%	60%	4%
Rpod	4%		10%		86%

The application of our Bayesian Program for door detection shows how well suited our approach is, for this type of recognition. If we regroup the three situations in which the door is partially or completely opened, we have three possible situations "*opened or partially opened door*", "*closed door*" and "*no door*". The results for these three situations are even better than the previously mentioned classification of door states, the percentage of successful recognition being of 94%, 76% and 90% respectively. Another interesting statistic was computed in order to detect the percentage of successful door and no-door detection. The results are quite convincing - 90% and 94% respectively.

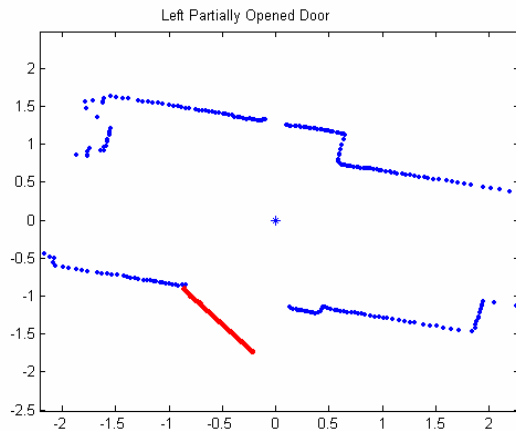


Figure 5.9: *False Positive Door Detection. A "left partially opened door" situation has been detected as an "opened door" situation.*

A combination between the two Bayesian Programs to perform the simultaneous indoor structure recognition and door detection has been implemented. This has been found to produce very promising results. A new learning corpus of 50 measurements has been constructed and 250 tests (50 tests for indoor structure identification and 200 tests for door detection) have been performed. An indoor structure or a door is recognized if the actual observation matches exactly with the real situation. Substitution errors occurred during the tests. We have divided the substitution errors in two types: satisfactory substitutions (applied only for the detection of doors) and false substitutions.

These are defined as:

- **Satisfactory Substitution:** The recognized situation is a confusion between the states: "*right partially opened door*", "*left partially opened door*", "*opened door*". For instance, the robot observes a "*right partially opened door*", when an "*opened door*" is present in the map.
- **False Substitution:** The confusion of a state with another one, not in the category of satisfactory substitution.

Table 5.3 summarized the results obtained for the global recognition.

TABLE 5.3: *The table shows the results of global recognition*

	Number	%
Tests	250	100%
Recognized	206	82.4%
Satisf. Substituted	18	7.2%
False Substituted	26	10.4%

From the experiments, it can be observed that the different situations (indoor structures and doors) are globally well recognized. The results have given a percentage of successful recognition (classification) of 82.4% and 7.2% of satisfactory substitution (see Table 5.3).

5.6 Summary

This chapter presented different exploration tools which are used in the subsequent chapters. The behaviors described are wall following, mid-line following, find and go in the center of free space, indoor structures identification, and doors detection.

Wall-following and mid-line following are two well known control laws that permit the robot to follow a wall or corridor. Mid-line following behavior is more suitable for typical indoor office environments due to its independence of corridor widths. In order to be able to ensure high distinctiveness of the observation, a new approach for positioning has been developed: the center of free space. This simple method has proven its efficacy in indoor environments. The robot reached an area of radius of 15 cm around the center of the free space, when the test was performed for the same room. It is very important for the mobile platform to determine the indoor structures that surround it, so as to easily navigate and apply exploration strategies. A new technique for identification of indoor structures and detection of doors has been presented. Bayesian Programming has been used. The success rate for recognition of indoor structures is in average 92.2% and for detection of doors is in average 80.4%. These are very good results. The main sensor used for all exploration techniques was the 2D laser range finder by SICK.

All these exploration tools help the robot to autonomously move within the indoor environment, and therefore to build maps of its environment.

6

Simultaneous Localization and Mapping (SLAM) with Fingerprints of Places

Navigation described by Gallistel [Gallistel90], as the capacity to localize itself with respect to a map, is an elementary task that a mobile and autonomous robot must carry out. To navigate reliably in indoor or outdoor environments a mobile robot must know where it is. For this, the robot needs to construct or maintain a spatial representation of the environment. In Chapter 4, different topological localization approaches for the case of a priori maps, were exhibited. Here, we approach the SLAM (Simultaneous Localization and Mapping) problem that is of a “chicken and egg“ nature – to localize the robot, a map is necessary and to update a map the position of the mobile robot is needed.

The objective of the work presented in this chapter is to enable autonomous navigation without relying on maps a priori learned and without using artificial landmarks. Therefore, this chapter describes a new method for incremental and automatic topological mapping and global localization with POMDP (Partially Observable Markov Decision Processes) using fingerprints of places.

6.1 Related Work

A robust navigation system requires a spatial model of the physical environment such as a metric or topological map. Approaches using metric maps are suited when it is

necessary for the robot to know its location accurately in terms of metric coordinates. However, the state of the robot can also be represented in a more qualitative manner, similar to the way humans do it. The information can be stored as mental or cognitive maps – a term introduced for the first time in [Tolman48] – which permit an encoding of the spatial relations between relevant locations in the environment. This has led to the concept of topological representation. The topological map can be viewed as a graph of places, where at each node the information concerning the visible landmarks and the way to reach other places, connected to it, is stored. The topological representation is compact and allows high-level symbolic reasoning for map building and path planning.

Although literature related to SLAM (Simultaneous Localization and Mapping) is very vast, we concentrate here on papers that we consider as most important and that have directly influenced our thinking and research work.

The SLAM problem, as the construction of maps while the robot moves through the environment and the localization with respect to the partially built maps, was introduced in robotics in a seminal paper by Smith and Cheeseman [Smith86] in 1986. One of the first implemented systems was developed by Moutarlier and Chatila [Moutarlier89]. This approach used the Extended Kalman Filter (EKF) to estimate the posterior over robot pose and maps. Leonard and Durrant-Whyte in [Leonard92] proposed a similar stochastic method to SLAM. Metric maps are spatial representations that have been extensively studied in the robotics community. The stochastic map technique to perform SLAM [Castellanos99], [Dissanayake01], [Leonard92], and the occupancy grids approaches [Thrun98] are typical examples belonging to this kind of space representation. More recent vision-based metric approaches use SIFT (Scale Invariant Feature Transforms) features [Se02]. The SIFT approach detects and extracts local feature descriptors that are invariant to illumination changes, image noise, rotation and scaling. All these metric methods are used with high precision sensors. Thus, mapping yields a precise representation of the environment and consequently localization is accurate. However, metric SLAM can become computationally very expensive for large environments. Thrun, in [Thrun00], proposes probabilistic methods that make the metric mapping process faster and more robust. However, metric approaches also suffer from other shortcomings. A negative aspect of metric maps is that they are not easily extensible so as to be useable for higher level, symbolic reasoning. They contain no information about the objects and places within the environment.

Topological approaches to SLAM attempt to overcome the drawbacks of geometric methods by modeling space using graphs. Significant progress has been made since the seminal paper by Kuipers [Kuipers78], where, an approach based on concepts derived from a theory on human cognitive mapping is described as the body of knowledge representing large scale space. Kortenkamp and Weymouth in [Kortenkamp94] have also used cognitive maps for topological navigation. They defined the concept of *gateways* which have been used to mark the transition between two adjacent places in the environment. They have used the data from sonars combined with vision information in order to achieve a rich sensory place-characterization. Their work has been an amelioration of Mataric's approach [Mataric90], contributing towards the reduction of the perceptual aliasing problem. The improvement is obtained by introducing more sensory information for place representation. A model by Franz, Schölkopf and Mallot [Franz98] was designed to explore open environments within a

maze-like structure and to build graph-like representations. Their method has been tested on a real robot equipped with an omnidirectional camera. In [Hafner00] and [Owen98], the authors have used a model based on a self-organizing map which creates a topological representation of the environment while the robot explores it. Another topological model is described in [Choset01]. The environment is represented with the help of a generalized Voronoi graph (GVG) and the robot is localized via a graph matching process. Most recently, Beeson et al. have used Extended Voronoi Graphs (EVG) to demonstrate place detection in the context of topological maps [Beeson05]. In general, topological maps are less complex and permit more efficient planning than metric maps. Moreover, they are easier to generate. Maintaining global consistency is also easier in topological maps compared to metric maps. However, the main problems to deal with, when working with topological maps are the perceptual aliasing (i.e. observations at multiple locations are similar) and the automatic establishment of a minimal topology (nodes).

Researchers have also integrated both the metric and topological paradigms, thereby obtaining a hybrid system. Thrun, in [Thrun00], uses occupancy-grid based maps in order to build the metric map. The topological map is extracted from the grid-based map. Learning a topological representation depends on learning a geometric map, which relies on the odometric capability of the robot. However, in large environments, it is difficult to maintain the consistency of the metric map, due to the drift in the odometry. In [Tomatis03], Tomatis et al. have conceived a hybrid representation, similar to the previously mentioned work, comprising of a global topological map with local metric maps associated to each node for precise navigation. Another hierarchical multi-resolution approach allowing for high precision for metric mapping using a relative map filter and distinctiveness for topological mapping with fingerprints of places is presented in [Martinelli03a]. The authors of [Lisien03] have illustrated an extension of the model described in [Choset01] to H-SLAM (i.e. Hierarchical SLAM), by combining the topological and feature-based mapping techniques. Another hybrid approach is described in [Dufourd04]. Their model combines different representations (i.e. frontier-based, space-based, grid-based and topological), allowing in this way to improve the SLAM robustness and creating a more complex and useful spatial representation for reasoning and path planning.

The method presented in this dissertation uses fingerprints of places to create a topological model of the environment. The fingerprint approach, by combining the information from all sensors available to the robot, reduces perceptual aliasing and improves the distinctiveness of places. For instance, in contrast with the SIFT features, the features used in the fingerprint of place are higher-level and they are “situated“ in the context of the environment. As they give higher-level information, they can draw out the semantics of the environment to a greater extent than the SIFT features.

The main contribution of this chapter is the construction of a topological mapping system combined with a localization technique, both relying on fingerprints of places. This reduces the SLAM problem to one of matching the fingerprints representing the environment and does not involve topological-graph matching itself. This fingerprint-based approach yields a consistent and distinctive representation of the environment and is extensible in that it permits spatial cognition beyond just pure navigation.

6.2 Environment Representation with Fingerprints of Places

The same environmental model as the one described in Chapter 4.2 is also used here. The environment is represented in a topological fashion. The topological map can be viewed as a graph of places, where at each node the information concerning the visible cues and the way to reach other places connected to it, is stored. Fingerprints of places, described in Section 3.4, are used to represent places and therefore the nodes in the topological framework. The topological representation obtained is compact and allows high level symbolic reasoning for map building and navigation.

6.3 Topological Localization and Map Building

Map building is a fundamental task for creating representations of the environment the robot is moving in. The maps thus built are used for localizing the mobile robot.

In this section the methods used to automatically build topological maps and globally localize the robot, are described. This approach has also been illustrated in the works [Tapus05b] and [Tapus04b], respectively.

6.3.1 Topological Mapping Technique

While navigating in the environment, the robot first creates and then updates the global topological map. One of the main issues in topological map building is to detect when a new node should be added in the map. Most of the existing approaches to topological mapping place nodes periodically in either space (displacement, Δd) or time (Δt) or alternatively attempt to detect important changes in the environment structure. Any of these methods cannot result in an optimal topology. In contrast, the approach presented in this work is based directly on the differences in the perceived features.

In the following sub-sections, the fingerprint-based approach for incremental and automatic topological mapping is described. In addition, they clearly illustrate how a reliable and distinctive representation of the environment is obtained.

6.3.1.1 Exploration Strategy

The exploration strategy used in this work is very simple. The robot explores the corridors in a depth-first fashion, using a *mid-line following* behavior (see Section 5.3). A *wall-following* behavior (see Section 5.2) would also have been suited for corridor exploration. When open doors are detected (see probabilistic methodology for indoor structures and doors identification – Chapter 5.5), the robot explores the rooms and uses the *go to the center of free space*, behavior described in Chapter 5.4. The robot returns in the hallways by using a backtracking methodology.

6.3.1.2 Automatic Map Building

Instead of adding a new node in the map by following some fixed rules (e.g. distance, topology) that limit the approach to indoor or outdoor environments, the method described in this work introduces a new node into the map whenever an important change in the environment occurs. This is possible using the fingerprints of places. A heuristic is applied to compare whether a new location is similar to the last one that has been mapped.

