FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

FEUP

A collaborative, non-invasive Hybrid Semantic Localization And Mapping system (HySeLAM).

Filipe Baptista Neves dos Santos

Programa de Doutoramento em Engenharia Electrotécnica e de Computadores

Supervisor: Paulo José Cerqueira Gomes da Costa (Professor Doutor)

Second Supervisor: António Paulo Gomes Mendes Moreira (Professor Doutor)

March 12, 2014

A collaborative, non-invasive Hybrid Semantic Localization And Mapping system (HySeLAM).

Filipe Baptista Neves dos Santos

Programa de Doutoramento em Engenharia Electrotécnica e de Computadores

Approved by . . . :

President: .....
Referee: .....
Referee: .....

March 12, 2014

Resumo

A robótica móvel que pressupõe interação com o humano requer novos paradigmas do conceito de inteligência. Tal passará pela capacidade de o robô entender e executar tarefas, como: "Robô, vai ao escritório e traz-me a caneta que está em cima da mesa." A maioria das abordagens ao projeto da camada de localização e mapeamento não é compatível com camadas superiores para a resolução deste tipo de tarefas.

O sucesso desta cooperação, robô/homem, requer do robô a capacidade de racionar acerca do seu conhecimento espacial e das suas próprias observações, e a capacidade de adquirir e partilhar o conhecimento espacial durante uma interação verbal. Ao contrário da maioria das abordagens ao problema do SLAM (Simultaneous Localization and Mapping), que não interpretam as observações para além do nível geométrico, é necessário definir uma nova representação do conhecimento espacial. Representação essa similar à dos humanos.

Este trabalho visa dar uma resposta à pergunta *Como deve ser construída a arquitetura do robô, a partir do SLAM convencional, por forma a incluir um humano no processo de mapeamento?*. Para esta resposta, foi explorada e investigada uma abordagem híbrida (métrica e semântica) para a camada de localização e mapeamento, que contempla uma estrutura de conhecimento similar à dos humanos. Esta abordagem permitirá ao robô completar e construir o seu conhecimento através de uma interação com o humano, associar conjuntos de características das observações a palavras associadas a objetos ou locais, bem como formalizar um raciocínio, ao nível semântico, sobre este conhecimento e as próprias observações.

Neste trabalho, é proposta uma nova extensão ao SLAM (Localização e Mapeamento em simultâneo) com uma estrutura compatível com o mapeamento topológico e semântico. À representação do mapa geométrico (mapa de ocupação em grelha) são acrescentados dois novos mapas, mapa topológico e mapa semântico. O nome desta extensão é HySeLAM, que deriva de *Hybrid Semantic Localization and MApping*. Esta extensão formaliza duas novas camadas (topológicas e semântica) que são relacionadas com o mapa de ocupação, camadas essas que tem o propósito de memorizar a descrição de lugares e dos objetos e de memorizar a relação espacial entre os lugares e objetos. O processo de mapeamento semântico acontece quando o conhecimento adquirido a partir de uma descrição humana é fundido nesses mapas topológicos e semânticos.

Para possibilitar a conversão de um mapa de ocupação num mapa topológico, foi necessário formalizar uma abordagem que efetuasse a discretização do mapa de ocupação e que, a partir dessa discretização, criasse o mapa topológico. Esta abordagem foi implementada com o nome Gr2To, que deriva de *gridmap to topological map tool*. Contudo, o mapa topológico obtido pelo Gr2To não se encontra completo, por duas razões: não se encontra validado por um humano e não inclui os nomes atribuídos pelos humanos a cada um dos locais segmentados. Para resolver este problema, foi também formalizada uma abordagem com o propósito de fundir uma descrição dada por um humano com o mapa topológico interno do robô. Esta abordagem foi implementada com o nome TopoMerg, que deriva de *topological merging tool*.

Neste trabalho é proposto um procedimento que localiza o robô num espaço definido por

palavras humanas, em detrimento de um espaço geométrico, como é realizado no SLAM convencional. A partir de uma nova abordagem, que extrai um descritor global de uma imagem, é construída uma assinatura visual que é utilizada pelo procedimento para reconhecer os locais através das imagens obtidas pelas câmaras do robô. Este procedimento foi validado num contexto real e verificou-se uma taxa superior a 80% na correta associação entre as imagens e os nomes dos respetivos locais. Com este procedimento, a camada topológica HySeLAM torna-se capaz de reconhecer um local através de uma única imagem, tal e qual os seres humanos. E torna-se capaz de detetar problemas no SLAM convencional e de fornecer uma pré-localização ao SLAM para um rápido reinício. Para além destas capacidades, este procedimento de localização semântica permitirá no futuro a deteção da alteração dos cenários, podendo esta acionar um novo mapeamento de objetos e um raciocínio acerca das alterações.

Os componentes formalizados na camada topológica da extensão HySeLAM foram totalmente implementados e testados individualmente. A fim de testar e avaliar o desempenho destes componentes, foram utilizados dados reais obtidos a partir de robôs reais. Os resultados obtidos pelo Gr2To foram comparados com os resultados obtidos por um conjunto de pessoas, a executar a mesma tarefa do Gr2To. A performance do procedimento TopoMerg foi avaliada utilizando mapas topológicos obtidos a partir de uma descrição do espaço dada pelo humano e obtidos a partir do processamento dos mapas de ocupação (utilizando o Gr2To). A performance do proposto procedimento para extração do descritor global, utilizado no procedimento SeloViS, foi comparada com a performance de outros procedimentos, bem conhecidos, para extração de descritores globais e locais. Estes descritores foram testados utilizando imagens reais adquiridas no laboratório e obtidas a partir de uma base de dados pública (COLD database). A precisão da localização semântica do procedimento SeloViS foi avaliada utilizando a localização estimada pelo SLAM, que posteriormente foi projetada no mapa topológica. O procedimento SeloViS foi capaz de identificar corretamente a localização semântica do robô em mais de 90% das imagens adquiridas pelo robô.

Abstract

Mobile robotics which presupposes interaction with humans requires new paradigms of the concept of intelligence. The robot's ability to understand and perform tasks such as: "Robot, go to the office and bring me the pen that is on the table", is a requirement. Most approaches to the localization and mapping layer are not in tune with the higher layers to solve this kind of task.

The robot's ability to reason about what it senses and knows, as well as to acquire and share knowledge when interacting with a human voice, are capabilities required for the success of this cooperation. Unlike traditional SLAM (Simultaneous Localization and Mapping) methods, which only interpret sensor information at the geometric level, these capabilities require an environment map representation similar to that of humans.

This work aims to give an answer to the question *How can a SLAM approach be extended in order to include the human into the mapping process?* and for that it explores and investigates a hybrid approach (metric and semantic) for the localization and mapping layer, in order to reach a human-like environment map representation (knowledge). This will allow for the development of a new strategy that will enable the robot to: complete and build this knowledge through human interaction; associate sets of characteristics of observations to words, which are, in turn, associated to objects and places; and reason at a semantic level about this knowledge and its own observations.

In this thesis, a novel semantic framework extension for SLAM is proposed in order to extend the geometric mapping into topological and semantic mapping. The name of this framework is HySeLAM, which stands for Hybrid Semantic Localization and MApping. This framework formalizes two new layers (topological and semantic) connected to the occupancy gridmap in order to describe places and objects and the relation between them. The semantic mapping process happens when the knowledge acquired from the human description is merged into these topological and semantic maps.

In order to translate the occupancy grid-map into the augmented topological map, the formalization of an approach to discretize a grid-map into a topological map was required. This approach was implemented with the name Gr2To, which stands for gridmap to topological map tool. This tool abstracts the gridmap with delimited places and the connectivity between places. The obtained topological map from Gr2To is not complete and was not validated by a human. To solve this an approach was formalized to merge a description given by a human into the internal augmented topological map. This approach was implemented with the name TopoMerg, which stands for topological merging tool.

A semantic localization procedure was also proposed, which locates the robot in a space defined by human words instead of a geometric space, as in a conventional SLAM. This approach was implemented with the name SeloViS. Using only a visual signature, which is obtained from a new global image descriptor, this procedure is highly accurate in recognizing a given place (more than 80% in the testing data). With this procedure, the HySeLAM topological layer is able to recognize a place at a glance as humans, to detect problems in the conventional SLAM and to help

reach a fast SLAM start. Moreover, this semantic localization procedure will help in the future to detect scene changes, which will trigger a new object mapping and reasoning.

The formalized topological components of HySeLAM framework were fully implemented and tested individually. In order to test and evaluate the performance of these components, real data/information obtained from the real robots was used. The results obtained by Gr2To were compared against the results obtained from the same task done by humans. The performance of TopoMerg procedure was evaluate using augmented topological maps obtained from descriptions given by humans and from computed gridmaps (by using the Gr2To). The performance of the new visual signature extractor, proposed for SeloViS procedure was compared against other well known global and local descriptor extractors. These descriptors were tested using real images acquired in the laboratory and images downloaded from a public database (COLD database). The semantic localization accuracy of SeloViS procedure was evaluated using as ground-truth the localization obtained from the SLAM procedure which was after projected on the augmented topological map. The topological layer, with its components, was fully tested in a real environment using an industrial robot. With this framework, the robot was able to extract the topological map from the occupancy gridmap. After that, the robot was able to infer the names that tag each place from a human description. With the augmented topological map, the SeloViS procedure was able to extract from the environment a set of visual signatures for each place. Using the topological map enriched with visual signatures and place names, the SeloViS classifier was trained. After this training step, the accuracy of the SeloViS procedure was tested. The SeloViS procedure was able to recognize correctly the robot location in more than 90% of the images acquired by the robot.

Acknowledgements

First and foremost I want to thank my advisers Professor António Paulo Moreira and Professor Paulo José Costa for their unwavering support, guidance and constant encouragement.

The members of the Robis-INESCTEC group have contributed immensely to my personal and professional time at Faculty of Engineering of University of Porto. The group has been a source of friendships as well as good advice and collaboration. Special thanks to André Bianchi Figueiredo, Héber Sobreira and Miguel Pinto.

Lastly, I would like to thank my family for all their love and encouragement. And most of all my loving, encouraging, and patient wife Luísa whose faithful support during all the stages of this Ph.D. is so appreciated. Thank you.

Filipe Baptista Neves dos Santos

Porto, Portugal

July 22, 2013

" If you can, help others; if you cannot do that, at least do not harm them. "

Dalai Lama

Contents

List of Figures

Chapter 1

Introduction

The main motivation behind this work is the belief that robots can make an important contribution to the future of our society. Instead of having robots that work in high-tech factories with preprogrammed tasks, we are going to have robots that work and cooperate alongside us.

In the last three decades we succeeded in developing robots that execute a limited set of tasks with precision in a repetitive and continuous manner. These robots are mainly used in high-tech factories for massive production. Such robotic automation has freed humans from heavy and tedious labor. However, if we take one of those robots and place it in our homes, offices, public spaces or even in small factories, it becomes inefficient or usefulness. Three reasons for this are:

- These robots were developed to work in a structured and controlled environment. Simple light variations or artificial landmarks missing can make them inoperative or malfunctioning.
- The set of tasks of these robots is limited and preprogrammed. These robots do not have the capacity to learn a new task from a human demonstration or human description.
- These robots can only accept a limited set of commands that are only known to a professional user. These robots can not acquire or share information from voice interaction, gestures or natural textual interaction.

With this in mind and knowing that the new generation of robots is going to interact with humans and work and live in our homes, offices, public spaces and small factories, new capabilities must be developed for them. Georges Giralt, in the book edited by Siciliano and Khatib (2008), emphasized four general characteristics these robots need:

- they may be operated by a nonprofessional user;
- they may be designed to share high-level decision making with the human user;
- they may include a link to environment devices and machine appendages, remote systems, and operators;

- the shared decisional autonomy concept (co-autonomy) implied here unfolds into a large set of cutting-edge research issues and ethical problems.

Rodney Brooks, also in Siciliano and Khatib (2008), describe these robot capabilities in terms of the age at which a child has equivalent capabilities:

- the object-recognition capabilities of a 2-year-old child;
- the language capabilities of a 4-year-old child;
- the manual dexterity of a 6-year-old child; and,
- the social understanding of an 8-year-old child.

Teresa Escrig, in Escrig, describes five features for these new robots that should be worked on and improved:

- Perception – *" not only taking data from the environment, but transforming it into knowledge (and even wisdom) to be able to interpret and modify its behavior according to a result of this perception."*
- Reasoning – *"drawing conclusions from data/knowledge taken from perception."*
- Learning - *" with new experiences, the robot needs to perceive and reason to obtain conclusions, but when the experiences are repeated, a learning process is required to store knowledge and speed up the process of intelligent response."*
- *" Decision Making, or the ability to prioritize actions, is necessary to be able to be safe and effective in the solution of different autonomous applications."*
- *" Human-Robot Interaction at many levels is also necessary. For example, natural language processing: understanding the meaning of sentences exchanged with humans, depending on the context and to be able to properly respond, and emotions rapport."*

This new generation of robots introduces not only the critical issue of human-robot interaction to the scientific community, but also the front-line topics encompassing cognitive aspects: user-tunable human-machine intelligent interfaces, perception (scene analysis, category identification), open-ended learning (understanding the universe of action), skills acquisition, extensive robot-world data processing, decisional autonomy, and dependability (safety, reliability, communication, and operating robustness).

1.1 Global Overview

Rhino by Burgard et al. (1999), figure 1.1, was a pioneer attempt to take a robot to a public space for human interaction. Rhino had been built to assist and entertain people in public places, such as museums. In May 1997, Rhino was deployed in the "Deutsches Museum Bonn". During a

six-day installation period the robot gave tours to more than 2,000 visitors. Burgard et al. (1999) reported a situation where more than a hundred people surrounded the robot from all sides, making it difficult for the robot to reach the exhibits as planned while not losing track of its orientation. Also, they reported that "*some of the users were not at all cooperative with the robot, imposing further difficulties for the software design. Often museum visitors tried to "challenge" the robot. For example, by permanently blocking its way, they sometimes sought to make the robot leave the designated exhibition area towards other parts of the museum, where several unmapped and undetectable hazards existed (including a staircase).*" The experiments with Rhino were useful to learn that we cannot necessarily expect humans to be cooperative with robots, so the safety of the system may not depend on specific behavioral requirements on the side of the users. On the other hand, people are thrilled if robots interact with them just like they are if people interact with them. Another important result from these experiments was the results of the interview made to museum visitors where most of them assigned more weight to the robot's interactive capabilities than to its ability to navigate

Figure 1.1: From left to right: the Rhino robot, by Burgard et al. (1999), Robox robot, by Siegwart et al. (2003), Jinny robot, by Kim et al. (2004).

After Rhino, we found several other similar robots in literature which were deployed in public spaces for human interaction, as Minerva by Thrun et al. (2000), Robox by Siegwart et al. (2003), Mobot by Nourbakhsh et al. (2003), and Jinny Kim et al. (2004). These robots rely on 2D accurate metric representations of the environment, derived from simultaneous localization and mapping (SLAM) techniques. They are developed to be reactive and they cannot acquire any knowledge from human interaction. They have quite limited communication capabilities and they can not interact with objects present in the environment.

The interaction with objects present in the environment is a demand for these robots to be useful as personal assistants. With this in mind, the Willow Garage company developed a new robot called PR2, figure 1.2. Comparing this robot to previous ones, regarding physicality, PR2

is augmented with two arms and a stereo vision system. PR2 is a research platform and it was created to help boost several research topics, as grasping/manipulation, human-robot interaction, motion planning, perception, and task planning. Bohren et al. (2011) report several experiences and results while they were building an autonomous robotic assistant using the PR2 platform and ROS [1]. This robot was developed for a particular application for a personal robot: fetching and serving drinks.

Figure 1.2: From left to right: PR2 robot, by Bohren et al. (2011), Care-o-bot Robot by Reiser et al. (2009), TUM-Rosie robot, by Beetz et al. (2009).

Developing a new robot requires the integration of many complex subsystems, as perception, motion planning, reasoning, navigation, and grasping. Bohren et al. (2011) noticed that even with an extensively validation process for each of these individual components, the subsequent step of integrating them into a robust heterogeneous system is a hard task which is not solved yet. Even so, they have successfully assembled a personal robot for fetching and serving drinks. However, this robot was developed with preprogrammed object detection tasks and with predefined actions. For example, the action to take a drink from the fridge was developed by the programmers for a specific fridge and bottle drink. This robot has not learned this action from a human demonstration. If the robot is deployed in a home that has a different fridge and bottle drink it will probably fail in the action, as it does not have the capacity to relearn or adapt the action.