The process of introducing a new node in the topological map is split into the following sequence of steps (see Figure 6.1):

- 1) Start with an initial node (i.e. fingerprint f_0)
- 2) Move and at each Δt (time) or Δd (distance), take a new scan with the laser scanner and a new image with the omnidirectional camera and generate the new fingerprint f_i
- 3) Calculate the probability of matching, $prob_matching$, between the fingerprints f_{i-1} and f_i respectively. Compute the dissimilarity factor, $dissimilarity$.

$$prob_matching = P(f_i | f_{i-1}) \quad (6.1)$$

$$dissimilarity(f_i, f_{i-1}) = 1 - prob_matching \quad (6.2)$$

- 4) If $dissimilarity(f_i, f_{i-1}) < \theta$ then
 - a. Add fingerprint f_i to the current node n_k
 - b. Calculate the new mean fingerprint of the node n_k
- Else
 - a. A new node n_{k+1} is inserted (added) in the map
 - b. Add fingerprint f_i to the node n_{k+1}
- 5) Repeat from step 2)

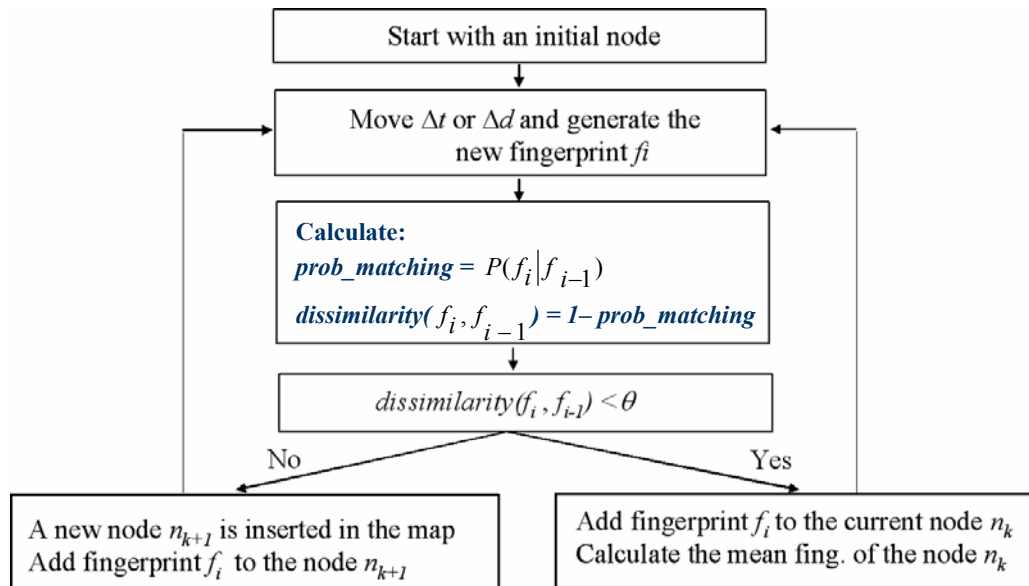


Figure 6.1: Flow-chart of the new topological node detection algorithm

In step 4), a threshold θ is defined as the maximum allowable dissimilarity (i.e. $1 - prob_matching$) between the fingerprints. The value of $prob_matching$ is calculated with the "global alignment with uncertainty" algorithm. This method is an adaptation of the global alignment algorithm usually used for comparing D.N.A. sequences, introduced by Needleman and Wunsch [Needleman70] (see Chapter 4.3.2.2). The value of the threshold is determined experimentally. The incremental nature of the approach permits incremental additions to the map and yields the most up-to-date map at any time. The basic process is depicted in the Figure 6.2.

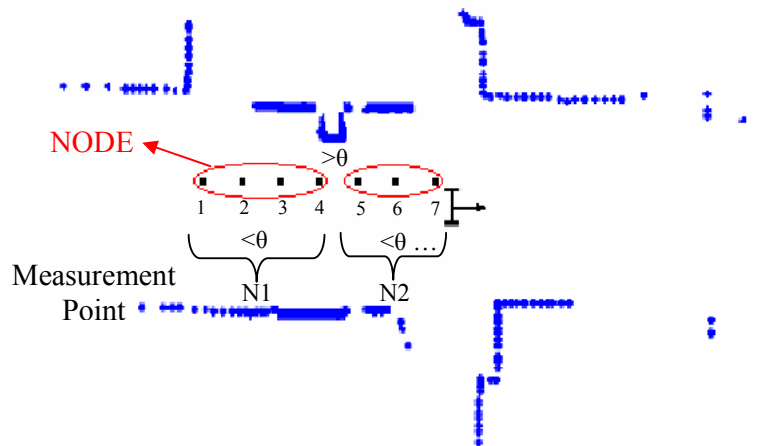


Figure 6.2: Adding a new node automatically to the topological map by moving in an unexplored environment. The image is composed of seven measurement points (i.e. fingerprints of places) represented by the black points. The blue points depict the data given by the laser range finder. The mapping system includes all the fingerprints of places in a node until a significant change in the environment occurs and the dissimilarity between the fingerprints is greater than the threshold θ .

It can be noticed in Figure 6.2 that the new node contains all posterior knowledge about the environment until the previous node.

As mentioned previously, a step in the construction of the map is the generation of a mean fingerprint for each node. The following sub-section explains this process. It uses the "global alignment with uncertainty" algorithm, described in Section 4.3.2.2 (see [Tapus04b]) for fingerprint matching.

6.3.2 Mean Fingerprint

One of the main characteristics of topological maps is their compactness. Each place is represented as a single node. As mentioned above (see Section 6.3.1.2), in this work, a new node is introduced in the topological map just when important changes into the environment occur. With this, at the end, each node will be composed of a set of similar fingerprints of places. In order to compact even more the current representation, a

unique identifier named the *mean fingerprint* is generated. This technique of clustering fingerprints of places into a single representation is described below.

6.3.2.1 Clustering of Fingerprints

As stated earlier, a fingerprint is extracted periodically in space (every Δd) or time (every Δt). A node is composed of several similar fingerprints that will be subsequently regrouped into a mean fingerprint. By choosing a suitable threshold θ , the mean fingerprint enables clustering of places into nodes.

As soon as a new fingerprint is added to the current node n_k , the mean fingerprint is updated by constructing the new mean fingerprint between the previous mean fingerprint and the newly introduced fingerprint.

The generation of the mean fingerprint between two fingerprints is performed in several steps, described briefly below. The first step in the mean fingerprint generation process consists of matching the two fingerprints involved. As the orientation of the robot is not known a priori or fixed beforehand, the robot estimates it by considering all the possible permutations of one fingerprint sequence with respect to another. The fingerprint matching algorithm, illustrated in Section 4.3.2.2, yields their best match. It can be seen in Figure 6.3 (Step 1), that once the two fingerprint sequences are aligned, they have the same length. The aligned fingerprints contain also the occlusion symbol, introduced in Section 4.3.2.1, representing a space inserted into the string sequence. In the second step, the mean fingerprint between two consecutively obtained fingerprints of places is computed. The mean fingerprint contains the features that matched during the fingerprint matching process and those with a high probability of existence.

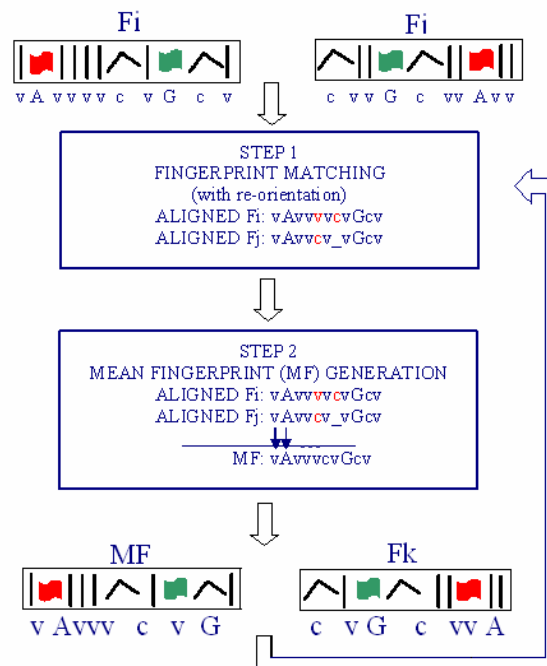


Figure 6.3: Illustration of the Mean Fingerprint Generation process. F_i , F_j and F_k are three consecutively obtained fingerprints.

The probability of existence of a feature (see section 3.4.4) is defined as the probability of being present in the environment when the robot perceives it. The higher the probability is, the more certain the existence of the feature is. The uncertainties of the features from the mean-fingerprint are weight averaged with the uncertainties of the features from the new fingerprint. For calculating the mean fingerprint for a specific node n_k , these two steps are repeated until all the fingerprints of that node are included in it. Figure 6.3 describes this process through a simple example.

The mean-fingerprint associates each node with a unique identifier. This enables the construction of a very distinctive and compact representation of the environment.

6.3.3 Indoor Topological Localization with POMDP

A series of localization techniques based on the *fingerprint* concept have been already presented in Chapter 4. These approaches perform a fingerprint-matching operation so as to localize the robot. The matching methods compare the observed features encoded in the fingerprints of places with the map fingerprints. Only the exteroceptive sensory information contained in fingerprints of places is used for matching, without taking into account the motion of the robot and the previous estimation.

Hence, for topological navigation, a *Partially Observable Markov Decision Process* (POMDP) model [Cassandra96] is used here. The POMDPs integrate both the motion and sensor reports data to determine the pose distribution. Thus, by adding the motion information to the system, new knowledge about the robot's position is acquired. The probability of being in a place is calculated in function of the last probability distribution, and the current action and observation.

A POMDP is defined as $\langle S, A, T, O \rangle$, where:

- S is a finite set of environment states;
- A is a finite set of actions;
- $T(s, a, s')$ is a transition function between the environment states based on the action performed;
- O is a finite set of possible observations;
- OS is an observation function.

With this information, the probability of being in a state s' (belief state of s') after having made observation o , while performing action a , is given by:

$$SE_{s'}^{t+1} = \frac{OS(o, s') \sum_{s \in S} T(s, a, s') SE_s^t}{P(o|a, SE^t)} \quad (6.3)$$

where SE_s^t is the belief of state S at the last step, SE^t is the belief state vector at the last step, and $P(o|a, SE^t)$ is the normalizing factor define as:

$$P(o|a, SE^t) = \sum_{s' \in S} OS(o, s') \sum_{s \in S} T(s, a, s') SE_s^t \quad (6.4)$$

The key idea is to compute a discrete approximation of a probability distribution over all possible poses in the environment. An important feature of this localization technique is the ability to globally localize the robot within the environment. More details about this approach can be found in [Cassandra96].

The elements of the POMDP are defined for the present indoor model as follows:

- Set of States S

The set of environment states is composed of the topological nodes automatically detected with the algorithm described in Section 6.3.1.2. Nodes are situated only in distinctive places within the environment. The results of mapping show in Section 6.4.1 that nodes are usually placed in corridors in front of doors, in offices and in other distinctive places within corridors.

- Set of Actions A

A represents the set of actions that the robot can execute. The office buildings are usually quite simple and most of the time orthogonal, which is the case of our institute building (see Figure 4.8) where the BIBA robot (see Figure 1.2(a)) operates. The actions used by the robot are *mid-line following* (described in Section 5.3) in the corridors, *go to the center of free space* (see Section 5.4) when doors are detected with the probabilistic method depicted in Section 5.5, and *go out the office room*, which is the following/trailing of the reverse trajectory of the *go to the center of free space* action.

- Transition Function

The transition function

$$T(s, a, s') = P(S^{t+1} = s' | S^t = s \wedge A^t = a) \quad (6.5)$$

calculates the probability of being in the world state s' assuming that the previous world state was s and the robot executed action a . As previously mentioned, the states, in which the robot can be, are the offices, and the corridors. To each fingerprint of a place, an indoor environment structure and a door situation is associated during map building. For each action a the probabilities of the transition function are calculated experimentally, and they are expressed as follows:

$$T(s, a, s') = \frac{\text{the number of transitions from state } s \text{ to state } s'}{\text{the number of transitions from state } s} \quad (6.6)$$

- Set of Observations O

The set of observations O is composed of the fingerprints of places generated by the robot in the environment. These observations are very distinctive since distinctiveness is one of the main characteristics of the fingerprints of places. An observation contains information given by the exteroceptive sensors and designates a subset of the world state.

- Observation Function

The information for the observation function OS within the topological framework is given by the fingerprint matching algorithm, described in Chapter 4.

The probabilistic observation function OS is given as follows:

$$\begin{aligned} OS(o, s') &= P(O^t = o | S^t = s') \\ &= P(O^t = f_{obs} | S^t = f_{map_i}) = \frac{1}{Z} \frac{1}{GlobalAlignment(f_{obs}, f_{map_i}) + 1}, \end{aligned} \quad (6.7)$$

where GA gives the probability of matching between two fingerprints and it is calculated with the global alignment with uncertainty algorithm described in Chapter 4, and Z is the normalization factor. The f_{obs} is the observed fingerprint and f_{map_i} is the map fingerprint, corresponding to the i -node. The normalization factor Z is described as:

$$Z = \sum_{s \in S} \frac{1}{GlobalAlignment(f_{obs}, f_{map_i_s}) + 1} \quad (6.8)$$

6.3.4 Control Strategy

The computation of an optimal POMDP control strategy for large environments is computationally intractable. In order to obtain sub-optimal solutions, simple heuristic control strategies are proposed [Cassandra96]. An example of such strategy is the *most likely state* (MLS). This means that the world state s with the highest probability is found and the action a that is optimal for that state is executed. However, in this dissertation, the entropy of the probability distribution over the states of the topological map is used. The entropy of a probability distribution p is:

$$H(p) = - \sum_{s \in S} p_s \log p_s, \quad (6.9)$$

where $p_s \log p_s = 0$ when $p_s = 0$. The lower the value is, the more certain the distribution is. When the robot is "confused", the entropy is high. So the POMDP is confident about its state if the entropy is smaller than a fixed threshold:

$$H(p) < \psi, \quad (6.10)$$

where ψ is the threshold experimentally calculated. When the robot is confident the action a that is optimal for that state is executed. Otherwise if the POMDP is unconfident about its state the robot does *mid-line following* if the preceding action was *mid-line following* and *go out the office room*, if the previous action was *go to the center of free space*. The robot tries to reach and follow the corridor where it expects to find more information.

6.3.5 Map Update

While navigating in the environment, the robot first creates and then updates the global topological map. By using a POMDP (Partially Observable Markov Decision Process), a discrete approximation of a probability distribution over all possible poses in the environment is computed.

The entropy (see Equation (6.9)) of a probability distribution is used here. Therefore, the strategy of updating the map will be the following:

- When the entropy of the belief state is low enough, the map will be updated and so the fingerprint and the uncertainty of the features will also be updated.
- If the entropy is above a threshold ψ , then the updating will not be allowed, and the robot will try to reduce the entropy by continuing the navigation with localization.

Similar to [Tomatis03], when the robot feels confident concerning its state, it can decide if an extracted feature is new by comparing the observed fingerprint to the fingerprint from the map, corresponding to the most confident state. This can happen either in an unexplored portion of the environment, or in a known portion where new features appear due to the environmental dynamics. The features from the fingerprint come with their extraction uncertainty $u_{feature}$. When a feature is re-observed, the uncertainty of the feature from the map fingerprint is weight averaged with the uncertainty of the extracted one. The weight depends on the type of feature. Since the extraction of features with the laser scanner is more robust than the ones extracted with the camera, a higher weight is given to them. In our case, we choose to represent that as follows:

$$u_{feature_map_t} = \begin{cases} \frac{u_{feature_map_t-1} + u_{feature_t}}{2}, & feature \in laser_features \\ \frac{2u_{feature_map_t-1} + u_{feature_t}}{3}, & feature \in camera_features \end{cases} \quad (6.11)$$

Otherwise, if the robot does not see an expected feature the uncertainty is decreased. The following equation expresses our choice for decreasing the uncertainty of a feature:

$$u_{feature_map_t} = u_{feature_map_t-1} - 0.1 \quad (6.12)$$

When the uncertainty of a feature from a map fingerprint is below a minimum threshold, than the feature is deleted, allowing in this way for dynamics in the environment.