Other similar robots that were developed to interact with objects are found in literature, as Care-o-bot by Reiser et al. (2009), TUM-Rosie by Beetz et al. (2009), and Herb by Srinivasa et al. (2009). When compared to Rhino, these robots are one step forward, being able to perform mobile pick-and-place tasks and even fold towels or prepare meals. However, performing human-scale manipulation tasks in dynamic environments, like a household, nursing home or even in

[1] ROS is an open source Robotic Operating System which was originally developed in 2007 under the name Switchyard by the Stanford Artificial Intelligence Laboratory. As of 2008, development continues primarily at Willow Garage, a robotics research institute. Today, this software initiative is supported by an international community of robotics researchers and managed by Open Source Robotics Foundation – `http://www.ros.org`

some small factories, remains very challenging because robots often do not have the knowledge to perform them the right way. The main reason is that the robot actions and object perceptions remain static even when it interacts with a human or sees the human doing the same action in the right way.

A personal robot assistant should have several processes related to attention, memory, language production and understanding, learning, reasoning, problem solving, and decision making. This means that these robots should have some degree of cognitive capacity. Thus, the iCub robot was developed, figure 1.3, to promote a collaborative and multi-disciplinary initiative in artificial cognitive systems. Sandini et al. (2007) describes this robot as an open-systems 53 degree-of-freedom cognitive humanoid robot, 94 cm tall, equivalent to a three year-old child. This robot is able to crawl on all fours and sit up, its hands allow dexterous manipulation, and its head and eyes are fully articulated. Also, it has visual, vestibular, auditory, and haptic sensory capabilities.

Figure 1.3: From left to right: Icub, Nao, Asimo and Ecce

Tikhanoff et al. (2011) reports several simulated experiments showing that the iCub robot is able to learn to handle and manipulate objects autonomously, to understand basic instructions, and to adapt its abilities to changes in internal and environmental conditions.

Broz et al. (2009) explores a system for the learning of behavior sequences based on rewards arising from social cues, allowing the iCub to engage a human participant in a social interaction game. In their work they reported that gaze is a powerful social cue, and it is also one that becomes socially significant at an early developmental stage; even young infants are responsive to other's gaze direction.

Figueira et al. (2009) reports a new technique where the iCub learns autonomously new objects from the environment. The objects are defined as clusters of SIFT [2] visual features. When the robot

[2] SIFT stands for Scale-Invariant Feature Transform and it is an algorithm in computer vision to detect and describe local features in images. The algorithm was published by David Lowe in 1999 and it is commonly used for object recognition, robotic mapping and navigation, image stitching, 3D modeling, gesture recognition, video tracking.

first encounters an unknown object, it stores a cluster of the features present within a distance interval, using depth perception. Whenever a previously stored object crosses the robot's field of view again, it is recognized, mapped into an egocentric frame of reference, and gazed at.

Miihlig et al. (2010) reports a new robot control and learning framework, applied to the ASIMO robot, figure 1.3. With this new framework the robot was able to learn an invariant and generic movement representation from a human tutor, and even to adapt this generic movement to a new situation.

In literature we found other similar robots that were used to developed cognitive capacity, as the Nao robot developed by Aldebaran Robotics company, and ECCEROBOT (Embodied Cognition in a Compliantly Engineered Robot) developed under the 7th framework program of the European Union. These robots, figure 1.3, are a step forward but faraway from the robot capabilities described by Rodney Brooks. Object-recognition, language and manual dexterity capabilities are still a challenge, as correct world knowledge representation and learning process are also open fields for research in the next decade. As reported by Bohren et al. (2011), even when a good approach to one of this problems is found the integration with the other robot components is also a challenge.

1.2 Research questions

This work evolves from previous research works in robotics, mainly in navigation, localization and field mapping. In the last two decades these three fields were intensively explored and nowadays there are valuable solutions making it possible for robots to navigate and explore a building autonomously. The main contributions were novel SLAM approaches based on laser range finders and techniques derived from Bayes's rules, as described in the section 2.4. Nerveless, there are still some open issues related to localization and field mapping, such as:

- *How to validate the location estimation obtained from a SLAM approach?*

  When the robot is placed in an unknown place of a symmetrical building most SLAM approaches which do not use artificial landmarks may converge to a solution that may not be the correct one. With a more robust SLAM approach to the scenario, better sensors and optimized trajectories, the correct solution may be reached but it will take more time than a human does and it will be less efficient because it requires more movements from the robot. This happens because these algorithms rely on geometric information and/or in rudimentary visual features as the ones used by SIFT and SURF algorithms.

- *How to detect a Kidnapping situation or recover from it?*

  Kidnapping is one of the hardest problems to solve in the field of SLAM. Kidnapping a robot refers to the act of picking up a robot in the middle of its operation and place it somewhere else in the environment, without notifying it. It is similar to knocking a human over the head in order to make it unconscious, and moving them to some other location in

the building. Even if the new location is geometrically identical to the previous one, the human will recover because they can use natural features and reason in order to detect that something is wrong. In our knowledge, there is not a robotic approach as efficient as the human mechanisms because most SLAM approaches rely on geometric information or on rudimentary visual features as the ones used by SIFT and SURF algorithms.

- *How to detect if the associated features are the correct ones?*

  The robustness and accuracy of any SLAM approach depends on the correct association of obtained measurement and stored feature. Most SLAM approaches rely on rudimentary geometric or visual features which are sometimes associated to furniture present in the environment. If someone moves the furniture these features will get a new location, thus affecting localization estimation negatively. Most SLAM approaches will recover and update the feature location. However, during this time, there is an increase in the uncertainty of the estimation as well as in the risk of localization failure. The knowledge of features that are good or more reliable will help solve this problem. For that, information is needed about what belongs to the features during mapping and localization, but this requires a higher level of knowledge.

- *How to make this knowledge sharable to other kind of robots?*

  When we have a team of heterogeneous robots, the map acquired by one robot (using a SLAM approach) is not always easily transferable to others robots. This happens because these robots have different sensors, each robot has its own sensor configuration, each robot has an acquired map with its own referential of observation, the robot sensors cannot observe the same features as the other robots, and each robot can have its own map representation. In our point of view, the best approaches to solve this issue are those that translate this map to a higher level of representation, as topological or semantic maps. The higher level of representation makes it possible to abstract the low-level features and to share to other robots a description about the ways the place is organized. This description can be used later to help the robot build its own low-level map. Also, this common description can be useful to describe tasks for the robots in the same framework and it can be useful in the validation of maps.

As said before, I believe that there will be robots which will work and cooperate alongside us. So, robots will work in a place organized by humans, executing tasks described by humans. With these assumptions, we can tackle the previous questions considering human knowledge input in the mapping and localization process. However, this leads us to our fundamental research question:

- **How can a SLAM approach be extended in order to include the human into the mapping process?**

  The robots that will work alongside us should have a perception of the world that is compatible to ours, because they are going to manipulate objects and places that are structured

by humans. So, object interaction, perception and manipulation as well as strong communication abilities are required for the next generation of robots. To make this possible, we will need to extend geometric and/or topological based mapping and localization to semantic mapping and to build a common cognitive workspace, as suggested by Dellaert and Bruemmer (2004).

1.3 Main contributions

This work was driven by the fundamental research question *How can a SLAM approach be extended in order to include the human into the mapping process?*. But also taking in mind that the answer should help to solve the remain open questions in SLAM, enable the robot to understand human descriptions about the places and objects and to reason about the acquired knowledge through a SLAM approach.

During the construction of my approach to answer this question, the following contributions to the scientific community were carried out:

1. *A new semantic extension to SLAM.*

   In this thesis a novel semantic extension framework for SLAM approaches based on gridmaps was formalized. The name of this framework is HySeLAM, which stands for Hybrid Semantic Localization and MApping. This framework formalizes two layers connected to the SLAM approach in order to describe places and objects and the relation between them.

2. *An approach to convert a grid-map into an augmented topological map.*

   In order to connect the grid-map to augmented topological map, it was required to formalize an approach to discretize a grid-map into this topological map. This approach was implemented with the name Gr2To, which stands for gridmap to topological map tool. This tool abstracts the gridmap with delimited places and the connectivity between places

3. *A formal strategy that answer to the question "how to merge a place description given by a human into the augmentation topological map".*

   The obtained topological map from Gr2To is not completed and was not validated by a human. To solve an approach to merge a description given by a human into the internal augmented topological map was formalized. This approach was implemented with the name *TopoMerg*, which stands for topological merging tool.

4. *An approach for visual place recognition, based on a new global descriptor and filtered by a Markov chain.*

   In order to get a redundant approach for the conventional SLAM approach, a visual place recognition is proposed. This approach localizes the robot on the augmented topological map. Two main contributions were made in this topic: a new global descriptor based on

local binary patterns operator was proposed, and a new filter based on Markov chain was proposed to constrain the classifier probabilities to flow according the topological map.

1.4 Structure of this Thesis

In synthesis, the primary aim of this work is to address the problem of discovering and incorporating higher level structures into the maps generated by SLAM approaches in order to make this knowledge sharable to humans in a voice channel. In this introduction, the motivation for this work has been presented in the context of an overview of the existing literature, and each chapter presents a more detailed review of specifically related work.

The document is structured as follows :

- Chapter 2 makes a categorization of input information of the robot, an exhaustive review about basis and foundations in the topic of SLAM and a review about spatial knowledge representations. To create an internal knowledge about the world, the robot requires information provided by internal/external sensors and information provided by communication channels. In this work the most common sensors are categorized along with the way this information can be preprocessed and used to create knowledge about the world. An exhaustive review of different approaches to SLAM was carried out to make it possible to characterize how the world is stored and represented in the most common SLAM approaches. This will lead to an answer to the question of how to extend this to other kinds of representation. Finally, different approaches for spatial knowledge representation are also shown, namely grid maps and topological, semantic and hybrid maps.

- Chapter 3 addresses the fundamental question: How to make the robot knowledge closer to the human description. Section 3.1 presents the literature review related to the fundamental question, and section 3.2 details the proposed approach for semantic mapping, called Hybrid localization and mapping (HySeLAM).

- Chapter 4 presents a novel approach that translates a grid map, which was obtained from SLAM, into an augmented topological map defined by the HySeLAM framework. This approach is capable of translating a 3D grid map into an augmented topological map. It was optimized to obtain similar results to those obtained when performed by a human. Also, a novel feature of this approach is the augmentation of the topological map with features such as walls and doors. Section 4.1 presents a global overview of the problem and different approaches to translate a grid map into a topological map. Section 4.2 presents the novel approach to translate a grid map into a topological map. Section 4.3 presents the results obtained using this novel approach. Section 4.4 presents the chapter conclusions and future directions.

- Chapter 5 addresses these two fundamental questions: *How can a place description given by a human be translated into an augmented topological map?* and *How can two augmented*

topological maps be merged into a single one?. Section 5.1 presents a global overview of the problem and our approach to translating a place description given by a human into an augmented topological map. Section 5.2 presents a global overview of the graph matching problem. Section 5.3 presents our approach to merging the two topological maps obtained from a human description and an occupancy grid-map. These topological maps have a graph structure, so the merging problem results in a graph matching problem with constrains. The proposed approach to merging these graphs was developed based on the graph matching theory and recurring to tree search concepts.

- Chapter 6 addresses two questions: *How must the visual signatures be constructed to allow for robust visual place recognition and classification?* and *How can the augmented topological map be included in the semantic localization procedure for increased place recognition accuracy?*. Section 6.3 proposes the LBPbyHSV approach for the global descriptor extraction and compares the performance of this global descriptor against local and global descriptors for indoor place recognition. Section 6.4 proposes an approach based on Markov Chains formulation to answer the second question for this chapter.

- Chapter 7 presents the implementation of this HySeLAM framework into a real robot. The tests are described and results are shown.

- Chapter 8 presents the final discussion, the conclusions obtained from this work and future work directions.

Chapter 2

Background: How a robot sees the world

Cooperation between robot and humans creates the challenge of discovering and incorporating higher level structures into the information processed and stored by robots. To solve this challenge and to answer to the fundamental question of this work (*How can a SLAM approach be extended in order to include the human into the mapping process?*) one must know how the robot acquires information about the world and about itself and also how this information is processed and stored. The understanding of all these robotic processes (acquisition, processing and storing) involves several fields of knowledge, which are well detailed in the handbook of robotics edited by Siciliano and Khatib (2008).

The aim of this chapter is to give a brief overview about six essential topics, which are: localization, motion sensing, perception sensors, imaging processing, maps, and methodologies used for the simultaneous localization and mapping problem. They are important because they were essential to draw the proposed approach which answers to the fundamental question. This chapter is organized into four sections, as follow:

- Section 2.1 presents three kinds of information gathered by the robot: self motion, localization and perception raw data provided by the most common perception sensors. Understanding how this information is gathered and processed is important because one of the most interesting capacity of the robots over the humans is the high accuracy, precision and repeatability observed during their task execution. These are possible due the accuracy of the observations made by some sensors and the strong math formulation used for processing this information. Understanding how self motion and localization are gathered and processed mandatory in order to extend the robot capabilities without losing its inherent high accuracy, precision and repeatability.

- Section 2.2 presents a brief overview of the techniques used in image processing in order to show how complex this task is and to what kind of information can be extracted from the vision system. These vision systems are considered important for robotics because they

make it possible for the robots to observe the same things as humans do. This observation makes it possible to for the robots to gather the same references as humans do, which is important vital when a cooperation between robot and humans is required.

- Section 2.3 presents an overview about two different kind of maps used in robotics, topological and metric maps and how these maps are used. This is the primitive knowledge stored by the robot and it is the primitive knowledge to which the semantic mapping should be referred.

- Section 2.4 presents several approaches used in the simultaneous localization and mapping process. With these approaches the robot can "learn" the environment structure (through motion exploration) and they can estimate the robot location, based on the apprehended knowledge and sensor observations. As mentioned earlier, in chapter 1, these SLAM approaches will be the starting point for the semantic mapping process.

2.1 Motion, Localization and Perception Sensors

A global overview about the type of information gathered by robot sensors is useful to understand the relationship between information gathered from robot-human interaction and information gathered from robot sensors. This input information gathered by a robot can be classified into four main classes: self-motion information, localization information, world perception and interaction information. However, only the first three are described.

2.1.1 Self-motion information

Self-motion information is a class of information that describes the motion of a referential frame attached to a point of the robot. This information is gathered from processed readings obtained from motion sensors and encodes with values the magnitude of referential motion derivatives as: linear velocities and accelerations and angular velocities and accelerations. These derivatives are usually integrated and projected into another referential frame, in order to describe the robot motion or robot part motion. This motion can be described in one, two or three dimensions.

To this triplet motion sensor, processing and integration is usually denominated as dead reckoning system. The advantages of dead reckoning systems are the higher update rates and the independence of external sensors; it is a stand-alone system. The main disadvantage is the exponential degradation of accuracy over time. This happens because the estimation relies in the integration of sensor readings which are always affected by noise. To eliminate this disadvantage, the estimations of a dead reckoning system are usually fused to other systems using the Kalman filter theory, described by Kalman (1960).

The most widely used dead reckoning system is odometry. Odometry is a system that observes the rotation of one or more robot wheels and projects this rotation to linear and angular velocities, which are used to estimate the system motion. Borenstein et al. (1996) describes several methods

to project and to integrate this observation. Odometry provides good short-term accuracy, is inexpensive, and allows very high sampling rates. However, as dead reckoning system, the inaccuracy in pose estimation increases over time, and it is worse when there are mechanic imperfections, different radius of wheels and poor calibration of odometry. Borenstein and L.Fen (1996) have detailed and classified the main sources of errors in the odometry system.

Inspired by odometry, several researchers have replaced the sensors attached to the wheels by cameras. This technique is usually called visual odometry. "Visual odometry" was coined in 2004 by Nistér et al. (2004). Basically, the visual odometry approach takes two calibrated and successive images, then it extracts features from them and associates features from both images. From this features association it is possible to construct an optical flow field which is used to estimate the camera motion. This approach eliminates the mechanical errors of odometry systems but is extremely heavy in terms of processors usage and its accuracy depends on the quality and number of features extracted from the images. Maimone et al. (2007) have successfully applied this technique in a Mars exploration rover in order to autonomously detect and compensate for any unforeseen slip encountered during a drive. Scaramuzza and Fraundorfer (2011) presents the last thirty years of research related to visual odometry.

With advanced microelectromechanical systems (MEMS), it was possible to build low cost and small sensors to measure linear accelerations and angular velocities. These only take measurements from one dimension, and they are known as accelerometers and gyroscopes sensors. Usually, three accelerometers and three gyroscopes sensors are placed in one unit. This unit is called Inertial Motion Unity (IMU). IMU are the core of the inertial navigation system (INS). INS is a system that process the measurements of an IMU in order to continuously calculate via dead reckoning approach the position, orientation, and velocity of a moving object without the need for external references. Nerveless, as this system integrates the position from a second derivative, it will suffer an higher exponential accuracy degradation over time due to errors present in observations. Unsal and Demirbas (2012) improves the IMU performance by compensating the deterministic end stochastic error. Low cost INS have an acceptable accuracy in the orientation estimation. chul woo Kang and chan gook Park (2009) presents an attitude estimation method for a humanoid robot using an extended Kalman filter and one IMU. However, the accuracy of INS in the position estimation is extremely low after few seconds. For these reason, INS is usually fused to other systems.

2.1.2 Localization information

Localization information is a class of information that relates the robot position to the origin of navigation referential frame. This class of information is required by the robot for the navigation and mapping task. This information is obtained by taking distances measurements from the robot to artificial or natural landmarks and by using triangulation, trilateration or matching techniques. These systems generally have a lower update rate when compared to *Dead reckoning* systems, but because they are not dependent on previous measurements, the accuracy of position estimation is

not affected over time. The accuracy is only dependent on observation quality and geometry of robot and observed landmarks. This accuracy is described by a covariance matrix.

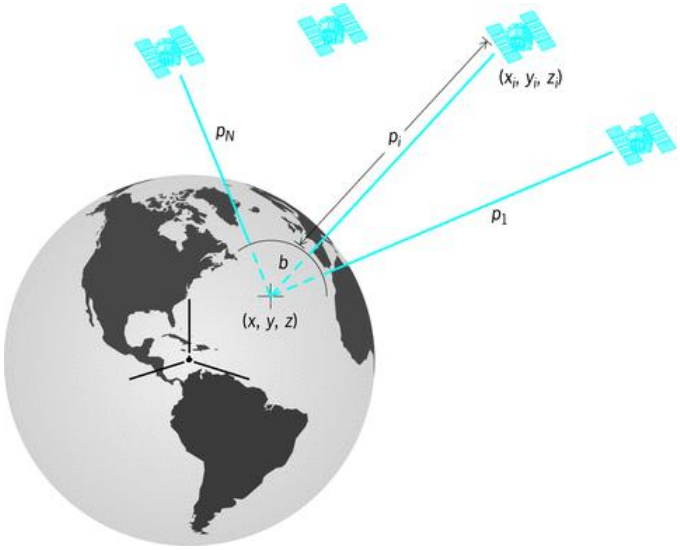

Figure 2.1: Global Positioning System (GPS) has 24 satellites in orbit. Each GPS satellite broadcasts radio signals providing their locations, status, and precise time from on-board atomic clocks. A GPS device receives the radio signals, taking their exact time of arrival, and uses these to calculate its distance from each satellite in view.

For outdoor robots, Global Navigation Satellite Systems (GNSS), as GPS and GLONASS, are the most common source of localization information. The theory behind a GNSS is simply that there are active beacons (satellites) emitting signals and receivers tracking these signals. In the receiver the reception and emitting time are annotated, and used to estimate the distance (pseudo-range) from the receiver to the satellites in view. These distances and satellite positions are used to estimate the receiver location, as described in Xu (2003). Deferential techniques and phase observation can be used to improve the accuracy of the system. Thrun et al. (2006) describes the Stanley robot, which has used GPS receivers to estimate the vehicle position during the DARPA Grand Challenge. The goal of this challenge was to develop an autonomous robot capable of traversing unrehearsed off-road terrain. The GPS was fundamental for the robot to know its position in real time in order to navigate through the environment. Other approaches, as described by Santos (2007), based on multiple GPS receivers, can also estimate the posture of the vehicle (pitch, roll and yaw).

The localization information based on GNSS is not available in all places, as indoor sites or obstructed places, or it is very inaccurate in very dense sites, as forests or very narrow streets. For these sites, other approaches have been proposed. The Aerospace Robotics Laboratory at Stanford University has developed a prototype, a GPS-based local area positioning system, to help filling this gap. Rather than employing orbiting satellites, small low-power transmitters called pseudolites (short for "pseudosatellites") are distributed over the place. LeMaster and Rock (2001) reports an accuracy of 0.8 meters using this approach in a Mars rover. This approach has the advantage

of using a common GPS receiver to collect the measurements and process the system position. The main disadvantage is these signals do not cross walls, which implies a high cost because several pseudolites must be used in each place delimited by walls. Usually this approach is only considered for large indoor open-spaces or to complement GNSS in outdoor places.

For indoor robots, Priyantha et al. (2000) proposes the Cricket location-support system for in-building. This approach has a low cost when compared to pseudolites and it uses the basic principle of GNSS. Instead of emitting radio-frequency signals only, this system emits ultrasound signals for the range distance measurement and radio-frequency signals for clock synchronization. There are several beacons emitting ultrasounds signals and there is the listener that listens to the messages from beacons. The listener uses the travel time of this messages to estimate distances to the beacons and then the position. There are other approaches using artificial landmarks, as the infrared spectrum approach, describe by Hijikata et al. (2009) and Sobreira et al. (2011), RFID by Zhou and Shi (2009), and the visible spectrum with bar codes by Liu et al. (2010).

Some of these localization systems, based on artificial landmarks, have minimal space invasion and they are easy to install, however most of the time it is not possible to install the artificial landmarks in part or all robot environment, due to their cost or a specification that limits artificial landmarks usage. For these particular situations, other strategies were sought during the last two decades. In general, these strategies create or use a map of geometric or visual features (natural landmarks) which are used to estimate or update the robot position based on feature observations obtained from robot sensors. There are different implementation approaches, described in section 2.4, which depend on world perception sensors, detailed in section 2.1.3, and which use one of the different map representations, detailed in section 2.3.

2.1.3 World perception information

World perception information is a class of information that is obtained from robot sensors and which contains information about shape and color related to the surroundings of the robot.

The information acquired by the robot is always discretized in terms of space and time. So the only way to obtain the shape of the world is getting samples of 3D points, and we will refer to a collection of 3D points as a *point cloud* structure $\wp$, adopting the notation used by Rusu (2009) . Point clouds represent the basic input data format for 3D perception systems, and provide discrete, but meaningful representations of the surrounding world. Without any loss of generality, the $\{x_i, y_i, z_i\}$ coordinates of any point $p_i \in \wp$ are given with respect to a fixed coordinate system, usually having its origin at the sensing device used to acquire the data. This means that each point p_i represents the distance on the three defined coordinate axes from the acquisition viewpoint to the surface that the point has been sampled on.

There are many ways of measuring distances and converting them to 3D point. In the context of mobile robotic applications, the three most used approaches are: Time-of-Flight (TOF) systems, triangulation techniques, and structured light sensing devices.

Time-of-Flight (TOF) systems Time-of-Flight (TOF) systems measure the delay in which an emitted signal hits a surface and returns to the receiver, thus estimating the true distance from the sensor to the surface; it involves sensing devices such as Laser Measurement Systems (LMS) or LIDAR, radars, Time-of-Flight (TOF) cameras, or sonar sensors, which send "rays" of light (e.g. laser) or waves of sound (e.g. sonar) into the world, which will reflect and return to the sensor. Knowing the propagation speed of the ray/wave, and using precise circuitry to measure the exact time when the ray was emitted and the signal returned, the distance d can be estimated as (simplified):

$$d = \frac{vt}{2} \tag{2.1}$$

where v represents the speed of the ray (e.g. speed of light for laser sensors is the constant c or if sound is used it is $v = 348m/s$), t is the amount of time taken for emission and reception of the ray, the 2 is due to the time to go and return.

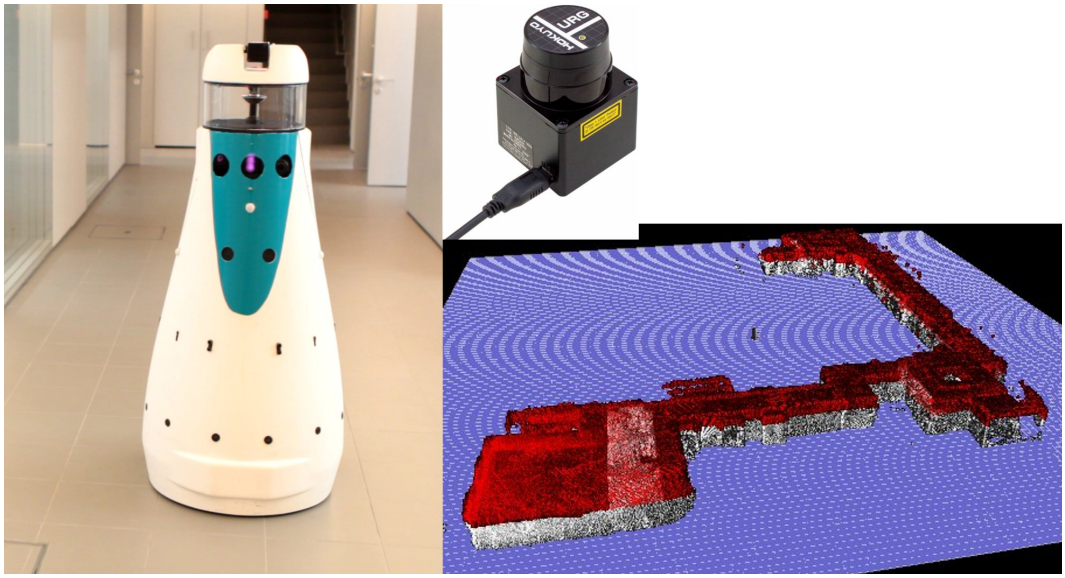

Figure 2.2: On the left, RoboVigil with a laser range finder tilting. On the right, the acquired point cloud is assembled into a 3D gridmap.

Laser measurement systems presented are inherently 2D, in the sense that they combine a laser range finder with a rotating mirror to measure distances on a plane. To obtain a 3D point cloud, they are placed on rotating units such as pan/tilt or robot arms, as done in RoboVigil figure 2.2. Using the kinematics of these units, we can obtain a multitude of such 2D measurement planes, which can then be converted into a consistent 3D representation.

A solution that can be employed to obtain dense 3D data faster is to use Time-of-Flight camera systems, such as those shown in figure 2.3. These systems can provide 3D data (point cloud) representing a certain part of the world with higher update rates (30 fps) thus enabling their usage in applications with fast reactive constraints. The resultant data is however noisier than the one

Figure 2.3: Time-of-Flight camera systems: D-IMager from Panasonic, Fotonic by Canesta and the well known SwissRanger, an industrial TOF-only camera originally developed by the Centre Suisse d'Electronique et Microtechnique, S.A. (CSEM) and now developed by the spin out company Mesa Imaging.

acquired using laser sensors, not as dense (as the resolution of most TOF cameras is very low, as 176x144 pixels), and can suffer from big veiling effects.

Triangulation techniques Triangulation techniques estimate distances by means of connecting correspondences seen by two different sensors at the same time. To compute the distance to a surface, the two sensors need to be calibrated with respect to each other, that is, their intrinsic and extrinsic properties must be known. Triangulation techniques usually estimate distances using the following equation (simplified):

$$d = \frac{fT}{\| x_1 - x_2 \|} \tag{2.2}$$

where f represents the focal distance of both sensors, T the distance between the sensors, and x_1 and x_2 are the corresponding points (features) in the two sensors. Though many different triangulation systems exist, the most popular system used in mobile robotics applications is the stereo camera.

Stereo camera based systems have higher acquisition speed when compared to those based on the Time-of-Flight principle. Another advantage, they are passive, in the sense that they do not need to project or send any light or sound sources into the world in order to estimate distances. Instead, they just need to find point correspondences in both cameras that match a certain criterion. However, a good texture in the environment is required in order to extract image features to find point correspondences and then depth. This means that point cloud will not get the number of points equal to the number of pixels in the image. The resultant point cloud datasets acquired using passive stereo cameras are not always very dense, or complete, and might contain holes for portions of the scene where texture is absent or point correspondences are hard to estimate. Lazaros et al. (2008) presents several stereo matching techniques for depth information extraction.

Structured light sensing devices Special category of sensors use structured light to obtain high precision point cloud datasets. A few years ago, these sensing devices were rarely used in robotic applications in indoor environments, and their use was mostly limited to 3D modeling applications in special constrained environments. This has changed with the low cost electronic device called kinect, which was developed for the game industry, and is nowadays widely used in indoor robots. This device features an RGB camera and depth sensor. The depth sensor consists of an infrared laser projector combined with a monochrome CMOS sensor; the laser projector enriches the environment with structured textures in almost any ambient light conditions. Using triangulation techniques allow for depth points. The advantages of this system are: it has a point cloud with a fixed number of points, a good texture for image features extraction, the color information is attached to each point and it is cheap solution when compared to others.

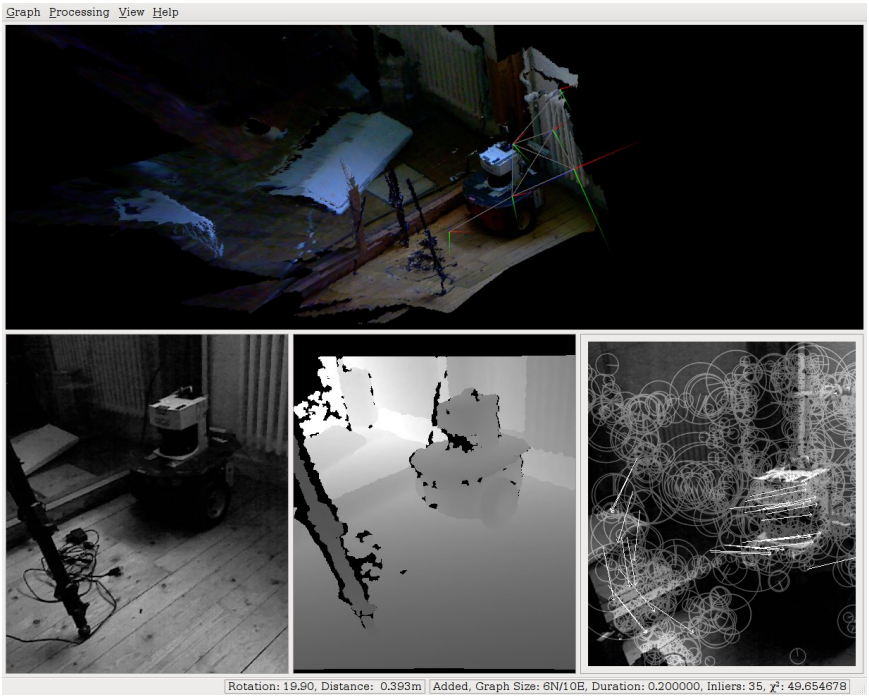

Figure 2.4: 3D Point cloud obtained from kinect, using ros and RGBDSLAM, image made available by Endres.

A solution in this sense is to make sure that the world contains only objects which exhibit good texture characteristics, or alternatively (in Kinect it happens all the time) to project random texture patterns onto the scene in order to improve the correspondence problem. The latter falls into the category of transforming the passive stereo approach into an active one, where intermittent flashes from texture generators could be projected on objects using a structured light projector. Figure 2.4 shows a 3D model obtained using a kinect sensor and an 6DOF SLAM.

Asus Xtion is a commercial alternative to the kinect. Both attach color information to the point cloud, so for that reason these sensors are also known as RGB-D sensors (RGB stands for color and D for depth). The RGB-D sensor gets the best information of vision and LRF, and is useful

for shape detection. RGB-D Indoor sensors.

Image sensors, in visible and invisible human spectrums (infrared and thermal), are also an important supplier of world perception information for the robot. However, extracting information from a lower level, low features as corners and edges, to a higher level, as object/human detection, requires complex algorithms, whose robustness depends on image sensor quality and external conditions (light intensity). Due to the importance of this sensor to this work, it is addressed in section 2.2 with detailed techniques and approaches for information extraction from images.

2.2 Image processing

Vision systems are considered a very important module for robots because they make it possible for robots to see the same things as humans. The eyes are the main sensors used by humans to navigate, localize and act in the external world. However, vision systems (lens and sensor) with the same resolution, performance and size as the human eyes do not exist. Besides that, humans and animals have improved their imaging process for a long period of time and the theory behind those processes is not totally known to humans. Nonetheless, fields such as computer vision, imaging processing and machine learning were enriched in the last three decades with techniques and approaches that make it possible to gather useful information to improve the execution of the tasks delivered to the robots. Also, the capacity of the electronics industry to produce more efficient and smaller sensors and microprocessors almost every day allied to the economic grow of the consumer market allows for the reduction of the gap between natural and artificial sensors and processors in terms of quality and processing capacity. In this subsection a brief global overview about the imaging processing problem is intended, for which Forsyth and Ponce (2002), Szeliski (2010) and Grauman and Leibe (2011b) were the two main references used.