6.3.6 Closing the Loop

One fundamental problem in SLAM is the identification of a place previously visited, if the robot returned to it. This is known as the *closing the loop* problem since the robot's trajectory loops back on itself. Thus, for topological maps, this means that if a place (i.e. a node) has been visited before, and the robot returns to it, the robot should detect it (see Figure 6.4).

Contrary to other methods used for solving this problem, based usually on the perception, loops are identified and closed with the help of the localization techniques. In order to accomplish consistency of the topological map, a method similar to the one described in [Tomatis03] is used. In this work the method employed is a non-explicit loop closing algorithm. Our loop closing method is based on the *localizer* (i.e. the POMDP). The robot is moving through the environment and incrementally builds the topological map (see Section 6.3.1.2). As soon as the robot returns in an already visited place (i.e. node) the probability distribution potentially should split up. Two candidates hypotheses should appear: one for the new place (i.e. node) currently created by the robot (e.g. in Figure 6.4, node *Q*) and another one for the previously created node already present in the map (e.g. in Figure 6.4, node *A*). As soon as the POMDP is unconfident, the algorithm tracks the two highest probability distributions showing that the distribution diverged in two peaks. A loop is thus identified if the probability distribution given by the *localizer* converges in two peaks that move in the same direction. In order to detect where the loop was closed, the two hypotheses are backtracked with localization until a single one remains.

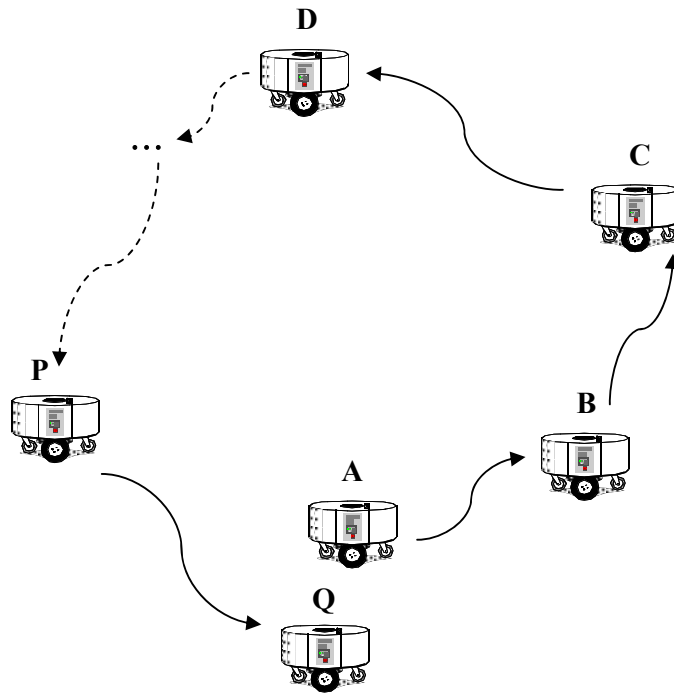


Figure 6.4: *Loop Closing Problem. The robot starts in place A and after moving through the environment arrives in place Q. The question to answer is: Has the robot returned to an already visited place or not? (i.e. Is place A equivalent to place Q?)*

6.4 Experimental Results

The approach for topological SLAM using the fingerprints of places technique was implemented and evaluated in various real world indoor and outdoor environments. In this section some of the indoor experiments carried out with the BIBA robot (see Figure 1.2(a)), and the experiments for outdoor topological mapping using the “SMART” vehicle (see Figure 1.2(b)) are presented. Both mobile platforms (indoor and

outdoor) are equipped with two 180° laser range finders and an omni-directional camera.

The first set of experiments demonstrates the robustness of the mapping module in two indoor real world scenarios and the mapping of urban outdoor environments. In particular, it illustrates the construction of distinctive and compact maps. In further experiments described in this section, the ability of the fingerprint-based localization method to globally localize the mobile robot on a map is shown. Closing the loop problem is also tested and validated through experiments.

6.4.1 Mapping Indoor Environments

The first indoor experiment was conducted in a portion of our institute building shown in Figure 6.5 and the second experiment was performed in the EPFL campus (see Figure 6.6(a)). The first test setup was the following: the robot started at the point S and ended at the point E as illustrated in Figure 6.5, the distance traveled being of 75m. For the second test the robot traveled a distance of 67m. While the robot explored the environment, it recorded, at every Δd (distance) (e.g. in our case $d = 15\text{cm}$), data readings from sensors (i.e. an image from the omni-directional camera and a scan from the laser scanner) in order to extract the fingerprints.

The robot had a *mid-line following* behavior in the hallways and *center of the free space* behavior in the open spaces. As explained in a previous chapter (see Section 5.4), it is assumed that the position in the room with the maximum free space around it is the one with the highest probability of extracting numerous and characteristic features. This ensures high distinctiveness of the observation. The map building process was performed off-line.

The threshold θ , defined as the maximum allowable dissimilarity and used for automatic mapping (see Section 6.3.1.2) is calculated experimentally. It is calculated for a small portion of the environment (i.e. 5 m), so that the map obtained matches the real structure of the environment. Once this threshold is determined, it is fixed for the rest of the indoor experiments.

6.4.1.1 Results

Figure 6.5 shows the topological map obtained by the system in the first test environment (i.e. in our laboratory), superimposed on an architectural sketch of the environment. The fingerprints used for this representation contain just the vertical edges and the corners as features. The color patches are not included because they are very sensitive to changes in illumination present in our case. The resulting map is composed of 20 nodes as shown in the Figure 6.5. Each node is represented by a mean fingerprint which is an aggregation of all the fingerprints composing the respective node. Typically, the nodes are positioned in the rooms and in the hallway. The offices are quite small and so the fingerprints that constitute the measurement points within the same room are very similar. A single node per room is thus enough. Four cases merit some additional discussion. The first special node is the one in-between Room 2 and Room 3. This node is justified because a door that connects the two hallways is



Figure 6.5: (a) Floor plan of the first environment where the experiments have been conducted. The robot starts at the point *S* and ends at the point *E*. The trajectory length is 75 m. During this experiment, the robot collected 500 data sets (i.e. images and scans) from the environment. The extracted topological map is superimposed on an architectural sketch of the environment. (b) The extracted topological map given by our method, superimposed on the raw scan map.

present. Another special node is introduced in the hallway between Room 4 and Room 5. The robot detected important changes in the environment due to the vertical pillar present in the corridor. Another node that deserves attention is the hallway node between the Room 7 and Room 8. The door of Room 8 is opened into the hallway, obstructing the view of the robot and making the environment very different in front and behind the door. A new node is therefore automatically introduced by the mapping system. The distance in the corridor between Room 8 and the end point E is quite significant. Since the robot detects distinguishing features due to the changes in this portion of the environment, a new node specifying this is required. The doors of some rooms remained closed at the time of experimentation; this explains why no node is present in front of the respective rooms (see Figure 6.5).

Figures 6.6(a) - (b) show the second test environment with the corresponding topological map, formed using the approach outlined in this work. The mapping system added a new node automatically each time a very distinctive measure (i.e. distinctive fingerprint) was encountered. The graph-like map thus obtained contains 8 nodes, as

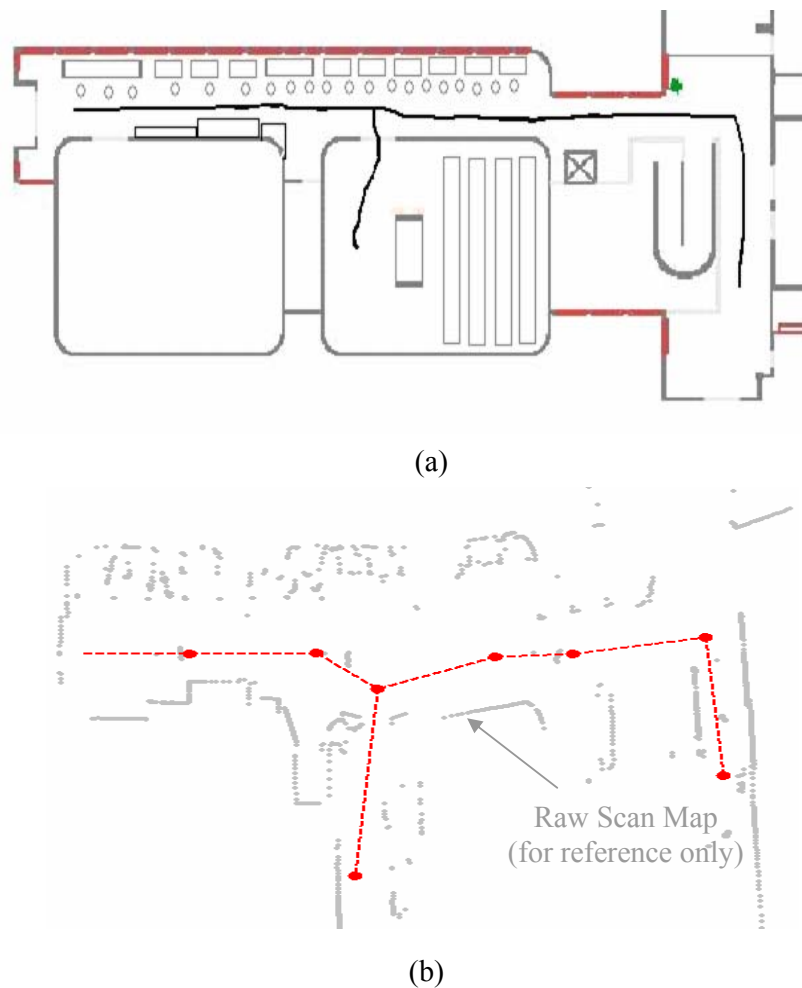


Figure 6.6: (a) The second test environment with the trajectory traveled by the robot. (b) The map of the second test environment with the graph representing the topological map.

shown in Figure 6.6(b). The same threshold used for the first test was employed here also, indicating the robustness of the overall method. The representations thus obtained (see Figure 6.5 and 6.6(b)) reproduce correctly the structure of the physical space, in a manner that is compatible with the topology of the environment. They also verify the consistency of the map and permit a distinctive modeling of it. It is important to mention that the maps are obtained by using only locally distinctive features composing the fingerprints and not by using the indoor structure of the environment (e.g. u-turn, x-crossing, etc).

6.4.2 Mapping Outdoor Environments

Compared to indoor environments, urban outdoor environments present many challenges for an autonomous vehicle. Coarse localization is often available from GPS. Most of the time, it is more useful to know the position of the robot with respect to buildings, trees, intersections, etc., than the exact latitude and longitude. In order to validate and to show the robustness of our approach, the method was also tested in an outdoor environment. The approach was tested in a part of the EPFL campus (structured environment), shown in Figure 6.7, on a 1.65 km of trajectory. The system mounted on the “SMART“ vehicle acquired data, both from the lasers and omnidirectional camera every 110 ms.

6.4.2.1 Results

A new threshold for outdoor environments was calculated experimentally in a small portion of the campus, in a similar way as for the indoor environments (see Section 6.4.1).



Figure 6.7: *The outdoor test environment (a part of the EPFL campus) with the trajectory of 1.65 km long traveled by the Smart vehicle. The magnifying glass represents the part of the environment used for the outdoor topological map exemplification (see Figure 6.8).*

Different thresholds can be used in function of the granularity of the environment that it is desired. High granularity maps, with numerous nodes, may be obtained by setting small thresholds. Alternatively, setting high values for the threshold yields maps with fewer nodes (low granularity). The outdoor threshold for obtaining high granularity maps is the same as the one used for indoor environments. For getting maps with fewer nodes, the outdoor threshold is set three times bigger than the indoor threshold. A map composed of 209 nodes for a high granularity is obtained and a map of 64 nodes containing only the big changes in the environment (i.e. intersections, new buildings, etc) is found. A small example is depicted in Figure 6.8, which represents a low granularity topological map obtained for a 200 m section of the environment (i.e. the zoomed view of the region under the magnifying glass shown in Figure 6.7). The map contains 7 nodes.

It can be noticed that the nodes are usually placed in front of buildings, at the crossings and when "big" changes occur (e.g. a building disappears from the field of view of the vehicle and driving signs, lamp-spots and trees appear).



Figure 6.8: *The low granularity outdoor topological map superimposed on an architectural sketch of a part of the EPFL campus. This part of the environment represents the zoomed of the rectangle shown in Figure 6.7.*

The map thus obtained for the entire trajectory shown in Figure 6.7 is compatible with the structure of the outdoor environment, taking into account the trees, the buildings and the lamp-posts.

6.4.3 Indoor Topological Localization with POMDP

The quality of the topological maps obtained with our fingerprint-based technique (see sections 6.4.1 and 6.4.2), can be evaluated by testing the localization on it. Localization experiments were conducted so as to show this. To test the localization, more than 1000 fingerprint sample, acquired while the robot was traveling new paths of 250 m in the indoor environments shown above (see Figure 6.5 and 6.6 (a)), were used to globally localize the robot with the POMDP. A mission is considered successful if

the place found, which corresponds to the world state with the highest probability, is the same with the correct node in the real world.

TABLE 6.1: *Summary of the indoor localization experiments.*

Fingerprints	1024 samples
Distance Traveled	250 m
Scenarios	10/10
Kidnapping	7/7
Fingerprint Matching	81%
POMDP localization	100%

The results are summarized in Table 6.1. It can be noticed that the results with the POMDP localization have given for the set of scenarios tested in this work a percentage of successful matches of 100%. The *kidnapping* problem (i.e. recovering from a lost position – the robot thinks that it is in a position where it is not) has also been tested. This was performed seven times and the robot succeeded to recover all the seven times, after one or two steps because of the very distinctive observations that corresponds to the fingerprints.