The processing capacity of the vision system is one of the biggest issues in the imaging processing task. Unlike positioning (global position system, GPS), inertial measurement unit (IMU), and distance sensors (sonar, laser, infrared) cameras produce the highest bandwidth of data. One video camera of very modest resolution yields a bandwidth of almost 140 Mbits/s (I.e.: 30 frames per second with a resolution of 640x480 pixels per frame and color depth of 16 bits per pixel). This amount of data increases when RGBD sensors are used, because the depth component is added to the RGB components. The parallel processing technique is the most common approach proposed by the scientific community to deal with such amount of data. Although, multi core CPU and computers arrays are the most used, CPU Graphics Processing Units (GPU) and field-programmable gate array (FPGA) are being considered for parallel imaging processing. Asano et al. (2009) presents an implementation of an imaging processing task using FPGA, GPU and CPU (quad-core) and concludes that GPU has the best performance during task execution. However, the development of the algorithm is much more complex than simple implementation of the algorithm by programing means, because it also requires hardware design skills from the developer.

Before using any vision system, each pair of camera/lens has to execute a process called camera calibration in order to estimate the intrinsic and extrinsic parameters.

The intrinsic parameters are position of image center in the image, focal length, different scaling factors for row pixels and column pixels, skew factor and lens distortion (pin-cushion effect).

The extrinsic parameters denote the coordinate system transformations from 3D world coordinates to 3D camera coordinates. Equivalently, the extrinsic parameters define the position of the camera center and the camera's heading in world coordinates.

Although in robots this coordinate system transformation is dynamic and estimated in real time based on the robot position and orientation estimation, the relation between the position of the cameras and the point of reference used by the robot has to be determined. These intrinsic and extrinsic parameters are used in the first stage of the imaging processing to correct and to rectify the image.

Depending on the vision system, several techniques can be used for image manipulation and fast image analysis such as linear and non-linear image filtering, geometrical image transformations (resize, affine and perspective warping, generic table-based remapping), color space conversion, and histograms. Linear Filters and Convolution can be applied to images in order to reduce the noise, salient relevant regions of the image or to estimate the value of a missing pixel. It should be indicated that the pattern of weights used for a linear filter is usually denominated the kernel of the filter, and the process of applying the filter is usually referred to as convolution.

Different experiences in building vision systems suggest that very often interesting things are happening in an image at the edges and it is worth knowing where the edges are. In the image, the edges usually define the boundaries of objects or important regions. These edges or edge points are frequently associated to points in the image where brightness changes sharply, even if this is a simple description for the complex definition of an edge. In literature it is possible to find several techniques for edge detection. While some are better than others, all have their pros and cons.

The simplest technique considers that an edge is a boundary between two homogeneous regions. First, a gray image is obtained from the original image and then the following definition is used: the gray level properties of the two regions on either side of an edge are distinct and exhibit some local uniformity or homogeneity among themselves. Then usually an edge is typically extracted by computing the derivative of the image intensity function, and this consists of two parts: magnitude of the derivative (measure of the strength/contrast of the edge) and direction of the derivative vector (edge orientation). The traditional derivative operators used are Roberts, Prewitt, Sobel or Laplacian. However, most of these partial derivative operators are sensitive to noise and their use results in thick edges or boundaries, in addition to spurious edge pixels due to noise.

In contrast, Marr and Hildreth (1980) suggests the use of the "Laplacian of the Gaussian" (LoG) operator to detect edges, which will produced edges as Zero-Crossings in the output function. However, the output does not give any idea of the gradient magnitude or orientation of the edges. A better and widely used approach is suggested by Canny (1986). Canny suggested an optimal operator, which uses the Gaussian smoothing and the derivative function together. He has proved that the first derivative of the Gaussian function is a good approximation to his optimal operator. Smoothing and derivative when applied separately were not producing good results under

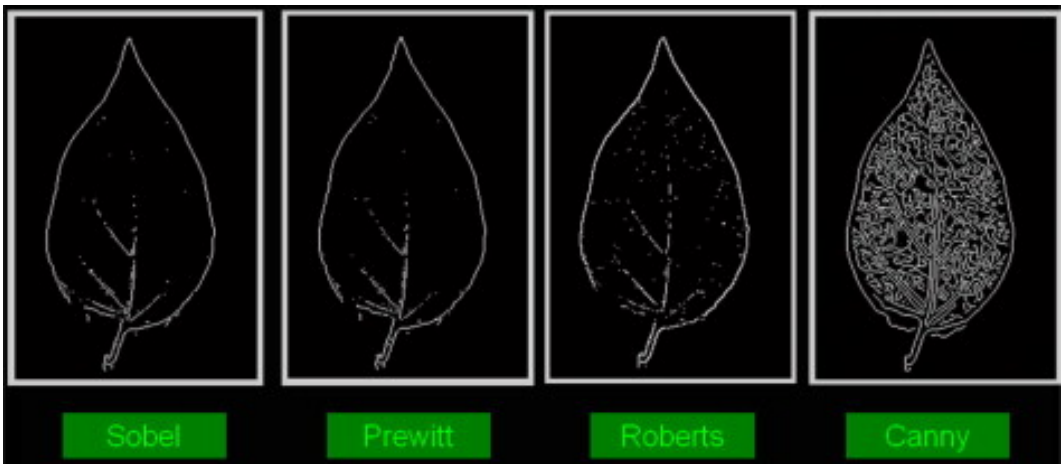

Figure 2.5: Edge detection comparison using different operators, by Husin et al. (2012).

noisy conditions. This happens because one opposes the other; so, combining them both will help to produce the desired output, as shown in figure 2.5. Among these neuro-fuzzy techniques, phase congruency and neural base models are also used to improve the edge detection.

In scenarios made by humans it is common to find structures or objects defined by straight lines or by primitive geometric figures (circles, ellipses). One powerful global method for detecting edges (lines and parametric curves) is the Hough transform. Although the common use of Hough transform is applied to detect parametric straight lines, it can be used to detect other kind of curves and shapes. Indeed, Ballard (1981) introduces the Generalized Hough Transform to detect arbitrary shapes. The Generalized Hough Transform is a modification of the Hough Transform using the principle of template matching. This modification enables the Hough Transform to be used not only to detect an object described with an analytic equation (e.g. line, circle, parabolic, etc.), but also to detect any arbitrary object described with its model. However, the complexity of search time increases exponentially with the number of model parameters, and it is also hard to choose the appropriated grid size for the output.

When the imaging processing is used for object detection, image matching or image stitching it will be required to detect unique points (features) such as corners. A corner can be defined as the intersection of two edges or as a point to which there are two dominant and different edge directions in the neighborhood of the point. Moravec corner detection algorithm is one of the earliest corner detection algorithms and defines a corner to be a point with low self-similarity. Harris and Stephens (1988) improved upon Moravec's corner detector by considering the differential of the corner score with respect to direction directly, instead of using shifted patches; it searches for local neighborhoods where the image content has two main directions (eigenvectors). Harris corner detector is widely used but other corner detectors can be found on literature, such as Hessian corner detector, multi-scale Harris operator, SUSAN corner detector by Smith and Brady (1997), Trajkovic and Hedley corner detector by Trajković and Hedley (1998).

However, these corner detectors do not always detect the best local features (keypoints) in the

images. In order to be possible to make the optimal correspondence between points of two images the best local features have to be obtained. Considering that each image is obtained at different distance, perspective and light condition, the best local features (keypoints) are:

- Invariant to translation, rotation, scale;
- Invariant to affine transformation;
- Invariant to presence of noise and blur;
- Detectable by the detector when occlusion clutter occurs and illumination changes (Locality);
- Detectable all the time by the detector (Repetitive);
- Defined by a rich structure (Distinctiveness);
- Available in an enough number of points in order to represent the image (Quantity); and,
- Simple to process requiring a low computational time in order to be detected;

Lindeberg (1998) suggests a detector for blob-like features that searches for scale space extrema of a scale-normalized Laplacian-of-Gaussian (LoG). Instead of considering the concept of zero crossing, when LoG is used for edge detection, this detector considers the point which gets the maximum value among its 26 neighbors. The LoG can thus both be applied for finding the characteristic scale for a given image location and for directly detecting scale-invariant regions by searching for 3D (location + scale) extrema of the LoG. A similar approach but computationally more efficient is suggested by Lowe (2004). Lowe shows that scale-space Laplacian can be approximated by a difference-of-Gaussian (DoG), which can be more efficiently obtained from the difference of two adjacent scales that are separated by a factor. As in the case of the LoG detector, DoG interest regions are defined as locations that are simultaneously extrema in the image plane and along the scale coordinate of the $D(x,\sigma)$ function. Such points are found by comparing the $D(x,\sigma)$ value of each point with its 8-neighborhood on the same scale level, and with the 9 closest neighbors on each of the two adjacent levels. Several other features (keypoints) detectors have been proposed, such as Hessian/ Harris Laplacian detector by Mikolajczyk and Schmid (2001) and Maximally Stable Extremal Regions (MSER) by Matas et al. (2004) among others which are compared by Mikolajczyk and Tuytelaars (2005).

Once a set of features (keypoints), also described as regions of interest, has been extracted from an image, their content needs to be encoded in a descriptor that is suitable for discriminative matching. The most popular choice for this step is the Scale Invariant Feature Transform (SIFT) descriptor introduced by Lowe (2004). The SIFT is a combination of a DoG interest region detector and a corresponding feature descriptor. This descriptor aims to achieve robustness to lighting variations and small positional shifts by encoding the image information in a localized set of gradient orientation histograms. The descriptor construction is divided into four steps:

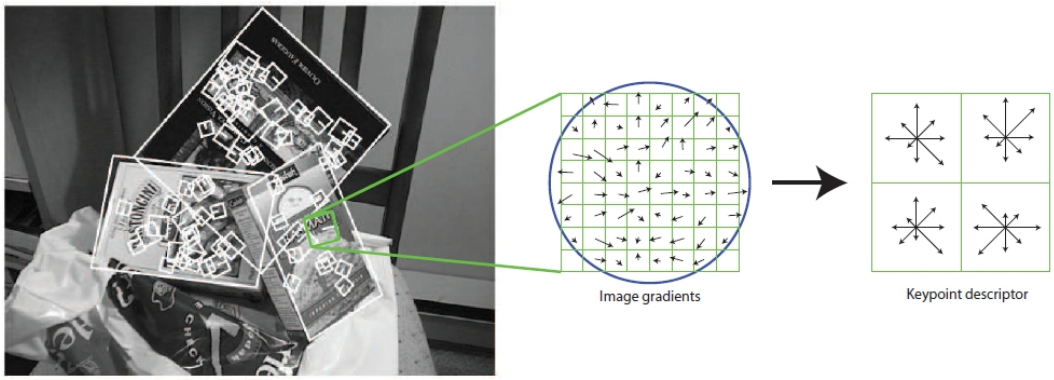

Figure 2.6: Visualization of the SIFT descriptor computation. For each (orientation normalized) scale invariant region, image gradients are sampled in a regular grid and are then entered into a larger 4×4 grid of local gradient orientation histograms (for visibility reasons, only a 2×2 grid is shown here), by Grauman and Leibe (2011b).

1. Scale-space extrema Detection - Detect interesting points (invariant to scale and orientation) using DoG.

2. Keypoint Localization -Determine location and scale at each candidate location, and select them based on stability. Low contrast points and points that lie on the edge are rejected.

3. Orientation Estimation - Use local image gradients to assigned orientation to each localized keypoint in order to preserve theta, scale and location for each feature.

4. Keypoint Descriptor - Extract local image gradients at selected scale around keypoint, as depicted at figure 2.6, and form a representation invariant to local shape distortion and representation invariant to local shape distortion and illumination them.

The SIFT keypoint descriptor is a normalized vector with 128 values which are related to the gradient in the neighborhoods of the keypoint. With this descriptor for each keypoint it is possible to find the best pairs matching among keypoints found in two images. Inspired by the success of SIFT descriptor other researchers have proposed other similar approaches with better performance, such as Speed-Up Robust Feature (SURF) by Bay et al. (2006), Histogram of Oriented Gradient (HOG) by Dalal and Triggs (2005), Gradient Location Orientation Histogram (GLOH), PCA- SIFT by Ke and Sukthankar (2004), Pyramidal HOG (PHOG), Pyramidal Histogram Of visual Words (PHOW) among others. Juan and Gwun (2009) presents a performance comparison between SIFT, SURF and PCA-SIFT.

With the edges, corners, basic shapes, keypoints, regions of interest and keypoint descriptors detected by the previous techniques it is possible to perform visual object recognition, image matching and place classification, as detailed by Forsyth and Ponce (2002), Szeliski (2010) and by Grauman and Leibe (2011b). In vision systems the capacity to classify, train and learn usually appears associated to the use of *Support vector machine (SVM)* and neural networks techniques, as shown in Robles-Castro et al. (2011) Grauman and Leibe (2011a) and by Rasolzadeh et al. (2009).

Scale-invariant key points and SIFT features have been used by Kosecká and Li (2004). They perform global localization indoors by recognizing locations and exploit information from neighborhood relations from a map using Hidden Markov Models. Wolf et al. (2002) propose a similar approach. Others have used histogram descriptors to represent the scenes. Ulrich and Nourbakhsh (2000) compute color histograms from omnidirectional camera images and match them to stored images in combination with predictions from a topological map. In that way near-real-time performance is obtained through a simple voting process over the color bands. The method is successfully applied to indoor as well as outdoor environments. Davison and Murray. (2002) used actively controlled cameras to find landmarks indoors following Bajcsy's active perception paradigm. Chang et al. (2010) present a vision-based navigation and localization system using two biologically-inspired scene understanding models which are studied from human visual capabilities, gist model which captures the holistic characteristics and layout of an image and saliency model which emulates the visual attention of primates to identify conspicuous regions in the image. Labbani-Igbida et al. (2011) present a method for spatial representation, place recognition and qualitative self-localization in dynamic indoor environments, by using omnidirectional images and in spatial representation built up from invariant signatures based on Haar invariant integrals.

2.3 Map representations

In general a robot senses the continuous world, processes the sensed information and stores the processed information in order to reuse it in the future. So, a correct and complete representation of the world, through environment models, in the robotic knowledge structure is a very important resource because it can improve the reliability, flexibility and efficiency of Localization/Mapping and Mission/Task Plan tasks of the robotic system.

The knowledge of the structure and the current state of the world is usually encoded in the form of a *map*. The problem of how to represent, build, and maintain maps has been one of the most active areas of research in robotics, since Smith and Cheeseman (1986) and Leonard and Durrant-Whyte (1991). This problem is also usually associated to Simultaneous Localization and Mapping (SLAM) problem, as detailed in section 2.4.

Most of this research is based on geometrical and topological representations of the spatial structure of the environment, such as: metric maps which capture the geometric properties of the environment, by Elfes (1989) and Chatila and Laumond; topological maps which describe the connectivity of different places, by Mataric (1990) and Kuipers and Byun (1991); or appearance-based maps, by Kröse et al. (2000). Hybrid maps is another approach based in the combination of metric and topological maps, by Buschka and Saffiotti (2004). In the current state of the art very valuable solutions are now available for robots which must plan and navigate to a given numeric point, avoiding collisions, and sometimes mapping or updating the unknown environment.

The first representation of the world in the robotics field was done using metric maps (occupation grids 2D and 3D) whilst more recently the use of a higher level of map representation is considered, such as semantic mapping. The metric maps are closer to sensor observation and

semantic maps are more abstract and more further away from sensor observation. Semantic mapping makes the reasoning about the map possible and makes the association of complex dynamics to the objects possible. Both can coexist and improve the quality of the world representation, as shown in chapter 3. This section will detail the definition and construction of occupancy grid map, features based maps, topological maps and octomaps.

2.3.1 Topological maps

The mobile robotics literature often distinguishes topological from metric representations of space. While no clear definition of these terms exists, topological representations are often thought of as course graph-like representations, where nodes in the graph correspond to significant places (or features) in the environment. For indoor environments, such places may correspond to intersections, T-junctions, dead ends, and so on. The resolution of such decompositions, thus, depends on the structure of the environment. Alternatively, one might decompose the state space using regularly paced grids. Such decomposition does not depend on the shape and location of the environmental features.

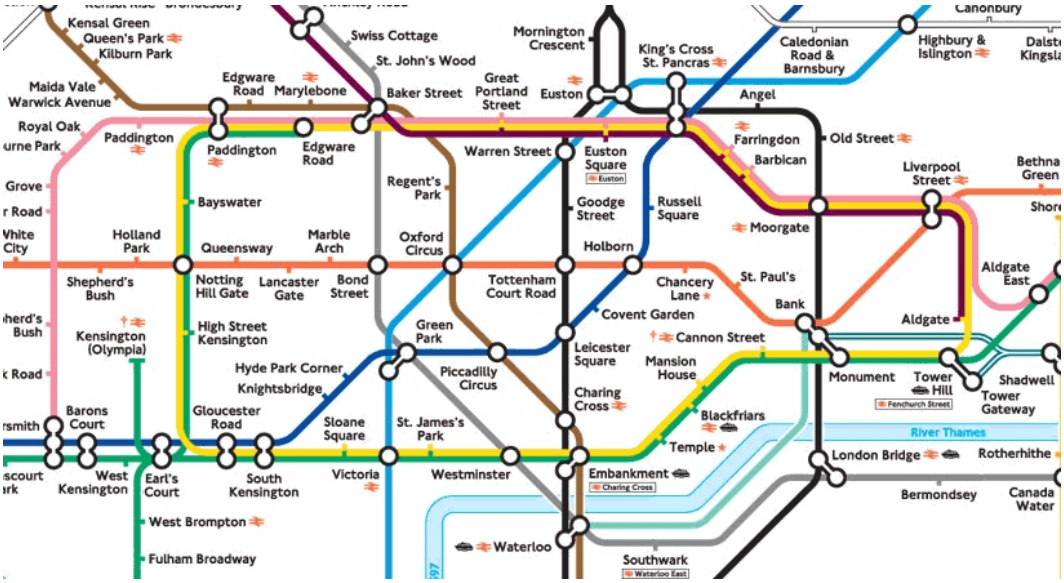

Figure 2.7: The London tube map is an example of a topological map. A topological map contains less metric details and is centered in describing the main places and their connectivity.