6.4.4 Closing the Loop

The localization with POMDP is used for identification of loops. As explained earlier, the robot moves through the environment and incrementally builds the topological map. The loop closing problem was tested 5 times in different situations within the environment. The robot succeeded to close the loop in all the situations. Figure 6.9 shows only a simple example that is explained below.

In Figure 6.9, it can be noticed that the robot started in the corridor, in point S. It traveled in the corridor till the door that separates the two hallways was detected (i.e. important change into the environment - node $N1$), continued in the corridor (i.e. node $N2$), then entered and went out the Room 3 (i.e. node $N3$). Once it returned in the corridor, the robot turned left and entered in an already visited place, corresponding to node $N2$. The robot temporarily creates a new node $N4$. As soon as the robot returned in an already visited place, the POMDP became unconfident and the probability distribution divided in two possible candidate states. Two hypotheses appeared: one for the new place (i.e. node $N4$ circled in red on Figure 6.9) currently created by the robot and another one for the previously created node already present in the map (i.e. node $N2$). The automatic mapper is turned off. The robot moved toward node $N1$ and labels it at node $N5$. Node $N5$ was very similar to node $N1$, and the correct match is made. A loop is thus identified if the probability distribution given by the *localizer* converges in two peaks. In order to detect where the loop is closed, the automatic mapping system is turned off and the two hypotheses are backtracked with localization until a single one remains. In the present case, this occurred when node $N5$ was detected. At that point the robot realized that node $N4$ is node $N2$ and that node $N5$ is node $N1$. Thus, the loop was closed correctly.

In order to make use of the information obtained when a place is revisited, the map is updated (see Section 6.3.5). The nodes $N1$ and $N2$ are updated with the data brought by the revisited nodes $N5$ and $N4$, respectively.

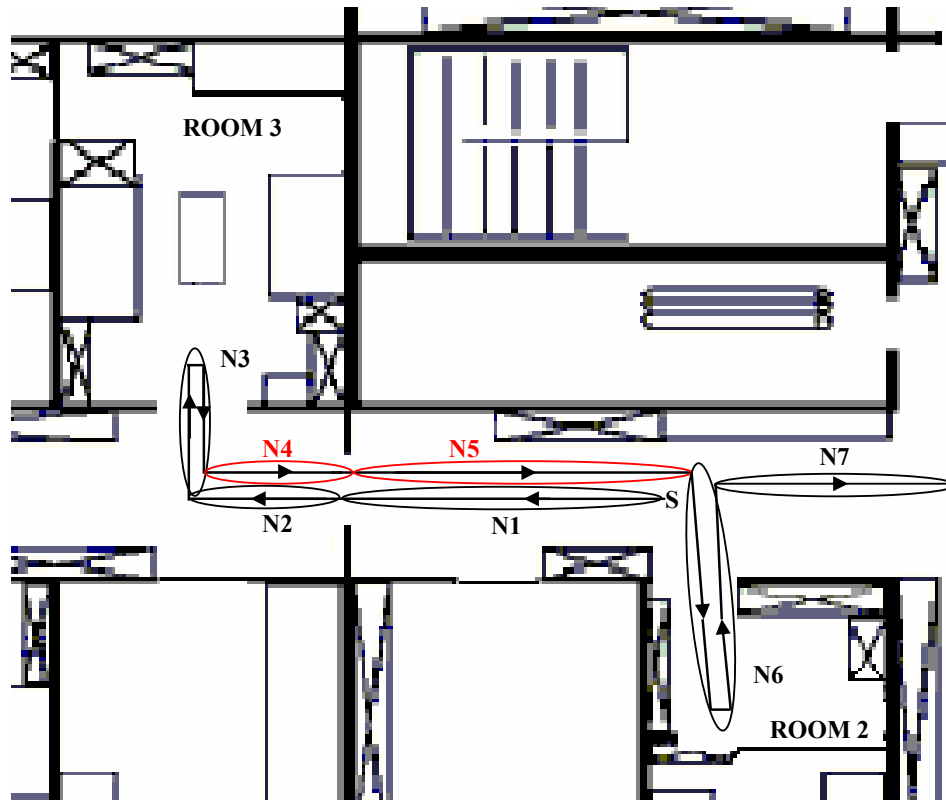


Figure 6.9: *Loop Closing:* Shows the directed path that the robot traveled. The robot starts in point S . It can be noticed that the robot arrives in a visited place (i.e. node $N4$) once it goes out the office (ROOM 3) and goes to the left, re-visiting again node $N4$.

As explained earlier, due to the fact that the offices are quite small, the fingerprints of places are very similar, and thus a single node per room is enough. Since a node contains a posterior knowledge about its environment and is the aggregation of all the fingerprints of places between the last node and the current place where an important change into the environment occurred, closing the loop problem does not appear in these cases (i.e. when one node per office is sufficient).

6.5 Discussion and Limitations

This work presents a topological SLAM system based on the fingerprints of places. As shown in previously described chapters, fingerprints of places represent the environment in a very distinctive fashion. Localization and mapping approaches are both based on this concept. It was shown that the fingerprint-based approach for mapping gives a consistent and distinctive representation of the environment. The

incremental mapping system can be seen in the same time advantageous and disadvantageous due to the heuristic used. The positive part is that once we have the heuristic (i.e. the threshold) various kinds of environments can be mapped (i.e. indoor, outdoor). The hard part is to determine the heuristic. Since in our method the heuristic is previously tuned for each type of environment, an improvement would be the automatic learning of this heuristic. An adapted methodology to do that would be reinforcement learning. The POMDP for localization shown here improves the results obtained with the approaches shown in Chapter 4. Adding the motion of the robot enables to decrease further the pose uncertainty to a level that could never be reached by fingerprint matching alone. A success rate of 100% was obtained for the tests performed in this work. However, the approach has to be extensively tested in different types of environment in order to make a real estimation of the quality of the method.

6.6 Summary

This chapter presented a new technique for topological SLAM based on the fingerprint for both localization and mapping. The fingerprint provides a compact and distinctive methodology for space representation and place recognition – it permits encoding of a huge amount of place-related information in a single circular sequence of features. This representation is suitable for both indoor and outdoor environments.

The experiments verify the efficacy and reliability of our approach. The topological maps thus obtained are very compact and distinctive. The indoor maps are compatible with the topology of the environment, verify the consistency of the map and permit a distinctive modeling of it. The outdoor maps reproduce correctly the structure of the physical space. The localization with POMDPs was also tested. The results were much better than the ones obtained in Chapter 4, as expected. The robot traveled 250m and a success rate of 100% was detected for the scenarios used in this work. The robot could also recover after kidnapping. Loop closing problem has also been tested and it was shown how this can be done at the localization level with POMDPs.

7

Conclusions

“Science never solves a problem without creating ten more.”

George Bernard Shaw (1856-1950)

This chapter is divided into three parts: a review of the major ideas and contributions composing this thesis, a study that shows the counterpart of our topological navigation system based on *fingerprints of places* in neurobiology, similarities between hippocampal place cells and the *fingerprints of places*, and finally an outlook on future works. The latter will try to highlight new directions that will enable a robot to move and act in human-centered environments.

7.1 Review and Contributions

This thesis is based on the conviction that distinctive space representation, multi-modal perception, and probabilistic SLAM, are all needed in order to obtain a robust and reliable framework for navigation. The topological navigation system designed in this dissertation is suitable for fully autonomous mobile robots, operating in structured indoor and outdoor environments.

A multi-sensory perception system is used in this work, since individual sensors suffer from robustness limitations. The resulting measurement of the world state is thus much

more reliable. The sensors used in this work include wheel encoders for odometry and two exteroceptive sensors composed of a laser range finder that give a 360° view of the environment and an omnidirectional camera.

Usually sensors yield an enormous amount of raw data and it is difficult to process this as it is. In the context of navigation, extracting significant, robust and stable features from sensory data is very important. The different features extracted from the exteroceptive sensors (i.e. corners, vertical edges, color patches, etc.) are fused and combined into a single, circular, and distinctive space representation called a *fingerprint of a place*. This representation is adapted for topological navigation and is the foundation for the whole dissertation.

The robot localization issue is a very important problem in making truly autonomous robots. Localization is the task of determining robot's position with respect to some underlying representation. Several topological localization techniques based on the fingerprint approach are presented in this work. In contrast to most of the previously presented methods, our fingerprint-based methods combine multimodal perceptual information and perform well both in indoor and outdoor environments. Localization on a fingerprint-based representation is reduced to a problem of fingerprint matching. Two of the methods make use of the *Bayesian Programming* (BP) formalism and the two others are based on *dynamic programming*.

Autonomous robot map generation warrants suitable exploration tools so as to enable the robot to discover its environment. Different behaviors and methods used to explore indoor environments are depicted. They include wall following, mid-line following, center of free space of a room, door detection, and environment structure identification.

To navigate reliably in indoor or outdoor environments a mobile robot must know where it is. For this, the robot needs to construct or to maintain a spatial representation of the environment. The main objective of this work is to enable the navigation of an autonomous mobile robot in structured environments without relying on maps a priori learned and without using artificial landmarks. A new method for incremental and automatic topological mapping and global localization with *Partially Observable Markov Decision Processes* (POMDP) using fingerprints of places is described. The mapping method presented in this dissertation uses *fingerprints of places* to create a topological model of the environment. The construction of a topological mapping system is combined with the localization technique, both relying on *fingerprints of places*, in order to perform *Simultaneous Localization and Mapping* (SLAM). This fingerprint-based approach yields a consistent and distinctive representation of the environment and is extensible in that it permits spatial cognition beyond just pure navigation.

All these methodologies have been validated through experiments. Indoor and outdoor experiments have covered more than 2 km. The *fingerprints of places* proved to provide a compact and distinctive methodology for space representation and place recognition – they permit encoding of a huge amount of place-related information in a single circular sequence of features. The indoor maps are compatible with the topology of the environment, verify the consistency of the map and permit a distinctive modeling of it. The outdoor maps reproduce correctly the structure of the physical space. The localization with POMDPs was also tested. The robot was able to localize itself all the times and it could also recover after kidnapping and closing the loops.

7.2 The Hippocampal Place Cells and the Fingerprints of Places: Spatial Representation Animals, Animats and Robots

In all our daily behaviors, the space we are living and moving in plays a crucial role. Many neurophysiologists dedicate their work to understand how our brain can create internal representations of the physical space. Both neurobiologists and roboticists are interested in understanding the animal behavior and their capacity to learn and to use their knowledge of the spatial representation in order to navigate. The ability of many animals to localize themselves and to find their way back home is linked to their mapping system.

The seminal discovery of *place cells*, by O'Keefe and Dostrovsky [O'Keefe71], in the rat hippocampus – cells whose firing pattern is dependent on the location of the animal in the environment – led to the idea that the hippocampus works as a cognitive map of space [O'Keefe78]. It was shown in [Cho98] (for a review see e.g. [Redish99]) that the lesion of the hippocampus impairs the performance of rodents in a wide variety of spatial tasks indicating a role of the hippocampus in map-based navigation.

The framework for topological SLAM (Simultaneous Localization and Mapping) proposed in this thesis organizes spatial maps in cognitive graphs, whose nodes correspond to *fingerprints of places*, and may be seen as a possible mechanism for the emergence of place cells. The computational model describes how a mobile agent can efficiently navigate in the environment, by using an internal spatial representation (similar to some extent to hippocampal place cells). This model builds a topological (qualitative) representation of the environment from the sequence of visited places. Many vision based systems for place fields using metric information have been extensively discussed in literature (e.g. [Arleo00], [Hartley00] and [Kali00] are just some of them).

It was possible to see all along this thesis that a fingerprint is associated to each distinctive place within the environment. Thus, the result given by the fingerprint matching algorithm is strongly correlated (linked) to the location of the mobile agent in the environment, giving high or the highest probability to the correct place associated to the fingerprint. The firing of place cell units can be seen as the manifestation of fingerprint matching. The closer to the center of the place field the animal is, the higher the rate of neural firing. Similarly, the nearer the new observation of the robot (i.e. the new observed fingerprint) will be with respect to the registered (learned) place (i.e. a known fingerprint), the higher the probability of the mobile agent of being in an already explored place.

The methodology presented in this thesis can efficiently create representations of places in an environment and locate the robot/animat in the environment. The place cells in the hippocampus accomplish the same task: the activation of a place cell, or perhaps better, of an assembly of place cells connected to each other, indicates that the hippocampus is locating the animal in a certain place. It can be suggested here that the hippocampus may indeed extract place from its sensory input by constructing fingerprints of places, similar to that described in this work. Indeed, in environments rich in landmarks, or features, the hippocampal cognitive map is dominated by the sensory inputs (see e.g. [O'Keefe96], [Gothard96], [Battaglia04]). Changing the relative position of landmarks can cause a complete change in place cells activity (“remapping”) so that a new set of place cells gets assigned to a given place, just as it

would be the case for our fingerprint algorithm [Cressant02]. Many theoreticians have proposed models of place cells based on visual inputs, where the visual stream is encoded in metric terms, that is, in terms of the distances between the landmarks, and between each landmarks and the agent (e.g. [Arleo00], [Hartley00] and [Kali00]). Fingerprint representations are based on the relative angular position of the landmarks from a given point of view, a much simpler and robust measure, and may be able to explain many of the experimental evidences on place cells, at least those in which multiple landmarks were available to the animal.

For the brain to perform the fingerprint matching, several building blocks are necessary: first, the identification of the landmarks, which may take place for example in the inferotemporal cortex, second, the determination of the relative position of multiple landmarks, which probably takes place in the parietal lobe ([Cressant02], [Poucet03]). The hippocampus may gather this information and produce a unitary representation (which would correspond to a fingerprint), presumably in terms of an attractor configuration of the CA3 module (which is very rich in recurrent synaptic connections and is thought to work as an attractor network module). At the moment of localization, the current input may be fed into the attractor dynamics, and, if the fingerprint matches one of the previously stored ones, the corresponding attractor is recalled. In the case of a no-match, the attractor dynamics will not produce an attractor state, and this fact may be use to signal a novel situation, and trigger the plasticity processes that allow the storage of a new memory.