A topological map is or can be an abstraction of an occupation grid-map, usually this map segments the 3D space into main places and relates these places by connections. Compared to a metric map, it contains less 3D space details and models the environments as a list of significant places/nodes, which are connected via arcs/edges. The London tube map is good example of a topological map, where the metric geometry is lost but the essential information remains (the main places and their connectivity) , as depicted in figure 2.7. In the state of the art, arcs are usually annotated with information on how to navigate from one place to another. In synthesis,

the main contribution of the topological map is to simplify the metric map and maintain only vital information for the definition of places and connections (the arcs).

Since Mataric (1990) and Kuipers and Byun (1991), several approaches using topological maps have been found in the robotics domain, mainly to simplify the navigation task but also to help solving the problem of robot localization and mapping. Byeong-Soon and Hyun Seung (1999) present the ETMap (Enhanced Topological Map), a topological structure where nodes were landmark places (door, open (junction)) and arcs are adjacency links (with rough metrical information as length and orientation). The ETMap was built up with sonars and odometry, and several limitations were found in curved corridors and outdoor navigation. In contrast, Duckett and Nehmzow (1999) presents a map-based exploration system, in which a topological map of the environment is acquired incrementally by the robot, using artificial neural networks to detect new areas of unexplored territory. Using this approach, no manual intervention in the map acquisition process is required and all computation is carried out in real-time on board of the robot. They suggest also a novel approach where the addition of a place node has two stages, predicted and confirmed.

Others works, Angeli et al. (2008), Mozos (2008), Rady et al. (2010), Paul and Newman (2010) and Gu and Chen (2011) have constructed a topological map from the observation obtained from vision systems. Wang et al. (2009) proposed the construction of a topological map based on laser's free beams (free space) and visual scale-invariant features. This was used to perform navigation without global localization. The non-existence of a precise global pose, in the world reference frame, induced several limitations and could fail in spacious surroundings, so an hierarchical map, composed by a grid and a topological map was proposed. He et al. (2006) uses monocular image sequences to build the topological map of environment. The problem of extraneous objects, seasonal and lighting variations in the image representation was considered. The spatial relationships between features in each image are ignored, although they convey considerable location information. Due to that, some data points on the manifold can easily collapse into a tight cluster, because of noise in the learned representation, or local minima in the embedding procedure. Booij et al. (2007) presents appearance topological maps based on visual information, from an omnidirectional vision system. Localization was performed using this map. An epipolar geometry and a planar floor constraint in the heading computing enabled the robot to drive robustly in a large environment. This work did not take into account dynamic objects and people. In Liu et al. (2009), the localization and topological mapping were based on scene recognition to build a topological map of the environment and perform location-related task. The scene recognition process uses an adaptive and lightweight descriptor for omnidirectional vision named FACT (Fast Adaptive Color Tags), making it possible to add nodes to a topological map automatically and solve the localization problem of the mobile robot in real time.

In contrast, Schmidt et al. (2006),Joo et al. (2010) and Choi et al. (2011) embraced the challenge of building the topological map from grid-maps, this subject is detailed in the chapter 4.

2.3.2 Metric maps

Grid-based or feature-based approaches are two common metric map representations and they are widely used by simultaneous localization and mapping approaches. Metric maps are supposed to represent the environment geometry quantitatively correctly, up to discretization errors.

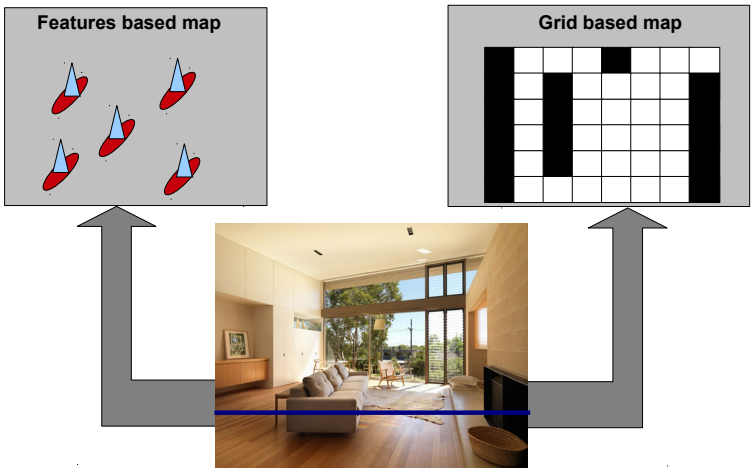

Figure 2.8: Feature-based map vs Grid-based map.

Historically, Elfes (1989) was the first to present an occupancy grid mapping algorithm, where a map was represented by a fine-grained grid that modeled the occupied and free space of the environment. The grid-based approaches are useful for local navigation planning tasks, but they have a high computational cost in feature matching.

In the feature-based (or geometric) approaches, the environment is represented with a set of geometric beacons (or Artificial/Natural features/landmarks) and the position of the robot is estimated by matching the sensed features against the world model. Those algorithms are vulnerable to sensing errors and environmental uncertainties in the past because they rely on precise metrical information.

Sebastian Thrun et al. (2005) formally describes a map $\mathcal{M}$ as a list of objects in the environment along with their properties:

$$\mathcal{M} = \{m_1, m_2, ..., m_N\} \tag{2.3}$$

Here N is the total number of objects in the environment, and each m_n with $1 \leq n \leq N$ specifies a property. Maps are usually indexed in one of two ways, known as feature-based and location-based.

In feature-based maps, n is a feature index. The value of m_n contains, next to the properties of a feature, the Cartesian location of the feature.

In location-based (or Grid-based) maps, the index n corresponds to a specific location. In Grid-based (planar) maps, it is common to denote a map element by $m_{x,y}$ instead of m_n, to make explicit that $m_{x,y}$ is the property of a specific world coordinate (x, y).

An occupancy grid map is a classical metric map representation, figure 2.9, and is based on the idea that occupancy grids represent the map as a field of random variables, arranged in an evenly spaced grid. They assign to each (x,y) (or even in 3D (x,y,z)) coordinate a binary occupancy value which specifies whether or not a location is occupied with an object. Occupancy grid maps are great for mobile robot navigation: they make it easy to find paths through the unoccupied space.

So, let m_i denote the grid cell with index i. If an occupancy grid map is partition of the space into finitely many grid cells, then:

$$m = \sum_i m_i \tag{2.4}$$

Each m_i has attached to it a real number which defines the occupancy value; this specifies whether a cell is occupied or free. Usually when this cell takes the value 1 it means that it is an occupied cell and when it takes 0 it means that it is an empty cell. Usually, the notation $p(m_i = 1)$ or $p(m_i)$ denotes the probability of a grid cell being occupied.

The gold standard of any occupancy grid mapping algorithm is to calculate the posterior over maps given the data:

$$p(m|z_{1:t}x_{1:t}) \tag{2.5}$$

where m is the map, $z1 : t$ the set of all measurements up to time t, and $x1 : t$ is the set of robot poses from time 1 to t.

The types of maps considered by occupancy grid maps are fine-grained grids defined over the continuous space of locations. By far the most common domain of occupancy grid maps are 2-D floor plan maps, which describe a 2-D slice of the 3-D world as depicted in figure 2.9. 2-D maps are often sufficient, especially when a robot navigates on a flat surface and the sensors are mounted so that they capture only a slice of the world.

Occupancy grid map representations are often thought of as metric although, strictly speaking, it is the embedding space that is metric, not the decomposition. In mobile robotics, the spatial resolution of grid representations tends to be higher than that of topological representations.

Occupancy grid based techniques can be generalized to 3-D representations, but at significant computational and memory expenses. However, the requirement of 3D detailed maps of the environment and appearance of RGB-D sensor and 3D Laser Range Finder at more accessible cost have motivated the optimization of the techniques that manages these 3D occupancy grid based maps. Instead of considering an occupancy grid map with a constant size for each cell, these new techniques consider different sizes and depth of detail for each cell.

For example, Hornung et al. (2013) have suggested a new optimized 3D occupancy grid mapping approach based on octree data structure which also considers a probabilistic occupancy estimation. They have made available to the community the *OctoMap library* which implements the suggested 3D occupancy grid mapping approach and provides data structures and mapping algorithms.

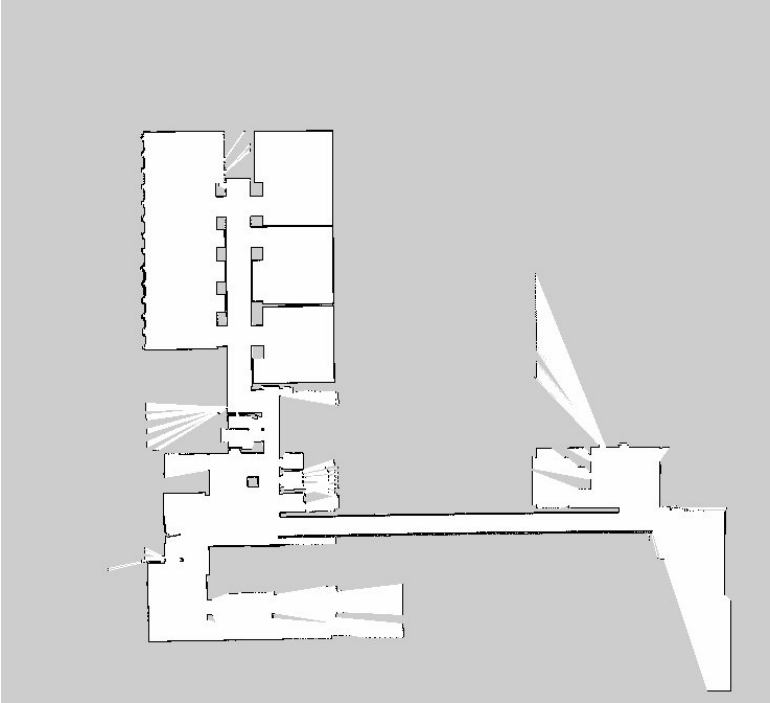

Figure 2.9: An occupancy grid-map built by a robot using localization and mapping technique. This map was obtained at the Faculty of Engineering of the University of Porto, by a robot with ROS, Laser range finder and the Hector SLAM package.

An octree is a hierarchical data structure for spatial subdivision in 3D (Meagher (1982); Wilhelms and Gelder (1992)). Each node in an octree represents the space contained in a cubic volume, usually called a voxel. This volume is recursively subdivided into eight sub-volumes until a given minimum voxel size is reached. The minimum voxel size determines the resolution of the octree. Since an octree is a hierarchical data structure, the tree can be cut at any level to obtain a coarser subdivision if the inner nodes are maintained accordingly.

Octree nodes can be extended to store additional data to enrich the map representation. Voxels could, for example store terrain information, environmental data such as the temperature, or color information. Each additional voxel property requires a method that allows several measurements to be fused. This approach has advantages over other 3D representations, such as Point Clouds, elevation maps and multi-level surface maps, figure 2.10. These advantages are memory-efficiency, differentiation between obstacle-free and unmapped areas and means of fusing multiple measurements probabilistically.

2.4 The SLAM Problem and approaches

The traditional SLAM problem without the semantic map has been largely studied since Leonard and Durrant-Whyte (1991) and in the last two decades tremendous progress in the field has been

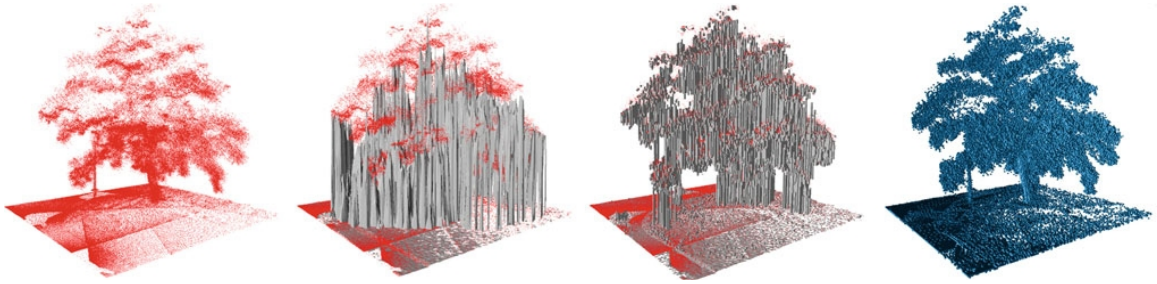

Figure 2.10: 3D representations of a tree scanned with a laser range sensor (from left to right): Point cloud, elevation map, multi-level surface map, and the volumetric (voxel) representation suggested by Hornung et al. (2013). The volumetric representation explicitly models free space but for a better perception only occupied volumes are visualized.

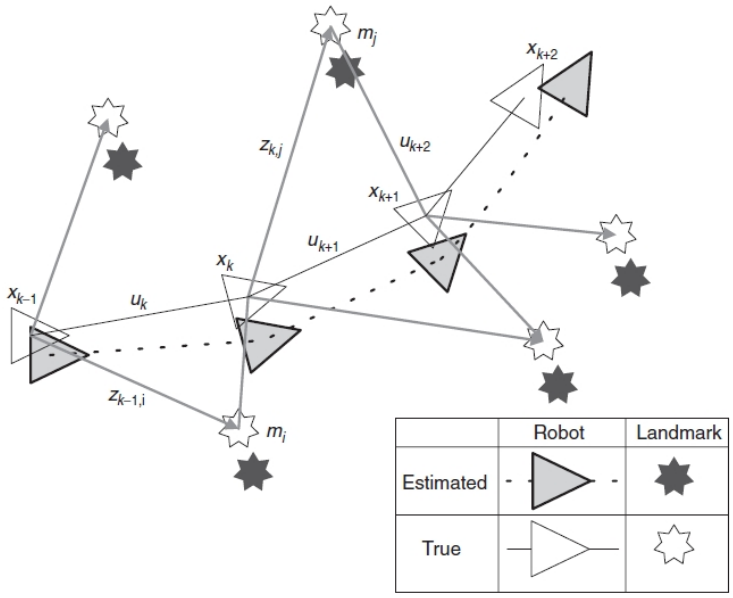

Figure 2.11: The essential SLAM problem. A simultaneous estimate of both robot and landmark locations is required. The true locations are never known or measured directly. Observations are made between true robot and landmark locations, by Durrant-Whyte and Bailey (2006)

seen, most of which is detailed by Durrant-Whyte and Bailey (2006). Dissanayake et al. (2001) describes the SLAM problem using this sentence : *The simultaneous localization and map building (SLAM) problem asks if it is possible for an autonomous vehicle to start in an unknown location in an unknown environment and then to incrementally build a map of this environment while simultaneously using this map to compute absolute vehicle location.*

Sebastian Thrun et al. (2005) considers the localization and mapping task a "chicken-and-egg" problem, reason for which it is often referred to as the simultaneous localization and mapping (SLAM) or concurrent mapping and localization problem. When the robot moves through its environment, it accumulates errors in odometry, making it gradually less certain as to where it is.

In the absence of both an initial map and exact pose information, even in SLAM approaches, the robot has to do both: estimate the map and localize itself in relation to this map. It is an inherently complex problem since an error on the robot pose leads to an error on the map and vice versa, as depicted in figure 2.11.

In literature there is intense work with 2D SLAM, horizontal representation. But in the last years, some groups have been using a pitching or rotating *laser range finder (LRF) 2D* sensor resulting in 3D data which supplies richer information; this kind of approach is used by Ruiz-del Solar et al. (2011) Lee and Song (2011) and Miettinen et al. (2007). The laser range finder is widely used due to its robustness, accuracy and quantity of information available. However, its dimension, cost and lack of color information implies the use of other solutions, when low cost or color information are required. In that line of work, Srinivasan (2010) Klippenstein and Zhang (2009) Lee and Song (2010) and Aghili (2010) have used artificial vision systems to detect artificial or natural visual features, which are used to perform SLAM. A step forward, Lv and Zhang (2011) proposed a SLAM approach based on data provided by a vision sensor and LRF.