This vision of hippocampal space representations highlights the role of the hippocampus as a processor of combinatorial information, whose importance transcends the purely spatial domain. In the case of space computation the hippocampus would process combinations of landmark identity and relative position information, and produce an index, which can be attached to a physical location. It is important to mention here that in our scheme the place representation does not entail any notion of Euclidean space, contrarily to what hypothesized in [O'Keefe78] and in a number of more recent works (see review in [Redish99]).

In our view, the computation of places from sensory input (through a fingerprint-like procedure), is integrated by the idiothetic information, which plays an important role especially in conditions in which only poor sensory input is available (for example, in the dark), and to disambiguate situations of perceptual aliasing (see e.g. [Skaggs98]).

The topological navigation framework based on fingerprints of places presented in this dissertation, underlies the interest of mutual inspiration between robotics, biology and neurophysiology. Our computational model finds a counterpart in neurobiology, being similar with the *hippocampus*, which plays a crucial role in spatial representation. The proposed spatial representation is an incrementally learned representation, based on fingerprints of places; the fingerprint place modeling being comparable with the place coding model in the animals (rats) hippocampus.

This study is more developed and detailed in [Tapus05c].

7.3 Open Issues

Some questions concerning the topological navigation of mobile robots have been answered by the work presented in this dissertation, but many also have been raised by

it. As *George Bernard Shaw (1856-1950)* told us “*Science never solves a problem without creating ten more.*” Therefore, some directions for future work are highlighted here.

The representation of space is a crucial point. As mentioned and demonstrated along all this work, *fingerprints of places* permit encoding a huge amount of place-related information in a single circular sequence of features. In this work, low-level features (such as vertical edges, horizontal lines) have been used. An interesting extension of the model is the addition of other modalities and features to the fingerprint framework (e.g. auditory, smell, or higher level features such as doors, table, fridge, etc.). This will help to improve the reliability and accuracy of the method and to add semantics to it. The first attempts in constructing semantics maps are depicted in [Brezetz94], [Tapus05a] and [Limketkai05].

An interesting question that may rise is: how robust is the topological navigation system presented in this work? Is the mapping system capable of coping with dynamics in the environment? The experiments depicted here have already shown that the robots can cope with variation in the world. It was also proven that the fingerprint matching algorithms can take into account small variations in the *fingerprints of places* sequences. However, more studies and experiments for the mapping system are required and an extension of the work presented in [Martinelli03a] (i.e. a multi-level SLAM) would be enriching.

The undeniable trend of research in robotics is to endow robots with the capability of understanding the world we are in, thus permitting them to help us and to be a part of our lives. An ideal companion-robot should be designed to feature sufficiently complex cognitive capabilities permitting it to understand and to interact with the environment, to exhibit social behavior, and to focus its attention and communicate with people.

This makes robotics particularly exciting, with numerous interesting problems and fascinating applications awaiting our solutions and discoveries.

Appendix A

Bayesian Programming – Basic Concepts

Since Chapter 4 and Chapter 5 make use of the Bayesian Programming formalism, a short overview of this basic concept is given here. This appendix first introduces the concepts, postulates, definitions, notations and rules that are necessary to define a Bayesian program and then addresses the Bayesian Programming concept in detail. It is interesting to show that some simple rules (basic probability theory) and the formalism presented here are sufficient to form a unifying framework for most of the probabilistic approaches found in the literature. This appendix summarizes the works described in [Bessière03], [Lebeltel04], [Mekhnacha00] and [Bellot03].

A.1 Fundamental Definitions

A.1.1 Proposition

The first concept described is the familiar notion of a *logical proposition*. Propositions are denoted by lowercase alphabets. They may be composed of propositions themselves. This is possible using the standard logical operations - conjunction, disjunction and negation. In notational terms, $a \wedge b$ and $a \vee b$ respectively represent the conjunction and disjunction of two propositions a and b . Also $\neg a$ represents the negation of a proposition a .

A.1.2 Variable

The notion of *discrete variable* is the second concept that it is required. Variables are denoted by names starting with an uppercase letter.

By definition, a *discrete variable* is a set of logical propositions such that these propositions are mutually exclusive (for all i, j with $i \neq j$, $x_i \wedge y_j$ is false) and exhaustive (at least one of the propositions x_i is true). x_i stands for «variable x takes its i^{th} value». $\lfloor X \rfloor$ denotes the cardinality of the set X (the number of propositions x_i).

The conjunction of two variables X and Y , denoted $X \wedge Y$, is defined as the set of $\lfloor X \rfloor \times \lfloor Y \rfloor$ propositions $x_i \wedge y_j$. $X \wedge Y$ is a set of mutually exclusive and exhaustive logical propositions. As such, it is a new variable. Of course, the conjunction of N variables is also a variable and, as such, it may be renamed at any time and considered as a unique variable in the sequel.

A.1.3 The probability of a proposition

To be able to deal with uncertainties, probabilities are attached to propositions.

Let consider that, to assign a probability to a proposition a , it is necessary to have at least some *preliminary knowledge*, represented by a proposition π . Consequently, the probability of a proposition a is always conditioned, at least by π . For each different π , $P(\cdot|\pi)$ is a process that assigns to each proposition a , an unique real value $P(a|\pi)$ in the interval $[0,1]$.

The same reasoning will be followed so as to calculate the probabilities of the conjunctions, disjunctions and negations of propositions, denoted, by $P(a \wedge b|\pi)$, $P(a \vee b|\pi)$, and $P(\neg a|\pi)$, respectively.

The probability of proposition a conditioned by both the preliminary knowledge π and some other proposition b is denoted by $P(a|b \wedge \pi)$.

For simplicity and clarity, probabilistic formulae use variables in place of propositions. By convention, each time a variable X appears in a probabilistic formula $\Phi(X)$, it should be understood as $\forall x_i \in X, \Phi(x_i)$. For instance, given three variables X, Y , and Z , $P(X \wedge Y|Z \wedge \pi) = P(X|\pi)$ stands for:

$$\begin{aligned} \forall x_i \in X, \forall y_j \in Y, \forall z_k \in Z \\ P(x_i \wedge y_j | z_k \wedge \pi) = P(x_i | \pi) \end{aligned} \tag{A.1}$$

A.2 Inference rules and postulates

This section contains elementary inference rules and postulates needed to carry out probabilistic reasoning. They include: the conjunction postulate, the normalization postulate, the disjunction rule, and the marginalization rule and are described below.

A.2.1 The conjunction postulate (Bayes theorem)

The probability of the conjunction of two variables X and Y can be computed according to the conjunction rule:

$$P(X \wedge Y|\pi) = P(X|\pi) \times P(Y|X \wedge \pi) = P(Y|\pi) \times P(X|Y \wedge \pi) \quad (\text{A.2})$$

This rule is also known in the form of the so called Bayes theorem:

$$P(X|Y \wedge \pi) = \frac{P(X|\pi) \times P(Y|X \wedge \pi)}{P(Y|\pi)} \quad (\text{A.3})$$

However, in this work the first form is preferred, which clearly states that it is a means of computing the probability of a conjunction of variables according to both the probabilities of these variables and their relative conditional probabilities.

A.2.2 The normalization postulate

The *normalization rule* states that the sum of the probabilities of X and $\neg X$ is one.

$$P(X|\pi) + P(\neg X|\pi) = 1 \quad (\text{A.4})$$

A.2.3 The disjunction rule

The utilization of conjunction and normalization postulates permits the derivation of the disjunction. This is states as:

$$P(X \vee Y|\pi) = P(X|\pi) + P(Y|\pi) - P(X \wedge Y|\pi) \quad (\text{A.5})$$

A.2.4 The marginalization rule

A very useful rule, called the marginalization rule, may be derived from the normalization and conjunction postulates.

This rule is given by the following formula:

$$\sum_x P(X \wedge Y|\pi) = P(Y|\pi) \quad (\text{A.6})$$

This rule can be derived as follows:

$$\sum_x P(X \wedge Y|\pi) = \sum_x P(Y|\pi) \times P(X|Y \wedge \pi) = P(Y|\pi) \times \sum_x P(X|Y \wedge \pi) = P(Y|\pi) \quad (\text{A.7})$$

\downarrow
 from
conjunction rule

\downarrow
 from
normalization rule

A.3 Bayesian Programming Formalism

In this section, the Bayesian Programming formalism is presented. Using the very simple postulates and rules described earlier, it is possible to define a generic formalism to specify probabilistic models. This generic formalism is known as the Bayesian Program (the technique is referred to as Bayesian Programming). The Bayesian Programming formalism enables the usage of a uniform notation and provides a structure to describe probabilistic knowledge and its use.

A.3.1 Structure of a Bayesian Program

The elements of a Bayesian Program are illustrated in Figure A.1. A Bayesian Program is divided in two parts: a description and a question.

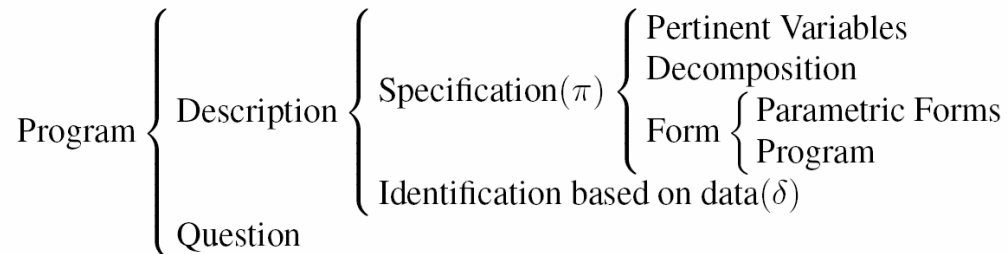


Figure A.1: *Structure of a Bayesian Program*

A program is constructed from a description, which constitutes a knowledge base (declarative part), and a question, which restitutes via inference some of this knowledge (procedural part). A description is constructed in 2 phases: a specification phase where the programmer expresses his/her preliminary knowledge and an identification (or learning) phase where the experimental data are taken into account. Preliminary knowledge is constructed from a set of pertinent variables, a decomposition of the joint distribution into a product of simpler terms, and a set of forms, one for each term. Forms are either parametric forms or questions to other Bayesian programs.

The components of the Bayesian Program are explained in detail in the following sections.

A.3.2 Description

The purpose of a description is to specify an effective method to compute a joint distribution on a set of variables V_1, V_2, \dots, V_n , given a set of experimental data δ and a preliminary knowledge π . This joint distribution is denoted as: $P(V_1 \wedge V_2 \wedge \dots \wedge V_n | \delta \wedge \pi)$.

A.3.3 Specification

The specification phase is the most critical part for the programmer. During this phase, the preliminary knowledge, that is used as input to the description part and that results from a process on specific experimental data, is expressed. Three types of preliminary knowledge can be distinguished: the set of pertinent variables, the decomposition of the joint distribution into a product of simpler terms, and a set of forms.

i. Structural Preliminary Knowledge – the choice of pertinent variables

The *structural preliminary knowledge* is the preliminary knowledge permitting the definition of a set of variables V_1, V_2, \dots, V_n for the description. All the other variables are hence assumed to be non-pertinent for the considered problem.

In robotics, these variables can be classified in three main sub-sets:

- The exteroceptive and proprioceptive sensorial variables;
- The driving variables
- The internal variables, permitting the representation of the internal states of the robot or of the sensors.

ii. Dependency Preliminary Knowledge – the choice of the decomposition of the joint distribution

As previously mentioned, the description over the set of pertinent variables V_1, V_2, \dots, V_n aims to define the joint distribution $P(V_1 \wedge V_2 \wedge \dots \wedge V_n | \delta \wedge \pi)$. This mathematical formula is a probability distribution over n dimensions. The conjunction postulate (A.2) allows for the decomposition of this joint distribution into a product of simpler terms.

Let's take a small example of a joint distribution $P(X \wedge Y \wedge Z)$ composed of three variables X, Y , and Z . By applying the conjunction postulate (A.2), this joint distribution can be written as follows:

$$P(X \wedge Y \wedge Z) = P(Z) \times P(Y|Z) \times P(X|Y \wedge Z) \quad (\text{A.8})$$

In this, second step, the specification, also permits the expression of the independence and dependence relationships between the variables. The independence relationships allow for the reduction in the dimensionality of individual terms appearing in the decomposition.

If in our previous example, the variables X and Y are supposed to be independent knowing the value of the variable Z , the expression become:

$$P(X \wedge Y \wedge Z) = P(Z) \times P(Y|Z) \times P(X|Z) \quad (\text{A.9})$$

iii. Observational Preliminary Knowledge – the choice of parametric forms

A parametric form is associated with each term appearing in the previously chosen decomposition of the joint distribution, a parametric form is associated. By initially making these choices (the steps given above), the values of the various probability distributions and the way they are modified is fixed.

The parametric forms are usually standard probability distributions (e.g. the uniform distribution and the normal distribution). A *question* to another description can also be used as a parametric form, as a probabilistic sub-program (meaning that nesting a Bayesian Program is permissible or Bayesian Program can be recursive).

A.3.4 Identification

The parametric forms can contain free parameters, like the mean or the standard deviation in a Gaussian distribution. It is necessary to give numerical values to these parameters so as to finish the description. These numerical values can be obtained through a learning process or they can be a priori fixed by the programmer.

A.3.5 Utilization

Once the specification and the identification parts are fixed, the description is complete. The utilization part consists in the application of the description in order to answer probabilistic questions.

i. Question (definition)

Asking a question consists of searching the probability distribution for a certain number of variables ε_q from the description, knowing the values of the others variables ε_c , and ignoring the values of the third type of variables ε_i . Thus, a probabilistic question has the following form:

$$P(V_k \wedge K \wedge V_l | v_m \wedge K \wedge v_n), \quad (\text{A.10})$$

where $\varepsilon_q = \{V_k, K, V_l\} \neq \emptyset$, $\varepsilon_c = \{V_m, K, V_n\} \neq \emptyset$, and $\varepsilon_i = \{V_o, K, V_p\}$ is the set of variables not included neither in the set ε_q , nor in the set ε_c . These three sets of

variables should be partitions of the set of pertinent variables in order to have a correct Bayesian Programming formulation.

ii. Inference

Any probabilistic question that has the form of the Equation (A.10) can be answered, by knowing the joint distribution $P(V_1 \wedge V_2 \wedge \mathbf{K} \wedge V_n)$ and by applying the conjunction postulate and marginalization rule.

Let's first apply the Bayes rule:

$$P(V_k \wedge \mathbf{K} \wedge V_l | v_m \wedge \mathbf{K} \wedge v_n) = \frac{P(V_k \wedge \mathbf{K} \wedge V_l \wedge v_m \wedge \mathbf{K} \wedge v_n)}{P(v_m \wedge \mathbf{K} \wedge v_n)} \quad (\text{A.11})$$

By applying the marginalization rule, the denominator and numerator can be expressed in function of the known joint distribution, as follows:

$$P(V_k \wedge \mathbf{K} \wedge V_l | v_m \wedge \mathbf{K} \wedge v_n) = \frac{\sum_{V_o, \mathbf{K}, V_p} P(V_k \wedge \mathbf{K} \wedge V_l \wedge v_m \wedge \mathbf{K} \wedge v_n \wedge V_o \wedge \mathbf{K} \wedge V_p)}{\sum_{\substack{V_k, \mathbf{K}, V_l \\ V_o, \mathbf{K}, V_p}} P(V_k \wedge \mathbf{K} \wedge V_l \wedge v_m \wedge \mathbf{K} \wedge v_n \wedge V_o \wedge \mathbf{K} \wedge V_p)} \quad (\text{A.12})$$

This method can be computationally expensive due to the normalization with respect to the variables V_o, \mathbf{K}, V_p (Bayesian inference is in general NP-complete). Therefore, different simplification forms can appear during the inference phase. These simplifications can appear due to the independencies between the variables in the joint distribution.

iii. Decision

The result of the inference provides a probability distribution over the researched variables ε_q . This probability distribution uses the a priori knowledge of the programmer on the problem.

In the field of robotics, this probability distribution can for example influence the variables directly controlling robot-motion. In order to control the robot, the values of these variables must be found. This problem takes the form of a decision-making problem. Many strategies can be conceived, but the simplest option is the choice of a value with the maximum probability or one that is chosen on the basis of a random selection performed on the distribution obtained.

A.4 Simple Example: Sensor Fusion

The following example is taken from the dissertation of Olivier Lebeltel [Lebeltel99]. It describes an experiment performed on a mobile robot, Khepera. This robot is equipped with eight infra-red sensors. The goal of this experiment is to determine the direction of the light source, in function of the output of the eight light sensors.

A.4.1 Bayesian Program Description

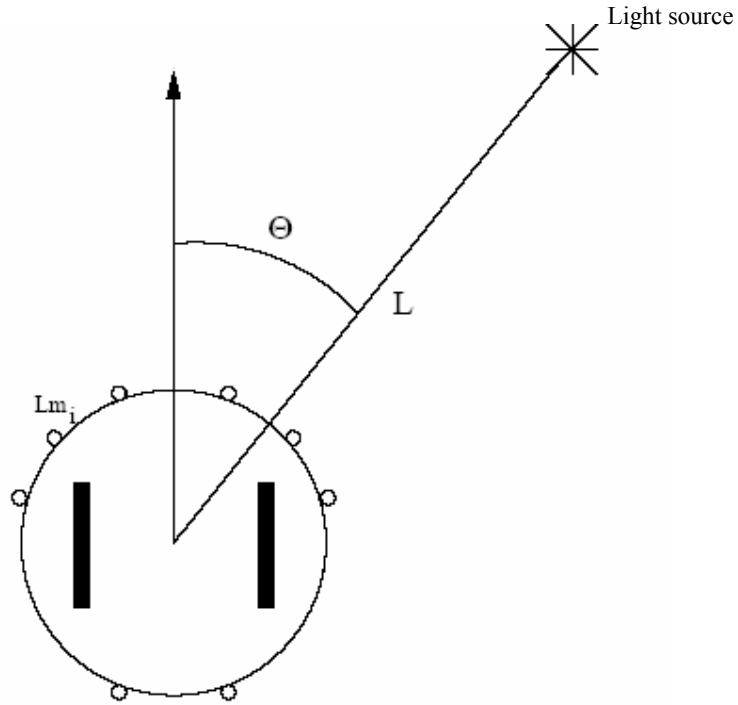


Figure A.2: Example of a simple Bayesian Program: light sensors fusion

- Pertinent Variables

The pertinent variables chosen to solve this problem are:

- The eight variables corresponding to the eight light sensors: $Lm_0, Lm_1, Lm_2, Lm_3, Lm_4, Lm_5, Lm_6$, and Lm_7 . The values of these variables are contained in the interval $[0,511]$, where 0 corresponds to strong luminosity and 511 corresponds to diminished luminosity;
- The variable characterizing the bearing of the light source: θ_L . The values of this variables are in between -180° and 180° and the step is of 10° .
- The variable designating the distance between the robot and the light source D . Its values belong to the following interval $[0,25]$ cm and the step is of 1 cm.

These pertinent variables define the probability of the joint distribution:

$$P(Lm_0 \wedge Lm_1 \wedge K \wedge Lm_7 \wedge \theta_L \wedge D). \quad (\text{A.13})$$

- Decomposition

The decomposition of the joint distribution is the job of the programmer and usually is not easy. In the case of this example, many decompositions of the distribution are possible.

The programmer usually chooses the decomposition in function of some a priori knowledge of the system. In this example, it is assumed that the output of the light sensors can be modeled as a function of the position of the light source, this can be expressed as follows: $P(Lm_i|\theta_L \wedge D)$.

The successive application of the conjunction postulate permits the joint distribution to be decomposed to the following form:

$$P(Lm_0 \wedge K \wedge Lm_7 \wedge \theta_L \wedge D) = \tag{A.14}$$

$$P(\theta_L \wedge D)P(Lm_0|\theta_L \wedge D)K P(Lm_7|Lm_0 \wedge K \wedge Lm_6 \wedge \theta_L \wedge D)$$

A strong conditional independence hypothesis is made here. It is clear that the read values of two sensors are not independent. However, it is assumed that two sensors are completely independent (i.e. no two sensory input are correlated in any way). Thus, each sensors reading is contingent only on θ_L and D . This information is incorporated into the source knowledge. Hence,

$$P(Lm_i|Lm_{i-1} \wedge K \wedge Lm_0 \wedge \theta_L \wedge D) = P(Lm_i|\theta_L \wedge D) \tag{A.15}$$

This allows us to simplify the joint distribution (see Equation A.14) as follows:

$$P(Lm_0 \wedge K \wedge Lm_7 \wedge \theta_L \wedge D) = P(\theta_L \wedge D) \prod_{i=0}^7 P(Lm_i|\theta_L \wedge D) \tag{A.16}$$

- Parametric Forms

Because no a priori knowledge is available on the light source, the probability distribution $P(\theta_L \wedge D)$ is represented by a uniform distribution.

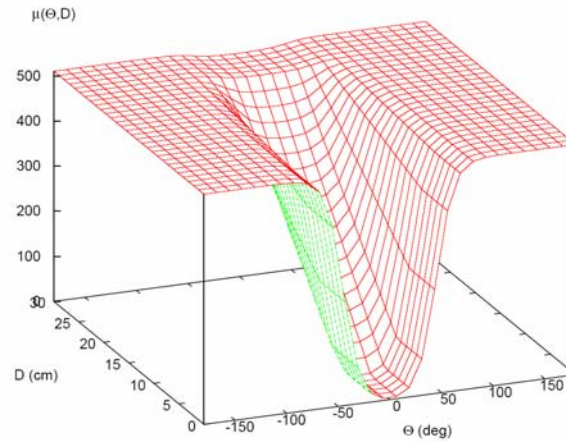


Figure A.3: *The mean value of the light sensor in function of the distance D and the bearing θ_L of the light source*

The different probabilities $P(Lm_i | \theta_L \wedge D)$ represent the output of a light sensor in function of the light source. A Gaussian distribution that depends on the light source and on the position of the sensor on the robot is chosen to represent it.

Figure A.3 depicts the function permitting to calculate the mean of these Gaussian distributions in function of the light source. In this figure, the position of the sensor on the robot is not taken into account. This function is chosen on the basis of the documentation (manuals) of the light sensors. The standard deviation is fixed at a constant value (e.g. in this case it was fixed to 20) independent of the position of the light source.

- Identification

Since the sensor model is known, there are no free parameters and therefore there is no identification step.

- Utilization

In order to estimate the bearing of the light source, knowing the observation of the eight light sensors, the following question is asked:

$$P(\theta_L | Lm_0 \wedge K \wedge Lm_7). \quad (\text{A.17})$$

For this question, the inference gives:

$$\begin{aligned}
P(\theta_L | Lm_0 \wedge K \wedge Lm_7) &= \frac{\sum_D P(Lm_0 \wedge K \wedge Lm_7 \wedge \theta_L \wedge D)}{\sum_{\theta_L, D} P(Lm_0 \wedge K \wedge Lm_7 \wedge \theta_L \wedge D)} \\
&= \frac{\sum_D \left(P(\theta_L \wedge D) \prod_{i=0}^7 P(Lm_i | \theta_L \wedge D) \right)}{\sum_{\theta_L, D} \left(P(\theta_L \wedge D) \prod_{i=0}^7 P(Lm_i | \theta_L \wedge D) \right)} \quad (A.18)
\end{aligned}$$

Since the probability distribution $P(\theta_L \wedge D)$ was represented as a uniform distribution, it is possible to simplify the above expression as:

$$\begin{aligned}
P(\theta_L | Lm_0 \wedge K \wedge Lm_7) &= \frac{\sum_D \left(\prod_{i=0}^7 P(Lm_i | \theta_L \wedge D) \right)}{\sum_{\theta_L, D} \left(\prod_{i=0}^7 P(Lm_i | \theta_L \wedge D) \right)} \quad (A.19) \\
&= \frac{1}{\alpha} \sum_D \left(\prod_{i=0}^7 P(Lm_i | \theta_L \wedge D) \right)
\end{aligned}$$

where α is the normalization constant.

In order to compare the results of the sensor fusion with the results of the measure of the bearing with a single sensor, the eight following questions are asked:

$$P(\theta_L | Lm_i), i = 0, K, 7. \quad (A.20)$$

For each of these questions the inference gives (only the example for $i=0$ is demonstrated here):

$$\begin{aligned}
P(\theta_L | Lm_0) &= \frac{\sum_{D, Lm_1, K, Lm_7} P(Lm_0 \wedge Lm_1 \wedge K \wedge Lm_7 \wedge \theta_L \wedge D)}{\sum_{D, Lm_0, K, Lm_7} P(Lm_0 \wedge Lm_1 \wedge K \wedge Lm_7 \wedge \theta_L \wedge D)} \\
&= \frac{\sum_{D, Lm_1, K, Lm_7} P(\theta_L \wedge D) \prod_{i=0}^7 P(Lm_i | \theta_L \wedge D)}{\sum_{D, Lm_0, K, Lm_7} P(\theta_L \wedge D) \prod_{i=0}^7 P(Lm_i | \theta_L \wedge D)} \quad (A.21)
\end{aligned}$$

Since we represented the probability distribution $P(\theta_L \wedge D)$ as a uniform distribution, it is possible to simplify this term. Thus, the inference result becomes:

$$P(\theta_L | Lm_0) = \frac{\sum_D \left(P(Lm_0 | \theta_L \wedge D) \sum_{Lm_1, K, Lm_7} \prod_{i=1}^7 P(Lm_i | \theta_L \wedge D) \right)}{\sum_{D, Lm_0} \left(P(Lm_0 | \theta_L \wedge D) \sum_{Lm_1, K, Lm_7} \prod_{i=1}^7 P(Lm_i | \theta_L \wedge D) \right)} \quad (\text{A.22})$$

The final inference result is:

$$P(\theta_L | Lm_0) = \frac{\sum_D P(Lm_0 | \theta_L \wedge D)}{\sum_{D, Lm_0} P(Lm_0 | \theta_L \wedge D)} = \frac{1}{\alpha_0} \sum_D P(Lm_0 | \theta_L \wedge D) \quad (\text{A.23})$$

where α_0 is the normalization constant.

A.4.2 Results and Discussion

Figure A.4 describes the results for the light source when the bearing angle $\theta_L = 10^\circ$. The peripheral diagrams presents the results obtained by using a single sensor, the answers at the questions expressed in Equation (A.23). The central diagram describes the result of the sensors fusion, the distribution $P(\theta_L | Lm_0 \wedge K \wedge Lm_7)$.

The information given by each of the eight sensors is very weak. The result of the probabilistic fusion of the eight light sensors is great, it delivers a Dirac on the 10° position of the bearing angle.