The Kalman-filter and particle-filter theories are present in the majority of the approaches that solve the SLAM problem efficiently. The success of these approaches depends largely on the accuracy of the sensors and the capacity to extract unique features from the world through the sensors.

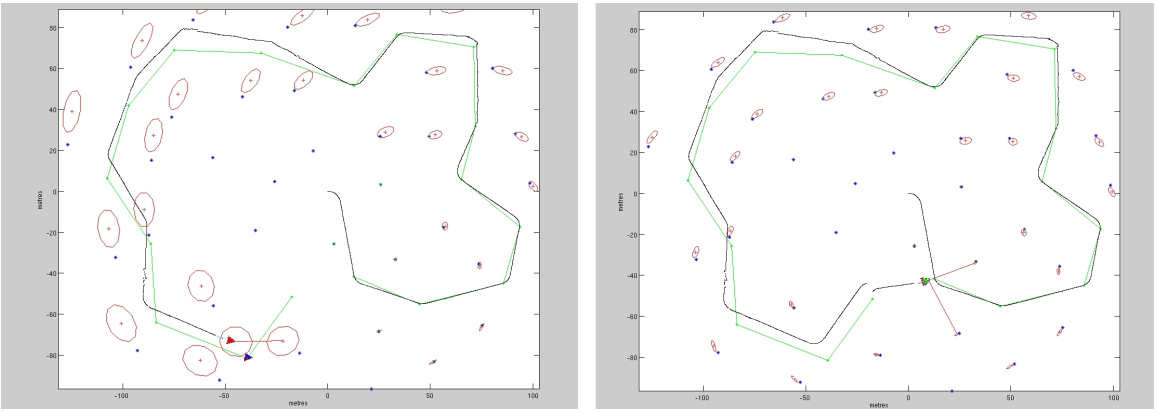

Figure 2.12: The EKF-SLAM map of features with uncertainty associated before and after the close loop, by Kasra Khosoussi (2009)

The two most well known simultaneous mapping and localization strategies based on kalman-filter are FastSLAM and EKF-SLAM, both relying on feature-based maps. In these approaches, the environment is modeled as a discrete set of features, such as lines corners and other features extractable from the external world observation obtained from the internal sensors of the robot. Each feature is described by a number of continuous state variables. The standard solution is to take a Bayesian approach, explicitly modeling the joint probability distribution over possible vehicle trajectories and maps.

In EKF-SLAM the joint probability is linearized and represented as a single high-dimensional

Gaussian. The EKF-SLAM is a variant of the Extended Kalman Filter and uses only one state matrix representing the vehicle state and the landmarks of the feature map. This state matrix is increased every time a new feature is found. Newman and Leonard (2002) proved in a moderate-scale indoor implementation the nondivergence properties of EKF-SLAM by returning a robot to a precisely marked starting point without seeing the robot and only guiding it by the feature-based map. Nonetheless, the EKF-SLAM solution is computationally heavy, quadratic with a number of features N, $O(N^2)$. In dynamic or large scenarios, such as malls, this map can be increased continuously, thus becoming non-applicable when a real time performance is a requirement. These aspects can be become worst if the dimension of the space increases, such as passing from 2D to 3D.

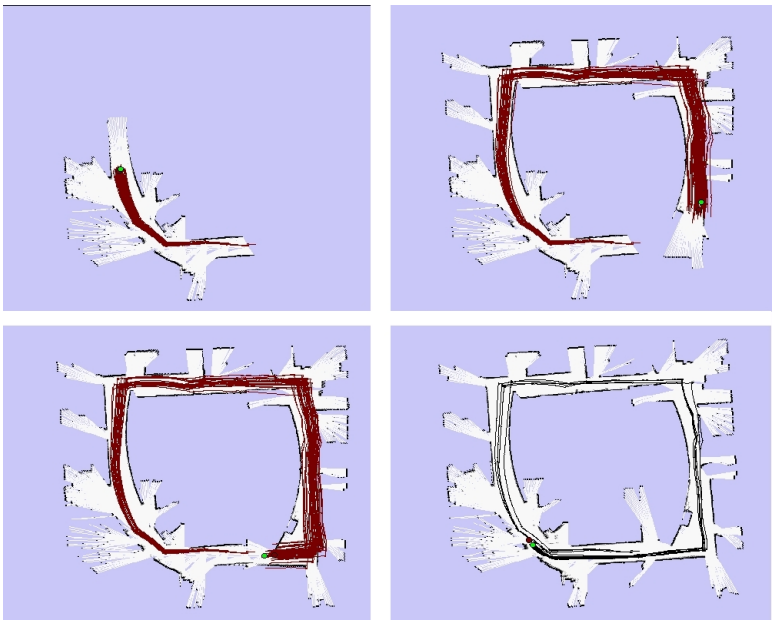

Figure 2.13: SLAM mapping using Rao-Blackwellised particle filters at four different times, by Hahnel and Burgard (2003)

In contrast, the FastSLAM, introduced by Montemerlo et al. (2002), marks a fundamental conceptual shift in the design of recursive probabilistic SLAM. While other works have remained with the essential linear Gaussian assumptions, the FastSLAM based on a recursive Monte Carlo sampling, or particle filtering, made it possible for the first time to represent directly the nonlinear process model and non-Gaussian pose distribution. The high dimensional state-space of the SLAM problem makes direct application of particle filters computationally infeasible. However, it is possible to reduce the sample space by applying the Rao-Blackwellization technique. The FastSlam approach can be seen as a robot position and a collection of N landmarks position estimation problems. Each particle which represents different robot states has its own robot pose estimation and each tiny state matrice represents each landmark position of the feature-based map. The FastSlam has a lower computational complexity, when compared with the EKF-SLAM, $O(MLogN)$, with M being the number of particles and N the number of landmarks.

Hahnel and Burgard (2003) based on FastSLAM concepts, presents another SLAM approach that combines Rao-Blackwellized particle filtering and scan matching. They use a scan matching technique in order to minimize the odometric errors during the mapping process. The Rao-Blackwellized particle filter is used to estimate a posterior of the path of the robot, in which each particle has associated to it an entire map. Figure 2.13 shows the occupancy grid-map construction and the convergence of the estimation when the robot revisits the start point.

Brooks and Bailey (2008) integrates the concepts used for FastSLAM and EKF-SLAM into a single approach under the name HybridSLAM. This SLAM approach combines the strengths and avoids the weaknesses of FastSLAM and EKF-SLAM.

Grisetti et al. (2007) suggests the GMapping approach which takes raw laser range data and odometry. GMapping is a highly efficient Rao-Blackwellized particle filer to learn grid maps from laser range data. This approach uses a particle filter in which each particle carries an individual map of the environment. Accordingly, a key question is how to reduce the number of particles. They present adaptive techniques for reducing this number in a Rao-Blackwellized particle filter for learning grid maps. An approach is also put forward to compute an accurate proposal distribution, taking into account not only the movement of the robot but also the most recent observation.

Dirk et al. (2003) suggests the GridSLAM, a fastSLAM implementation, which is based on a Rao-Blackwellized particle filter to learn grid maps from laser range data. This work presents algorithms that combine Rao-Blackwellized particle filtering and scan matching. In this approach the scan matching is used for minimizing odometric errors during mapping. A probabilistic model of the residual errors of the scan matching process is then used for the resampling steps. This way the number of samples required is seriously reduced. There is a simultaneous reduction of the particle depletion problem, which typically prevents the robot from closing large loops.

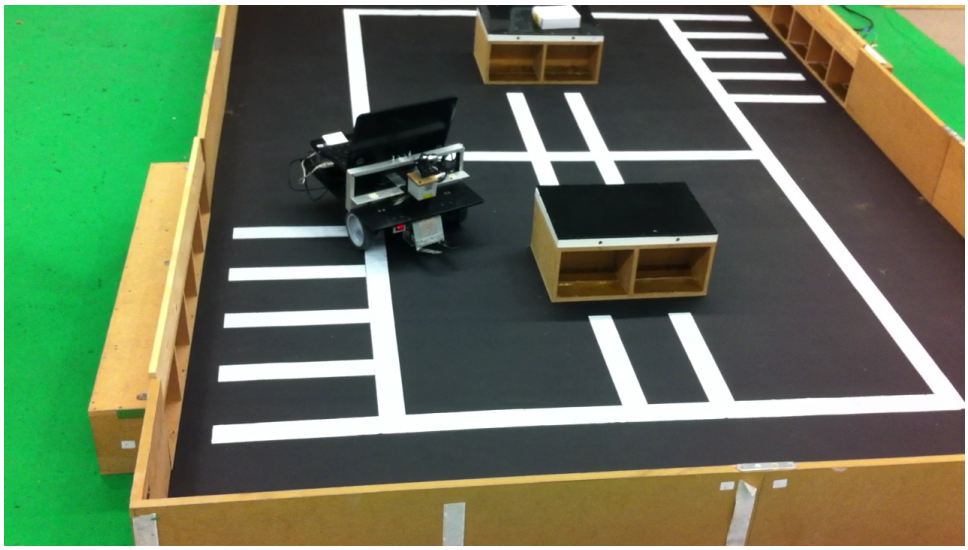

Figure 2.14: Xica Robot at Robot@factory field. This robot has one Laser Range Finder, one webcam and four distance sensors.

Motivated by the application of robots in Urban Search and Rescue (USAR) scenarios where robots need to learn a map of unknown environments, Kohlbrecher et al. (2011b) presents an optimized system for fast online learning of occupancy grid maps, called Hector SLAM. It combines a robust scan matching approach using a LIDAR system with a 3D attitude estimation system based on inertial sensing. By using a fast approximation of map gradients and a multi-resolution grid, reliable localization and mapping capabilities are obtained in a variety of challenging environments even when the system is used as a handheld mapping system.

Hector SLAM approach was made available as an open source package for ROS. During the works of this thesis this approach was being tested in a small robot (figure 2.14) at the first floor of the building I of the Faculty of Engineering of University of Porto, the output result is on figure 2.9.

Eliazar and Parr (2003) suggests that the DP-SLAM aims to achieve truly simultaneous localization and mapping without landmarks. While DP-SLAM is compatible with techniques that correct maps when a loop is closed, in literature it is stated that DP-SLAM is accurate enough that no special loop closing techniques are required in most cases. DP-SLAM makes only a single pass over the sensor data, and works by maintaining a joint probability distribution over maps and robot poses using a particle filter. This allows maintaining uncertainty about the map over multiple time steps until ambiguities can be resolved. This prevents errors in the map from accumulating over time.

Huang and Wang (2008) suggests the I-SLSJF SLAM which is a local submap joining algorithm for building large-scale feature based maps. The algorithm is based on the recently developed Sparse Local Submap Joining Filter (SLSJF) and uses multiple iterations to improve the estimate and hence is called Iterated SLSJF (I-SLSJF). The input to the I-SLSJF algorithm is a sequence of local submaps. The output of the algorithm is a global map containing the global positions of all the features as well as all the robot start/end poses of the local submaps.

The previous works are essentially based on 2D sensors and consequently the output of these approaches is a 2D occupancy grid map. However the world has a 3D representation and sometimes a 3D map is required by the robot to avoid collisions and plan the better strategy to reach the target (with the robot arms and by robot motion).

Pinto et al. (2013b) proposes an approach which has two stages - the 3D mapping stage and localization stage. The 3D mapping stage uses a typical 2D SLAM process for the robot trajectory estimation and uses a system with a laser range finder tilting (from -45 to +45 degrees) for the point cloud acquisition which will be used to construct the 3D map. The 2D SLAM is based on the EKF-SLAM approach, using natural features such as 2D lines (extracted from walls, doors or furniture observation) and invariant points to the robot motion (corners and columns). During the 3D mapping process the robot is guided through the environment and revisits previous points, all data (time, odometry and laser range finder observations) being recorded. With the recorded data the robot trajectory is extracted using the 2D SLAM approach. With this robot trajectory and laser range observations the 3D occupancy grid map is constructed. This 3D mapping stage happens only once, when the robot is inserted in the environment. Afterwards, the robot is able to estimate

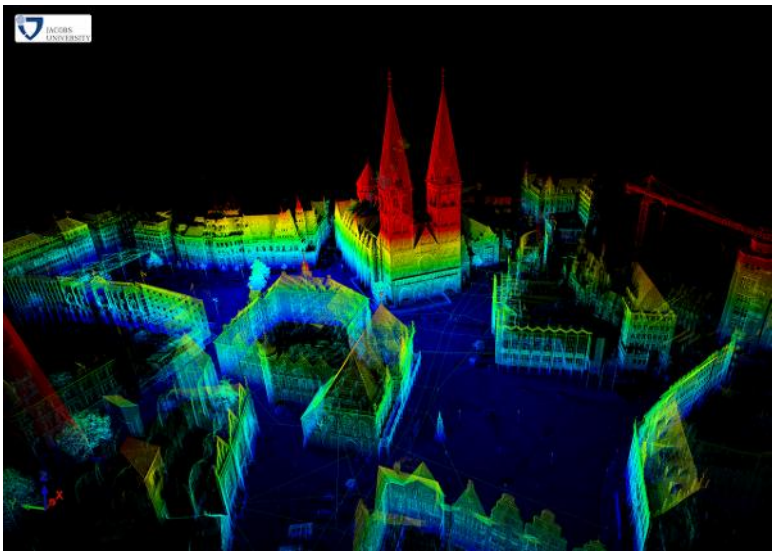

Figure 2.15: A 3D map of Bremen city obtained using the 6D SLAM, available at 3DTKSite (2013)

its location by using the odometry information and correct that estimation through a 3D matching process which uses the laser range finder observations and the upper side of a building.

In contrast, Nüchter (2009) proposes the 6DoF SLAM approach. Instead of considering the localization problem as an estimation of three variables (x, y, θ), horizontal position and heading, it considers it an estimation of six variables $(x, y, z, \theta, \phi, \psi)$, 3D position and orientation (pitch, roll and yaw). Also, instead of basing the SLAM approach on feature extraction and matching, it bases it on a matching optimization problem between successive observations. For the matching optimization problem, the 6DoF SLAM uses a very fast ICP (iterative closest/corresponding points) algorithm. The ICP algorithm requires a 3D map, a 3D point cloud observed (acquired by 3D acquisition system) and the motion estimation (acquired from motion sensors, such as odometry, IMU). From this motion estimation, it is possible to find an approximated 3D transformation (R,t) which makes it possible to project the point cloud into the 3D map, where R is the rotation matrix and t the translation vector. From here, the algorithm will relate each point of the observed point cloud to a point of the 3D map. This relation is based on the rule of the nearest distance. Then, the cost function is created, which includes the distance for each pair of points, and then the algorithm iteratively tries to improve the 3D transformation (R,t) in order to minimize the result of the cost function. In each improvement of the 3D transformation (R,t) the ICP algorithm calculates a new point correspondence. The corrected 3D transformation (R,t) appears when the local/global minimum of the cost function is reached. With the corrected 3D transformation (R,t) it is possible to update the 3D map with this point cloud. So, when this approach gets 3D point clouds with a rough 6 DoF pose estimation, it will create a consistent map without further manual interference and it will self-localize in the six degrees of freedom $(x, y, z, \theta, \phi, \psi)$. In figure 2.15 the 3D map obtained in a test case using the 6DoF SLAM is shown.

Endres et al. (2012) proposes the RGBDSLAM approach, which uses as main sensor a RGB-D camera (an hand-held Kinect-style camera). The RGB-D sensor is like a normal camera, except that it supplies the distance for each observed pixel, which allows to construct colored 3D maps. Instead of optimizing the 3D transformation (R,t) by matching successive point clouds as 6DoF SLAM, the RGBDSLAM approach uses visual features (SURF or SIFT) to match pairs of acquired images, and uses RANSAC to robustly estimate the 3D transformation (R,t). To achieve online processing, the current image is matched only versus a subset of the previous images. Subsequently, it constructs a graph whose nodes correspond to camera views and whose edges correspond to the estimated 3D transformations. The graph is then optimized with HOG-Man, by Grisetti and Kummerle (2010), in order to reduce the accumulated pose errors.

Chapter 3

Extending SLAM to Semantic mapping

Robots that will work alongside us should have a compatible perception about the world to ours, because they are going to manipulate objects and navigate through places, that are structured by humans. So, object interaction, perception, manipulation and strong communication abilities are required for the next generation of robots. To make this possible, we will need to extend geometric and/or topological based mapping and localization to semantic mapping, towards a common cognitive workspace.

The previous chapter described how a robot senses the world; this chapter will address the fundamental question: How to make the robot knowledge closer to the human description. Section 3.1 presents the literature review related to the fundamental question, and section 3.2 details the proposed approach for semantic mapping, called Hybrid localization and mapping (HySeLAM).