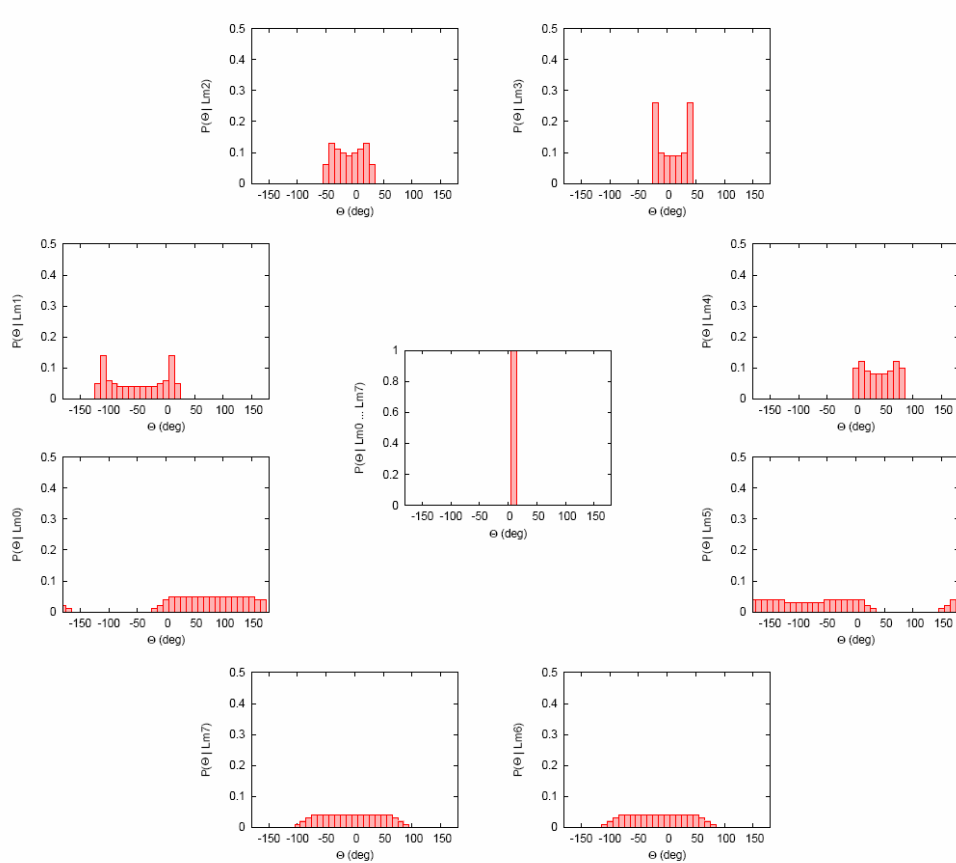


Figure A.4: The result of a sensor fusion for a light source with a bearing of 10°

A.5 Summary

The Bayesian Programming (BP) approach was originally proposed as a tool for robot programming (see [Lebeltel04]), but nowadays it is used in a wider range of applications ([Mekhnacha00] shows some examples).

The Bayesian Programming and the probabilistic reasoning in general took a new dimension with the BIBA European project. One of the major goals of this project was the demonstration of the biological plausibility of the probabilistic inference at the microscopic level as well as macroscopic one.

Appendix B

Glossary

agent	an entity or a computational process that senses its world and acts in it
allothetic source	allothetic sources of information provide external information about the environment.
animat	a robot that executes a mechanism described in the biological literature; in other words a robot that has a bio-mimetic behavior
Bayesian Programming	a generic formalism that enables the usage of a uniform notation and provides a structure to describe probabilistic knowledge and its use
behavior	a control law that achieves and/or maintains some goal
catadioptric system	employs both mirrors and lenses in their optics for image formation; are usually used to provide a far wider field of view than is easily possible using lenses or mirrors alone.
catoptric system	makes only use of mirrors for image formation.
cerebral cortex	is the outer layer of grey matter; is made up of neurons and supporting cells (glial cells) and functions to correlate information from many sources to

	maintain cognitive function (all aspects of perceiving, thinking and remembering).
cognitive graph	topological representation
dioptric system	system that uses only lenses for image formation
distinctiveness	a utter dissimilarity
dynamic programming	solves an optimization problem by caching sub-problem solutions rather than recomputing them.
equiangular mirror	a mirror in which each pixel spans an equal angle irrespective of its distance from the center of the image
exteroceptive sensor	measures environment features such as distance, color and luminosity.
fingerprint of a place	is a circular list of features that the robot can perceive around it
firing of place cells	corresponds to a matching mechanism; the place cells fires maximally in a place field.
hippocampus	a complex neural structure consisting of gray matter; has a central role in the formation of memories and processing of spatial information
hybrid map	is an integration of both the metric and topological space representation paradigms.
idiothetic	idiothetic source yields internal information about the mobile agent's movements (e.g. speed, acceleration, etc.).
inferotemporal cortex	is a region of the cerebral cortex
localization	is the task of determining robot's position with respect to some underlying representation
loop closing	is the identification of a place previously visited, if the robot returns to it; this is known as the <i>closing the loop</i> problem since the robot's trajectory loops back on itself.
map	representation

metric map	a map of the environment useful when it is necessary for the robot to know its location accurately in terms of metric coordinates (i.e. Cartesian coordinates).
neurobiology	the biological study of the nervous system or any part of it.
neurophysiology	the branch of physiology (i.e. the biological study of the functions of living organisms and their parts) that deals with the functions of the nervous system
perceptual aliasing	distinct locations within the environment appearing identical to the robot's sensors
place cell	units whose firing pattern is dependent on the animal location in the environment; gives give a model of how spatial information is encoded at the neural level.
place field	is the part of the environment where a place cell fires or fires maximally
proprioceptive sensor	measures and monitors the internal state of a robot (e.g. motor speed, temperature, acceleration).
reactive behavior	a control law that relies on the immediacy of sensory information, linked directly to motor behavior without the use of any symbolic representation
semantic map	is a map that includes elements that give a meaningful representation; it consists of the relationship between different elements and what they represent.
SLAM	Simultaneous Localization and Mapping – is the construction of maps while the robot moves through the environment and the localization with respect to the partially built maps
topological map	a qualitative representation of the environment; it can be viewed as a graph of places, where at each node the information concerning the visible landmarks and the way to reach other places, connected to it, is stored

Bibliography

- [Abe99] Abe, Y., Shikano, M., Fukuda, T., Tanaka, Y. (1999). Vision Based Navigation System for Autonomous Mobile Robot with Global Matching. IEEE International Conference on Robotics and Automation (ICRA), Detroit, Michigan, USA.
- [Aho90] Aho, A., V. (1990). Algorithms for finding patterns in strings, Elsevier Science Publishers B. V.: 254-300.
- [Althaus03] Althaus, P. (2003). Indoor Navigation for Mobile Robots: Control and Representations. Ph.D Thesis, Numerical Analysis and Computer Science. Stockholm, Royal Institute of Technology.
- [Arleo00] Arleo, A., Gerstner, W. (2000). "Spatial Cognition and Neuro-Mimetic Navigation: a model of Hippocampal Place Cell Activity." Biological Cybernetics **83**: 287-299.
- [Arras97] Arras, K., O., Siegwart, R. (1997). Feature Extraction and Scene Interpretation for Map-based Navigation and Map Building. In Proceedings of the Symposium on Intelligent Systems and Advanced Manufacturing.
- [Arras00] Arras, K., O., Tomatis, N., Siegwart, R. (2000). Multisensor On-the-Fly Localization using Laser and Vision. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Takamatsu, Japan.
- [Arras03] 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 **1056**: 1-13.

- [Aycard97] Aycard, O., Charpillat, F., Fohr, D., Mari, J., F. (1997). Place learning and recognition using Hidden Markov Models. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Grenoble, France.
- [Baeza-Yates99] Baeza-Yates, R., Navarro, G. (1999). Faster Approximate String Matching, *Algorithmica*, 23(2): 127-158
- [Battaglia04] Battaglia, F., P., Sutherland, G., R., McNaughton, B., L. (2004). "Local sensory cues and place cell directionality: additional evidence of prospective coding in the hippocampus." Journal of Neuroscience **24**: 4541-4550.
- [Beeson05] Beeson, P, Jong, N., K., Kuipers, B. (2005). Towards autonomous topological place detection using the Extended Voronoi Graph. IEEE International Conference on Robotics and Automaton (ICRA), Barcelona, Spain.
- [Bellot03] Bellot, D., Siegwart, R., Bessière, P., Tapus, A., Coué C., Diard, J. (2003). "Bayesian Reasoning for Real World Robotics: Basics, Scaling and Example." Dagstuhl Proceedings on Embodied AI, LNCS (Lecture Note in Computer Science), Springer.
- [Berthoz97] Berthoz, A. (1997). Le sens du mouvement. E. O. Jacob. Paris.
- [Bessière03] Bessière, P., et al. (2003). Survey: Probabilistic Methodology and Techniques for Artefact Conception and Development, Technical Report INRIA, RR-4730, European IST BIBA project, 2003.
- [Bilmes97] Bilmes, J. A. (1997). A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models, ICSI-TR-97-021.
- [Borenstein94] Borenstein, J., Feng, L. (1994). UMBmark - A method for measuring, comparing and correcting dead-reckoning errors in mobile robots, Technical Report UM-MEAM-94-22, University of Michigan.
- [Borenstein96] Borenstein, J., Everett, H., R., Feng, L. (1996). Where Am I? Sensors and Methods for Mobile Robot Positioning. Ann. Arbor, University of Michigan.
- [Borges04] Borges, G., A., Aldon, M.,J. (2004). "Line Extraction in 2D Range Images for Mobile Robotics." Journal of Intelligent and Robotic Systems **40**: 267-287.
- [Brezetz94] Brezetz, S., B., Chatila, R., Devy, M. (1994). Natural scene understanding for mobile robot navigation. IEEE International Conference on Robotics and Automation (ICRA), San Diego, USA.

- [Canny86] Canny, J., F. (1986). "A Computational Approach to Edge Detection." In IEEE Transactions on Pattern Analysis and Machine Intelligence **6**(8): 679-698.
- [Cassandra96] Cassandra, A., R., Kaelbling, L. P., Kurien, J., A. (1996). Acting under Uncertainty: Discrete Bayesian Models for Mobile-Robot Navigation. IEEE International Conference on Robotics and Automation (ICRA), Osaka, Japan.
- [Castellanos96] Castellanos, J., A., J.,D.,Tardos (1996). Laser-Based Segmentation and Localization for a Mobile Robot. 6th International Symposium on Robotics and Manufacturing (ISRAM96), Montpellier, France.
- [Castellanos99] Castellanos, J., A., Tardos, J., D. (1999). Mobile Robot Localization and Map Building: Multisensor Fusion Approach. Kluwer.
- [Chahl97] Chahl, J., S., Srinivasan, M., V. (1997). "Reflective surfaces for panoramic imaging." Applied Optics.
- [Cho98] Cho, Y., H., Giese, K., P., Tanila, H., T., Silva, A., J., Eichenbaum, H. (1998). "Abnormal hippocampal spatial representations in alphaCaMKII^{T286A} and CREB^{alphaDelta}- mice." Science **279**: 867-869.
- [Chong97] Chong, K., S., Kleeman, L. (1997). Accurate Odometry and Error Modelling for a Mobile Robot. IEEE International Conference on Robotics and Automation (ICRA), New Mexico, USA.
- [Choset01] Choset, H., Nagatani, K. (2001). "Topological Simultaneous Localization and Mapping (SLAM): Toward Exact Localization Without Explicit Localization." IEEE Transactions On Robotics and Automation **17**(2): 125-137.
- [Cressant02] Cressant, A., Muller, R., U., Poucet, B. (2002). "Remapping of place cells firing patterns after maze rotations." Journal on Experiences on Brain Research **143**: 470-479.
- [Dellaert99] Dellaert, F., Fox., D., Burgard, B., Thrun, S. (1999). Monte Carlo Localization for Mobile Robots. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyongju, Korea.
- [Diard03] Diard, J. (2003). La carte bayésienne, un modèle probabiliste hiérarchique pour la navigation en robotique mobile. Ph.D Thesis. INPG (Institut National Polytechnique de Grenoble). Grenoble.
- [Dissanayake01] Dissanayake, M., W., Newman, M., G., P., Clark, S., Durrant-Whyte, H., Csorba, M. (2001). "A Solution to the Simultaneous Localization and Map Building (SLAM) problem." IEEE Transactions on Robotics and Automation **17**(3): 229-241.

- [Douglas73] Douglas, D., Peucker, T. (1973). "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature." The Canadian Cartographer **10**(2): 112-122.
- [Dufourd04] Dufourd, D., Chatila, R., Luzeaux, D. (2004). Combinatorial maps for simultaneous localization and map buiding (SLAM). IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sendai, Japan.
- [Fischler81] Fischler, M., Bolles., R. (1981). "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography." Communications of the ACM **381**(24).
- [Fox98] Fox, D. (1998). Markov Localization: A Probabilistic Framework for Mobile Robot Localization and Navigation. Ph.D Thesis, Institute of Computer Science III. Bonn, Germany, University of Bonn.
- [Franz98] Franz, M., Schölkopf, O., Mallot, B., Bühlhoff, H., A. (1998). "Learning view graphs for robot navigation." Autonomous Robots **5**: 111-125.
- [Gallistel90] Gallistel, R. (1990). The Organization of Learning. M. Press. Cambridge, MA.
- [Gothard96] Gothard, K., M., Skaggs, W., E., Moore, K., M., McNaughton, B., L. (1996). "Binding of hippocampal CA1 neural activity to multiple reference frames in a landmark-based navigation task." Journal of Neuroscience **16**: 823-835.
- [Gutmann98] Gutmann, J., S., Burgard, W., Fox, D., Konolige, K. (1998). An Experimental Comparison of Localization Methods. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Victoria, Canada.
- [Hafner00] Hafner, V., V. (2000). Learning Places in Newly Explored Environments. S. P. S. Book. Honolulu, Publication of the International Society for Adaptive Behavior.
- [Hartley00] Hartley, T., Burgess, N., Lever, C., Cacucci, F., O'Keefe, J. (2000). "Modeling place fields in terms of the cortical inputs to the hippocampus." Hippocampus **10**: 369-379.
- [Herman02] Herman, H., Singh, S. (2002). Panoramic Mirror and System for Producing Enhanced Panoramic Images, US Patent, 6856-472.
- [Hough62] Hough, P., V.,C. (1962). A Method and Means for Recognizing Complex Patterns. US Patent, 3069654.