3.1 Literature review

Humans use words, which are abstract symbols, to describe the world to other humans. Also, these words are very useful to describe and ask for actions, in the world, to other humans. For instance, Siskind (2011) suggests three major language functions allowing humans: (i) to describe what they perceive, (ii) to ask others to perform a certain action and (iii) to engage in conversation.

In human-robot interaction contexts, it is important for robots to understand the meaning of these words to allow humans to tell them what to do. In order for robots to understand sentences from human teammates, they must be able to identify correspondences between words, elements of language, and aspects of the external world, as depicted in figure 3.1.

Hence, the the question emerges: How will the robot associate the human words to the correct object and/or place or action? The answer lies in a mapping solution, which maps words into aspects of the external world, and vice versa. This mapping solution, which Harnad (1990) called the symbol grounding problem, has been studied since the early days of artificial intelligence.

Since Winograd (1971), many authors have manually created symbol procedures that map between language and the external world, connecting each term onto a pre-specified action space and set of environmental features (Kuipers (1978); Bugmann et al. (2004); Roy (2005) MacMahon

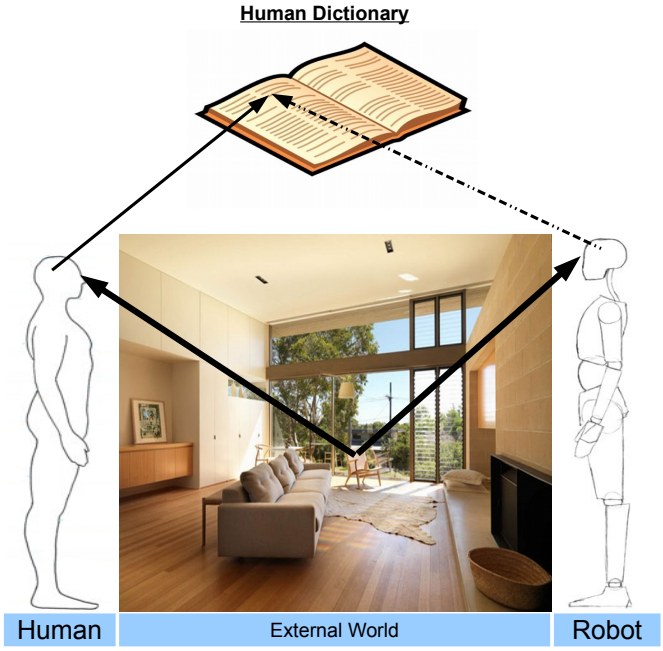

Figure 3.1: How will the robot associate the human words to the correct object and/or place or action?

et al. (2006); Hadas et al. (2008); and Dzifcak et al. (2009)). This approach takes advantage of the structure of spatial language. However, this usually does not involve learning, has little perceptual feedback, and has a fixed action space.

In contrast to these views, other works have looked into studies on child language acquisition, to understand the link by which words are connected with objects and actions. For example, Modayil and Kuipers (2007) describes how a physical robot can learn about objects from its own autonomous experience in the continuous world. The robot should develop an integrated system for tracking, perceiving, categorizing, and acting on objects. They claim that this is a key step in the larger agenda of developmental robotics, which aims to show how a robot can start with the "blooming, buzzing confusion" of low-level sensorimotor interaction,and can learn higher-level symbolic structures of commonsense knowledge. Other works which involve learning the meaning of words in the sensorimotor space (e.g., joint angles and images) of the robot, are shown by Marocco et al. (2010) , and by Sugita and Tani (2005). By treating linguistic terms as a sensory input, these systems must learn directly from complex features extracted by perceptual systems, resulting in a limited set of commands that they can robustly understand.

Other approaches are based in a learning process in order to convert from language onto aspects of the environment. These approaches may use only linguistic features (Ge and Mooney. (2005); Shimizu and Haas (2009)), spatial features (Regier (1992)) or linguistic, spatial and semantic features (Branavan and Chen (2009); Kollar et al. (2010a); Matuszek et al. (2010); Vogel and Jurafsky (2010);Tellex et al. (2011); and Matuszek et al. (2012)). These approaches learn the

meaning of spatial prepositions (e.g.,"above" Regier (1992)), verbs of manipulation (e.g., "push" and "shove" Bailey (1997)), and verbs of motion (e.g., "follow" and "meet" Kollar et al. (2010b)) and landmarks (e.g., "the doors" Kollar et al. (2010a)).

In contrast to these works, which do not define a clear structure of knowledge representation and do not grow bottom-up, others have started to define a structure for semantic mapping and space representation. In a reflection work, Dellaert and Bruemmer (2004) have proposed an extension of the FastSLAM approach by adding semantic information about the environment to each particle's map, which they call Semantic SLAM. They state that the main research challenges to be addressed are those of developing an appropriate representation, along computationally tractable algorithms. As we will see during this work, this makes sense because we will include information that comes from a human-robot interaction, which can be incongruent and is not all the time reliable. So a correct representation will simplify the process of describing the knowledge stored in the robot to the human and also to acquire knowledge from the dialog with a human, while discarding wrong associations.

In this line of work, Oberlander and Uhl (2008) proposed a SLAM algorithm based on FastSLAM 2.0 that maps features representing regions with a semantic type, topological properties, and an approximate geometric extent. The resulting maps enable spatial reasoning on a semantic level and provide abstract information, allowing for semantic planning and a convenient interface for human-machine interaction. In contrast, Nieto-Granda et al. (2010) claims a system capable of reasoning about spaces, which does not builds a topological map on top of a metric map. Instead, they suggest a continuous classification of the metric map into semantically labeled regions. The symbol grounding process happens when the human takes the robot on a tour of the space (either by driving the robot manually, or using a person following behavior), and teaches it by typing the appropriate label for the space that it is currently in. The regional analysis technique is to take a laser scan measurement, fit a Gaussian to the resulting points, and store the mean and covariance in the map along with the label provided by the human. Although this is an efficient way to tag places with the correct human words, this approach does not infer the place delimitation from higher features, as walls and objects, and neither can it infer the place names from a human description of the place.

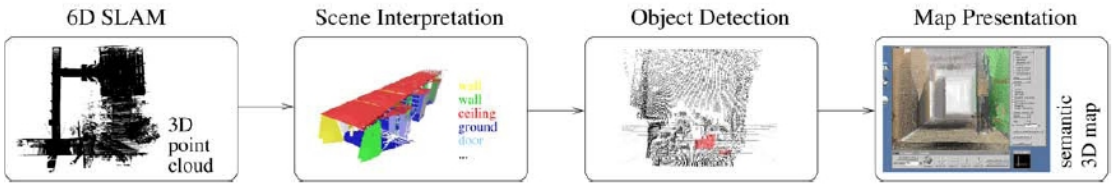

Figure 3.2: System overview from Nüchter and Hertzberg (2008): From left to right: 6D SLAM acquires and registers a point cloud consisting of multiple 3D scans; then there is a scene interpretation which labels the basic elements in the scene; after that object detection locates previously learned objects in 6 DoF, and finally the semantic map is presented to the user.

A step further, Nüchter and Hertzberg (2008) suggest a system for scene understanding. This system uses a 3D laser scanner as prime sensor. Individual scans are registered into a coherent 3D geometry map by 6D SLAM. Coarse scene features, as walls and floors in a building, are determined by semantic labeling. More delicate objects are then detected by a trained classifier and localized. In the end, the semantic maps can be visualized for human inspection; the process is detailed in figure 3.2. Although their system can infer and label high level features as door, floor, ceiling and walls, it can not detect and label the places names.

Other authors have focused in systems that recognize objects present in the real world, by the apprehension of visual representations of objects and/or integration of these semantic concepts with other robot behaviors (Meger et al. (2008); Civera et al. (2011); Jebari et al. (2011b); and Rusu (2009)). For example, Meger et al. (2008) claim a intelligent system that attempts to perform robust object recognition in a realistic scenario, where a mobile robot moves through an environment and uses the images collected from its camera directly to recognize objects. To perform successful recognition in this scenario, they have used a combination of techniques including a peripheral-foveal vision system, an attention system combining bottom-up visual saliency with structure from stereo, and a localization and mapping technique. The result was an object recognition system that can be trained to locate the objects of interest in an environment, and subsequently build a spatial-semantic map of the region.

In the same line of work, Ranganathan and Dellaert (2007) suggest the 3D extension of the constellation object model to represent places using objects and develop learning and inference algorithms for the construction of these models. Others, as Jebari et al. (2011a) build a semantic map based on the object categories table, and when one object is detected the system verifies if it is in the map. If it is the first time that the object is detected, the object position Kalman filter is initialized; otherwise there is an update of the filter. In Rasolzadeh et al. (2009), the semantic mapping classifies the object as a thing or as an object. A thing is an unclassified object. Things are stored in the map, but when a classification arises where a thing fits, this thing is classified as an object. Although these works have an important contribution in semantically mapping objects, they do not explain how the robot can infer the human word that tags the place and there is not a clear formal hierarchical representation.

In contrast to all these works, others have tried to formalize a hierarchical representation from lower layers (metric) to upper layers (semantic) (Galindo et al. (2005);Diard and Bessi (2008); Vasudevan and Siegwart (2008) and Zender et al. (2007)). For example, Galindo et al. (2005) describes a multi-hierarchical representation that includes object and semantic labeling of places in a metric map but assumes the identities of objects to be known. They have drawn the robot internal representation of its environment from two different perspectives: (i) a spatial perspective, which enables it to reliably plan and execute its tasks (e.g., navigation); and (ii) a semantic perspective, which provides it with a human-like interface and inference capabilities on symbolic data (e.g., a bedroom is a room that contains a bed). Figure 3.3, depicts their approach. It includes two hierarchical structures, the spatial and the conceptual hierarchies. The Spatial Hierarchy arranges its information in different levels of detail: (i) simple sensorial data like camera images or local

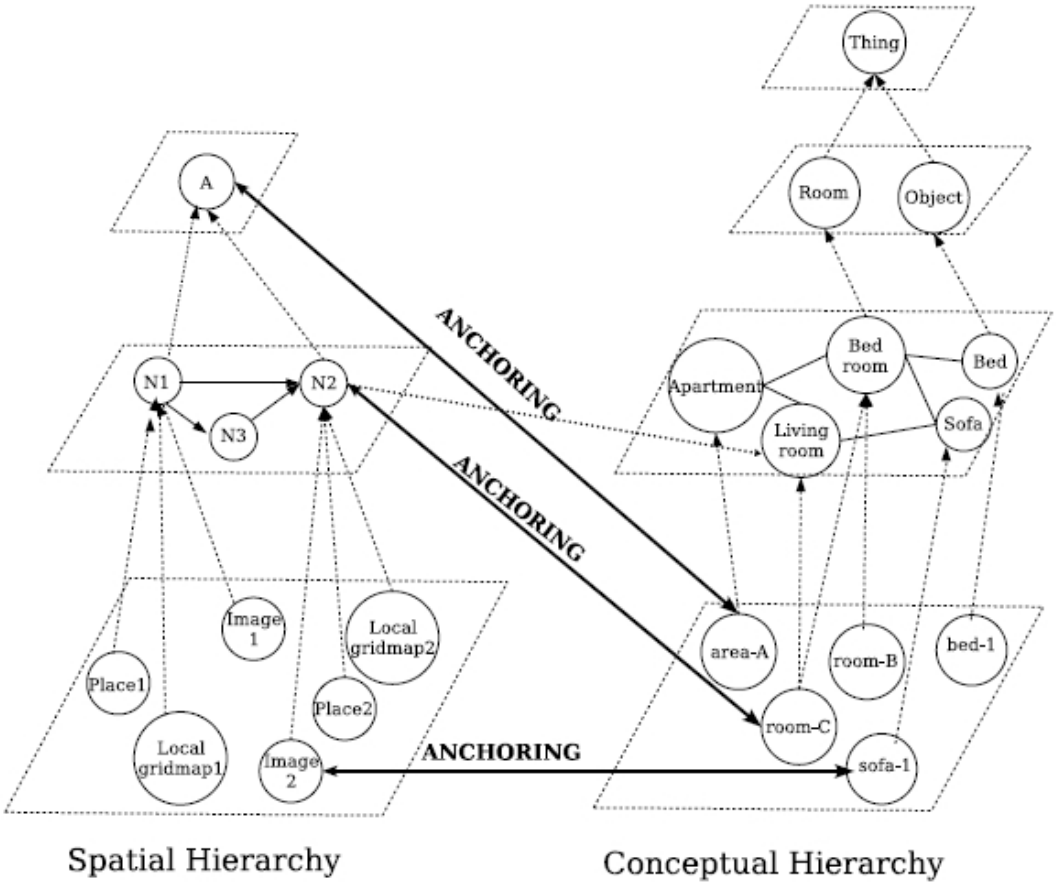

Figure 3.3: The spatial and semantic information hierarchies, by Galindo et al. (2005). On the left, spatial information gathered by the robot sensors. On the right, semantic information that models concepts in the domain and relations between them. Anchoring is used to establish the basic links between the two hierarchies (solid lines). Additional links can then be inferred by symbolic reasoning (dotted line).

gridmaps, (ii) the topology of the robot environment, and (iii) the whole environment represented by an abstract node. The Conceptual Hierarchy represents concepts (categories and instances) and their relations, modeling the knowledge about the robot environment. They claim that this simplifies the development of approaches for the robot to make inferences about symbols, which are instances of given categories. From here, Galindo et al. (2008) show two ways how semantic maps can improve task planning: extending the capabilities of the planner by reasoning about semantic information, and improving the planning efficiency in a domestic environment.

In the same line of work, a more complete and explored approach was suggested by Zender et al. (2007). The understanding of the spatial environment was explored as a whole, considering world perception, natural language, learning and reasoning. This knowledge was structured in

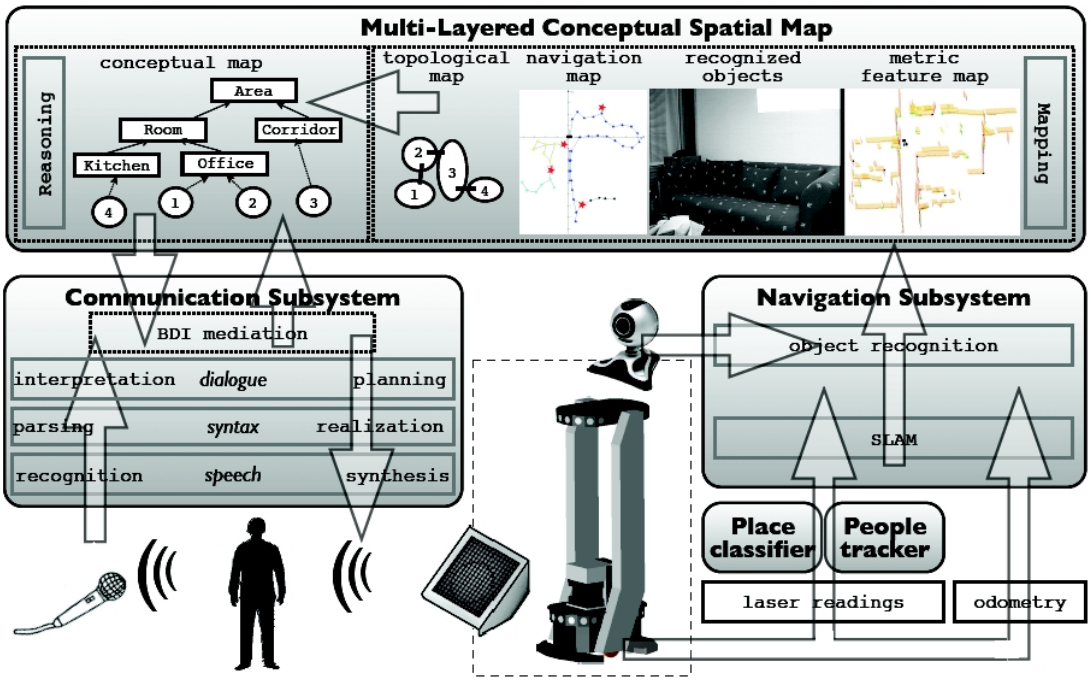

Figure 3.4: Multi-layered representation defined by Zender et al. (2007)

multiple levels of abstraction: metric map (obtained with conventional approaches, as SLAM), navigation map, topological map and concept map, as depicted in figure 3.4.

This framework, proposed by Zender et al. (2007), was explored by Jensfelt et al. (2007), Zender et al. (2008), Kruijff et al. (2007) and Pronobis (2011) mainly to perform spatial reasoning and to infer semantic room categories. For example, Pronobis and Jensfelt (2012) explores the perception layer and the problem of place classification. Figure 3.5 depicts how the sensory layer is connected to the conceptual layer. Place classification is characterized as a problem of pattern recognition and association of a region of the environment to one predefined class.