- [Jain95] Jain, R., Kasturi, R., Schunck, B., G. (1995). Machine Vision, McGraw-Hill.
- [Jensen04] Jensen, B. (2004). Motion Tracking for Human-Robot Interaction. PhD Thesis, Lausanne, Ecole Polytechnique Fédérale de Lausanne (EPFL).
- [Jensfelt99] Jensfelt, P., Christensen, H. (1999). Laser Based Position Acquisition and Tracking in an Indoor Environment. IEEE International Conference on Robotics and Automation (ICRA), Detroit, Michigan, USA.
- [Kali00] Kali, S., Dayan, P. (2000). "The involvement of recurrent connections in area CA3 in establishing the properties of place fields: a model." Journal of Neuroscience **20**: 7463-7477.
- [Kalman60] Kalman, R., E. (1960). "A new approach to linear filtering and prediction problems." Journal of Basic Engineering.
- [Kanade85] Kanade, T., Ohta, Y. (1985). "Stereo by Intra- and Inter-Scanline Search Using Dynamic Programming." IEEE Transactions on Pattern Analysis and Machine Intelligence **Vol PALMZ(3)**.
- [Kortenkamp94] Kortenkamp, D., Weymouth, T. (1994). Topological mapping for mobile robots using a combination of sonar and vision sensing. American Association for Artificial Intelligence (AAAI), Seattle, WA, USA.
- [Kuipers78] Kuipers, B., J. (1978). "Modeling Spatial Knowledge." Cognitive Science **2**(129-153).
- [Kuipers91] Kuipers, B., J., Byun, Y.,T. (1991). "A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations." Journal of Robotics and Autonomous Systems **8**: 47-63.
- [Lamon01] Lamon, P., Nourbakhsh, I., Jensen, B., Siegwart, R. (2001). Deriving and Matching Image Fingerprint Sequences for Mobile Robot Localization. IEEE International Conference on Robotics and Automation (ICRA), Seoul, Korea.
- [Lamon03] Lamon, P., Tapus., A., Glauser, E., Tomatis, N., Siegwart, R. (2003). Environmental Modeling with Fingerprint Sequences for Topological Global Localization. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, USA.
- [Lebeltel04] Lebeltel, O., Bessière, P., Diard, J., Mazer, E. (2003). "Bayesian Robot Programming, In Autonomous Robots." Autonomous Robots, **16(1)**: 49-79.

- [Lebeltel99] Lebeltel, O. (1999). Programmation Bayésienne des Robots. PhD Thesis, INPG. Grenoble.
- [Leonard92] Leonard, J., J., Durrant-Whyte, H.,F. (1992). Directed Sonar Sensing for Mobile Robot Navigation. Kluwer Academic Publisher. Dordrecht.
- [Limketkai05] Limketkai, B., Fox, D., Liao, L. (2005). Relational Object Maps for Mobile Robots. 19th International Joint Conference on Artificial Intelligence (IJCAI), Edinburgh, Scotland.
- [Lisien03] Lisien, B., et al. (2003). Hierarchical Simultaneous Localization and Mapping. IEEE/RSJ International Conference on Intelligent Robot and Systems (IROS), Las Vegas, USA.
- [Martinelli03a] Martinelli, A., Tapus, A., Arras, K., O., Siegwart, A. (2003). Multi-resolution SLAM for Real World Navigation. 11th International Symposium of Robotics Research (ISRR), Siena, Italy.
- [Martinelli03b] Martinelli, A., Tomatis, N., Tapus, A., Siegwart, A. (2003). Simultaneous Localization and Odometry Calibration for Mobile Robot. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, USA.
- [Mataric90] Mataric, M., J. (1990). Navigating with a rat brain: A neurobiologically-inspired model for robot spatial representation. In: From Animals to Animats: First International Conference on Simulation of Adaptive Behavior (SAB) J.A.Meyer, S.W.Wilson (Eds), MIT Press, Cambridge, MA, USA, pp. 432-441.
- [Mekhnacha00] Mekhnacha, K., Mazer, E., Bessière, P. (2000). A robotic CAD system using a Bayesian framework. IEEE-RSJ International Conference on Intelligent Robots and Systems (IROS), Takamatsu, Japan.
- [Moutarlier89] Moutarlier, P., Chatila, R. (1989). Stochastic Multisensory Data Fusion for Mobile Robot Location and Environment Modeling. 5th International Symposium on Robotics Research (ISRR), Tokyo, Japan.
- [Needleman70] Needleman, S., Wunsch, C. (1970). "A general method applicable to the search for similarities in the amino acid sequence of two proteins." Journal on Molecular Biology **48**: 443-453.
- [Nguyen05] Nguyen, V., Martinelli, A., Tomatis, N., Siegwart, R. (2005). A Comparison of Line Extraction Algorithms using 2D Laser Rangefinder for Indoor Mobile Robotics. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Edmonton, Canada.

- [O'Keefe71] O'Keefe, J., Dostrovsky, J. (1971). "The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat." Journal of Brain Research **34**: 171-175.
- [O'Keefe78] O'Keefe, J., Nadel, L. (1978). *The hippocampus as a cognitive map*. Clarendon. Oxford.
- [O'Keefe96] O'Keefe, J., Burgess, N. (1996). "Geometric determinants of the place fields of hippocampal neurons." Nature **381**: 425-428.
- [Ollis99] Ollis, M., Hermann, H., Singh, S. (1999). *Analysis and Design of Panoramic Stereo-Vision Using Equi-Angular Pixel Cameras*, Technical Report, CMU-RI-TR-99-04.
- [Owen98] Owen, C., Nehmzow, U. (1998). Landmark-based navigation for a mobile robot, In : Meyer, Berthoz, Floreano, Roitblat and Wilson (Eds.), *From Animals to Animate 5*, Proceedings of SAB'98, MIT Press, Cambridge, MA: 240-245.
- [Perez99] Perez, J., A., Castellanos, J., A., Montiel, M., M., Neira, J., Tardos, J., D. (1999). Continuous Mobile Robot Localization: Vision vs. Laser. IEEE International Conference on Robotics and Automation (ICRA), Detroit,, Michigan, USA.
- [Pfister03] Pfister, S., T., Kreichbaum, K., L., Roumeliotis, S., I., Burdick, J., W. (2003). Weighted Range Sensor Matching Algorithms for Mobile Robot Displacement Estimation. IEEE International Conference on Robotics and Automation (ICRA), Washington D.C., USA.
- [Poucet03] Poucet, B., Lenck-Santini, P., P., Paz-Villagran, V., Save, E. (2003). "Place cells, neocortex and spatial navigation: a short review." Journal of Physiology Paris **97**: 537-546.
- [Redish99] Redish, A., D. (1999). *Beyond the Cognitive Map: From Place Cells to Episodic Memory*. Cambridge, MA, MIT Press.
- [Remolina97] Remolina, E. (1997). *A Kalman Filter for Wall Following*.
- [Se02] Se S., Lowe, D., Little., J. (2002). "Mobile Robot Localization and Mapping with uncertainty using Scale-Invariant Visual Landmarks." The International Journal of Robotics Research **21**(8): 735-758.
- [Siadat97] Siadat, A., Kaske,A., Klausmann, S., Dufaut, M, Husson, R. (1997). An Optimized Segmentation Method for a 2D Laser-Scanner. Applied to Mobile Robot Navigation. In Proceedings of the 3rd IFAC Symposium on Intelligent Components and Instruments for Control Applications.

- [SICK00] AG, S. (2000). LMS2xx- Laser Measurement Systems, Technical Description - SICK AG.
- [Siegwart04] Siegwart, R., Nourbakhsh, I, R. (2004). Introduction to Autonomous Mobile Robots. The MIT Press. A. B. Book. Cambridge, Massachusetts.
- [Skaggs98] Skaggs, W., E., McNaughton, B.,L. (1998). "Spatial firing properties of hippocampal CA1 populations in an environment containing two visually identical regions." Journal of Neuroscience **18**: 8455-8466.
- [Smith86] Smith, R., C., Cheeseman, P. (1986). "On the Representation and Estimation of Spatial Uncertainty." International Journal of Robotics Research **5**(4): 56-68.
- [Tapus04a] Tapus, A., Heinzer, S., Siegwart, R. (2004). Bayesian Programming for Topological Global Localization with Fingerprints. IEEE International Conference on Robotics and Automation (ICRA), New Orleans, USA.
- [Tapus04b] Tapus, A., Tomatis, N., Siegwart, R. (2004). Topological Global Localization and Mapping with Fingerprint and Uncertainty. International Symposium on Experimental Robotics (ISER) Singapore, Singapore.
- [Tapus04c] Tapus, A., Ramel, G., Dobler, L., Siegwart, R. (2004). Topology Learning and Place Recognition using Bayesian Programming for Mobile Robot Navigation. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sendai, Japan.
- [Tapus05a] Tapus, A., Vasudevan, S., Siegwart, R. (2005). Toward a Multilevel Cognitive Probabilistic Representation of Space. International Conference on Human Vision and Electronic Imaging X (HVEI), part of the IS&T/SPIE Symposium on Electronic Imaging 2005, San Jose, CA, USA.
- [Tapus05b] Tapus, A., Siegwart, R. (2005). Incremental Robot Mapping with Fingerprints of Places. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Edmonton, Canada.
- [Tapus05c] Tapus, A., Battaglia, F., Siegwart, R. (2005). The Hippocampal Place Cells and the Fingerprints of Places: Spatial Representation Animals, Animats and Robots. 6th Symposium on Intelligent Autonomous Vehicles (IAV) (submitted), Tokyo, Japan.
- [Thrun98] Thrun, S. (1998). "Learning metric-topological maps for indoor mobile robot navigation." Artificial Intelligence **99**(1): 21-71.
- [Thrun00] Thrun, S. (2000). Probabilistic algorithms in robotics. Artificial Intelligence Magazine. **21**: 93-109.

- [Thrun01] Thrun, S., Fox, D., Burgard, W., Dellaert, F. (2001). "Robust Monte Carlo Localization for Mobile Robots." Artificial Intelligence **128**(1-2): 99-141.
- [Tolman48] Tolman, E., C. (1948). "Cognitive maps in rats and men." Psychological Review **55**: 189-208.
- [Tomatis03] Tomatis, N., Nourbakhsh, I., Siegwart, R. (2003). "Hybrid Simultaneous Localization and Map Building: a Natural Integration of Topological and Metric." Robotics and Autonomous Systems **44**: 3-14.
- [Ulrich00] Ulrich, I., Nourbakhsh, I., R. (2000). Appearance-Based Place Recognition for Topological Localization. IEEE International Conference on Robotics and Automation (ICRA), San Francisco, CA, USA.
- [vanTurenout92] van Turenout, P., Honderd, G., van Schelven, L., J. (1992). Wall-following control of a mobile robot. IEEE International Conference on Robotics and Automation (ICRA), Ottawa, Canada.

Personal Publications

Journals

Tapus, A., Siegwart, R. (2005). **Topological Simultaneous Localization and Mapping with Fingerprints of Places.** (submitted).

Book Chapters

Bellot, D., Siegwart, R., Bessière, P., Tapus, A., Coué, C. and Diard, J. (2003) **Bayesian Reasoning for Real World Robotics: Basics, Scaling and Example.** In Fumiya Iida and Rolf Pfeifer and Luc Steels and Yasuo Kuniyoshi, editors, Embodied Artificial Intelligence, Lecture Notes in Computer Science Series, volume 3139, pages 186-201. Springer-Verlag 2003.

Peer-reviewed Proceedings

Tapus, A., Siegwart, R. (2006) **Fingerprints of Places: A Cognitive Modeling of Space for Mobile Robot Navigation.** The 1st Annual Conference on Human Robot Interaction (HRI) (submitted), Salt Lake City, Utah, USA.

Tapus, A., Battaglia, F., Siegwart, R. (2006). **The Hippocampal Place Cells and the Fingerprints of Places: Spatial Representation Animals, Animats and Robots.** The 9th International Conference on Intelligent Autonomous Systems (IAS) (submitted), Tokyo, Japan.

Tapus, A., Siegwart, R. (2005). **Incremental Robot Mapping with Fingerprints of Places.** IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Edmonton, Canada.

Tapus, A., Vasudevan, S., Siegwart, R. (2005). **Toward a Multilevel Cognitive Probabilistic Representation of Space.** International Conference on Human Vision and Electronic Imaging X (HVEI), part of the IS&T/SPIE Symposium on Electronic Imaging 2005, San Jose, CA, USA.

Tapus, A., Ramel, G., Dobler, L., Siegwart, R. (2004). **Topology Learning and Place Recognition using Bayesian Programming for Mobile Robot Navigation**. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sendai, Japan.

Tapus, A., Tomatis, N., Siegwart, R. (2004). **Topological Global Localization and Mapping with Fingerprint and Uncertainty**. International Symposium on Experimental Robotics (ISER) Singapore, Singapore.

Tapus, A., Heinzer, S., Siegwart, R. (2004). **Bayesian Programming for Topological Global Localization with Fingerprints**. IEEE International Conference on Robotics and Automation (ICRA), New Orleans, USA.

Martinelli, A., Tomatis, N., Tapus, A., Siegwart, A. (2003). **Simultaneous Localization and Odometry Calibration for Mobile Robot**. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, USA.

Lamon, P., Tapus, A., Glauser, E., Tomatis, N., Siegwart, R. (2003). **Environmental Modeling with Fingerprint Sequences for Topological Global Localization**. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, USA.

Martinelli, A., Tapus, A., Arras, K., O., Siegwart, A. (2003). **Multi-resolution SLAM for Real World Navigation**. 11th International Symposium of Robotics Research (ISRR), Siena, Italy.

Tapus, A. and Aycard, O. (2003) **Searching a Target with a Mobile Robot**. In Proceedings of the 14th International Conference on Control Systems and Computer Science, Bucharest, Romania, July 2003.

Technical Reports

Tapus, A. (2002) **Markov Decision Processes Use for Mobile Robot Actions Planning and Execution**. Masters Degree Report in Computer Science, Joseph Fourier University, Grenoble, France.

Curriculum Vitae

Born on 11th February 1980, I grew up in Bucharest, Romania. In 1998, I graduated from the “Computer Science Tudor Vianu” high school. After one year of studies in Computer Science in Politechnica University of Bucharest (PUB), I received an ERASMUS Fellowship to continue my studies at the University Joseph Fourier, Grenoble, France. I continued in parallel at both universities, in France and in Romania and completed my studies, both in Computer Science and Applied Mathematics. I received my Masters in Computer Science from University Joseph Fourier, Grenoble, France, with a diploma thesis titled “Using Markov Decision Processes (MDP) for Mobile Robot Actions Planning and Execution”. Following this, I joined the Autonomous Systems Lab at Ecole Polytechnique Fédérale de Lausanne (EPFL) to work on my Ph.D thesis titled “Topological SLAM with Fingerprints of Places”, under the supervision of Prof. Roland Siegwart. Between 2001 and 2005 I have been a member of the Brain Inference Bayesian and Artifacts (BIBA) European project. My doctoral thesis has been completed in the context of this project.