The types of property assigned to places are:

- objects - each object class results in one property associated with a place encoding the expected/observed number of such objects at certain places
- doorway - determines if a place is located in a doorway
- shape - geometrical shape of a place extracted from laser data (e.g. elongated, square)
- size - size of a place extracted from laser data (e.g. large (compared to other typical rooms))
- appearance (e.g. office-like appearance) - visual appearance of a place

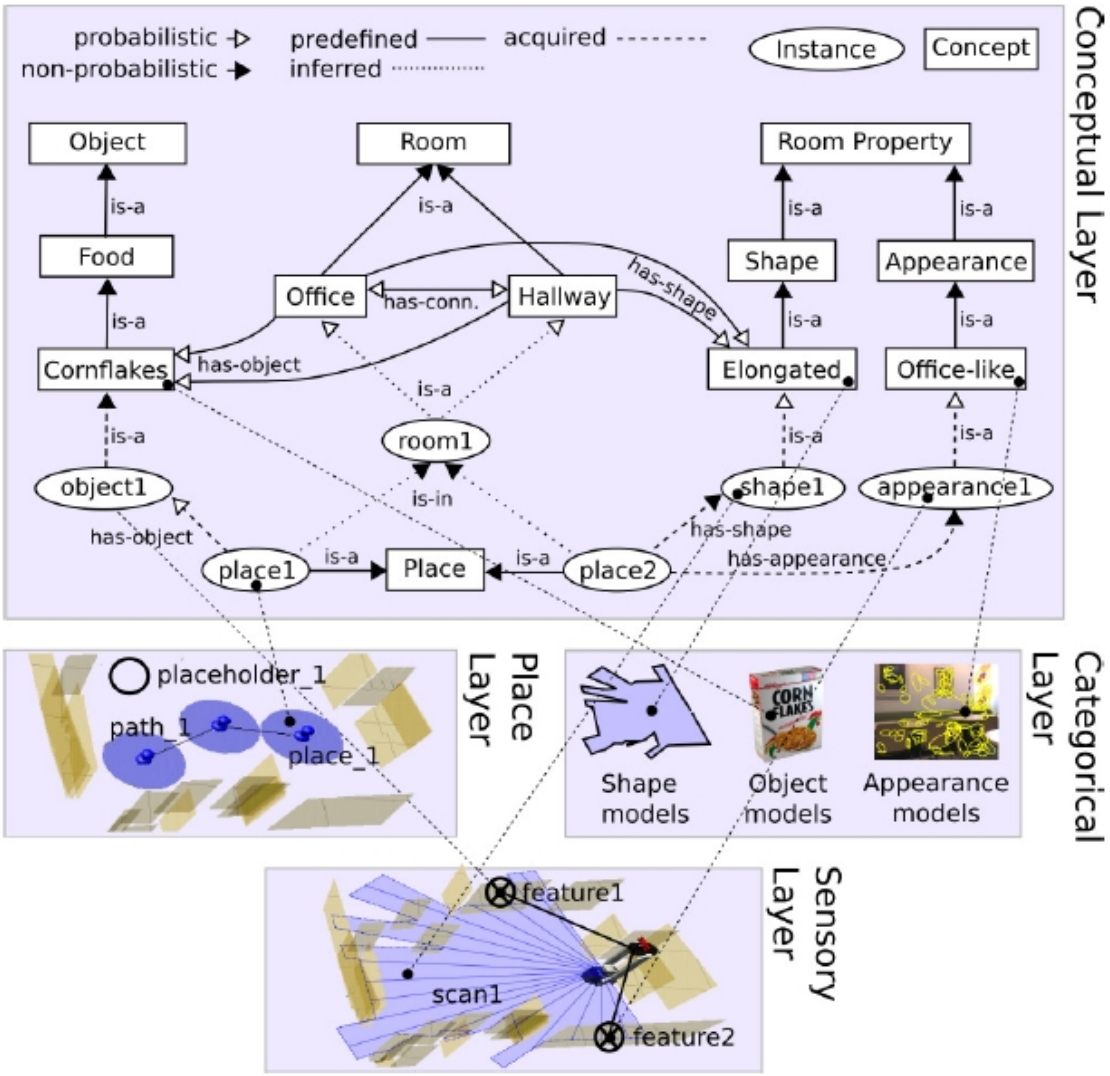

Figure 3.5: The layered structure of the spatial representation and the visualization of an excerpt of the ontology of the conceptual layer. The conceptual layer comprises knowledge about concepts (rectangles), relations between those concepts and instances of spatial entities (ellipses). By Pronobis (2011).

The problem of place classification is divided into place recognition and place categorization. "Support vector machine (SVM)", "Speeded Up Robust Features" (SURF) and "Composed Receptive Fields Histograms" (CRFH) are the main techniques used to solve problem of place classification, as depicted in figure 3.6. After evaluating their work with data from COLD-Stockholm database, in offline mode, they claim a recognition rate above 80%.

This approach, suggested by Zender et al. (2007) and worked by other authors, abstracts metric maps to conceptual maps and simplifies the human-robot interaction. However, they do not consider a SLAM approach with a three dimensional map, nor other features in their place definition, such as walls and windows, nor a place delimited by imaginary limits, or a specific layer to

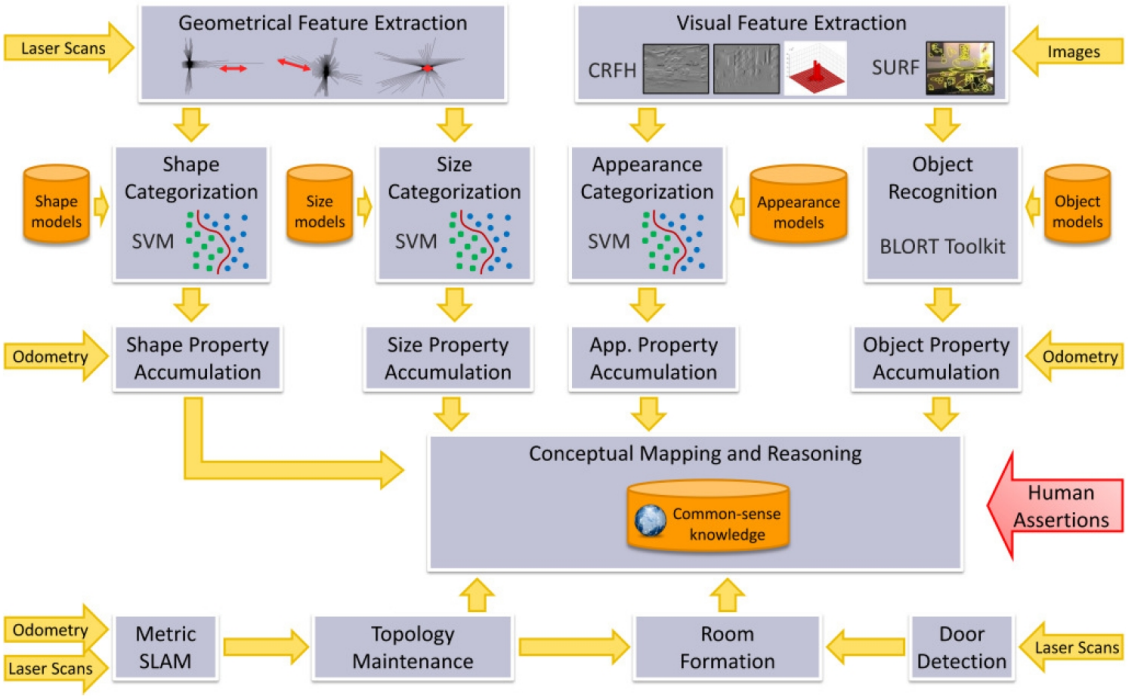

Figure 3.6: Pronobis and Jensfelt (2012) depicts the structure of the system and data flow between its main components.

deal with object classification and mapping.

3.2 HySeLAM - Hybrid Semantic Localization and Mapping extension

To answer the fundamental question on *How can a SLAM approach be extended in order to include the human into the mapping process*, and inspired by other works (Galindo et al. (2005);Diard and Bessi (2008); Vasudevan and Siegwart (2008); Zender et al. (2007); and Pronobis and Jensfelt (2012)) a new hierarchical map representation, called HySeLAM, is proposed. HySeLAM stands for Hybrid Semantic Localization and Mapping (figure 3.7) and it is a semantic mapping framework extension for the classical SLAM process.

In contrast to previous works, the HySeLAM framework was designed to be compatible to any SLAM approach, which works with a gridmap (2D or 3D). This simplifies the use of this framework over other systems which already have its own SLAM approach, RoboVigil by Pinto et al. (2013a). The process of defining, classifying and mapping is different for places and objects. The dynamic of objects is higher than that of places and the physical definition of place is more abstract than the object's. So, instead of using a conceptual map near to SLAM, as Zender et al. (2007), in the HySeLAM the mapping process is splitt in three layers: the metric layer, which

is managed by a SLAM approach, the topological layer where place definition and spatial symbol grounding happens, and objects mapping layer, where objects are mapped with a metric and semantic relation.

Pronobis (2011) classifies a place by observing the place geometry. However, high level features, as walls and windows, are not extracted to enrich the knowledge about the place. The use of these features would make this knowledge reusable by other robots that do not have the same sensor configuration. Also, there are places which are not delimited by physical barriers, such as the resting place of the robot, while other places are delimited by physical barriers but invisible to the robot sensors (for example, glass is invisible to some laser ranger finders and cameras). This requires the place definition to consider visible and invisible limits. In HySeLAM there is the definition of virtual and real walls, which will help to solve this problem.

When the robot asks the human to describe the place, it will get something like this: *Robot, you are on the corridor. On your left you have the bathroom, and then room1, room2 and room3. On your right you have room4 and room5. Room room3 is also known as John's office. Room2 is 6 meters by 8 meters. Room4 is 6 meters by 6 meters. Room5 is 6 meters by 4 meters. In room room3 there is a table with a PC Monitor on top.*. This description helps to find important tags that should be considered in our semantic mapped process, as for example (at, inside, in, on, left, right, top). In the HySeLAM framework these tags are used to relate spatially objects and places, and they can inherit metric values making the knowledge more precise.

A more detailed look into the HySeLAM framework shows us that it is divided in two layers: topological and objects mapping. In the topological layer, the topological map is the first grid map abstraction and it is managed by the topological engine. This topological map stores the place's features and connectivity. Each place is defined by a set of human words, a set of virtual or real walls and a set of visual signatures. Connectivity between places is given by virtual or real doors. Virtual walls and doors are used to defined a place not delimited by real walls. Place features, such as walls and visual perspectives, are related to the SLAM navigation referential.

In the objects mapping layer, the semantic map (or object map) stores the spatial relation between objects and the spatial relation between objects in the map of the place. The spatial relation is defined by a set of words $\{at, inside, in, on, left, right, top\}$. Each object stored in the object map inherits object features and properties from a generic object definition stored in the object dictionary.

In the objects mapping layer, the object dictionary stores the learned generic object definitions. This dictionary is managed by the OST module. OST stands for Object Segmentation and Tracking. Besides dictionary management, the OST module tracks and detects the objects present in the environment. The OST inputs are the object dictionary and visual images.

3.2.1 Augmented topological map

The augmented topological map stores the knowledge about places definition and connectivity, in the form of an attributed graph. Also, this map stores the relation between places and the gridmap. Places are defined by their delimitations (real or virtual walls), locations, visual signatures and by

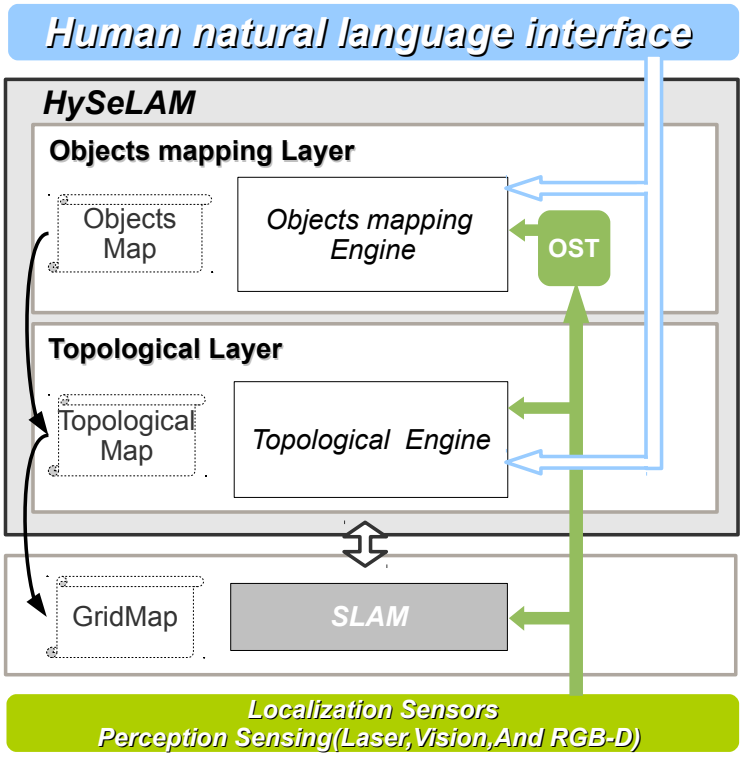

Figure 3.7: HySeLAM - Hybrid Semantic Localization and Mapping. HySeLAM is a semantic extension to classical SLAM, divided in two layers: topological and objects. The topological layer stores the place properties and place connectivity in the topological map. The objects mapping layer stores the relation between objects and between objects and places in the object map. OST - Object segmentation and tracking runs a process for object segmentation and tracking. OST manages an Object dictionary.

human words. The edges store the connectivity between places and are labeled with virtual or real doors. These doors are defined by their size, location and human words. This map is managed by the topological engine, which will be able to buildup the topological map using the gridmap, as described in chapter 4, and merge this map with descriptions received from other entities (other robots/people), as described in chapter 5.

Therefore, the topological map $\mathcal{M}_t$ is defined by an attributed graph:

$$\mathcal{M}_t = (\mathcal{P}, \mathcal{C}) \tag{3.1}$$

where: $\mathcal{P}$ is the set of vertices (places) and $\mathcal{C}$ the edges ($\mathcal{C} \subseteq \mathcal{P} \times \mathcal{P}$). The places are augmented with five attributes: semantic words, geometric description, visual signatures, area and central position. A place is defined as:

$$p_i = \{\mathcal{SP}, \mathcal{W}, \mathcal{V}, A_r, X_c\} \tag{3.2}$$

Where: $\mathcal{SP}$ is a semantic set of words labeling the place, $\mathcal{W}$ defines the real and/or virtual

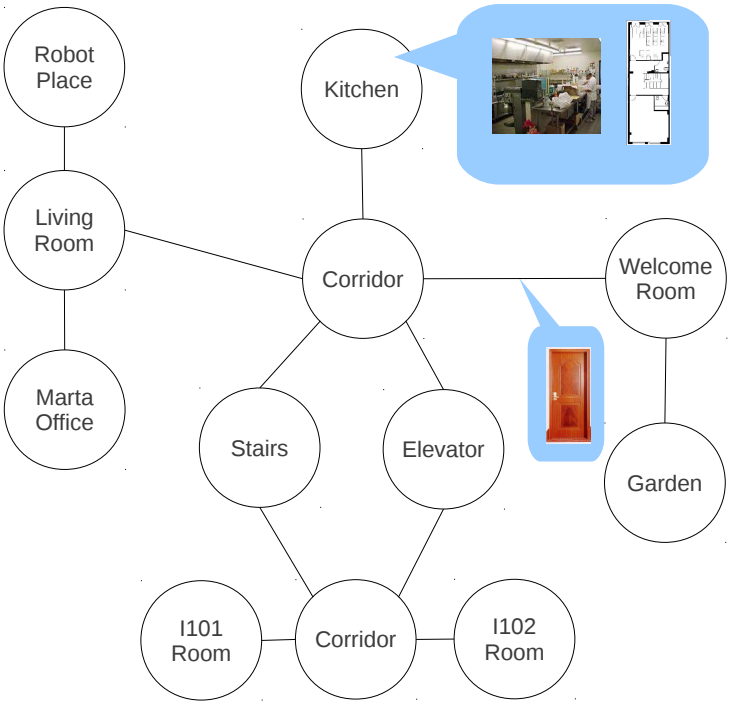

Figure 3.8: The Augmented Topological map defines a place according to its delimitation and visual gist. The edges/arcs define the connection between places and are labeled by doorways.

delimitation of the place with a set of wall's ($\mathcal{W} = \{w_0, w_1, ..., w_{nw}\}$); $\mathcal{V}$ is the visual signature described by a set of local visual signatures ($\mathcal{V} = \{v_0, v_1, ..., v_{nv}\}$); A_r is a real number which defines the place area; and X_c is the centric position ($X = [x, y, z]$) of the place.

The parameter wall w_i is defined by:

$$w_i = (X, \overrightarrow{V}, L, S, e) \tag{3.3}$$

Where: $X = ([x, y, z], \Sigma_X)$ defines the position of the mass center of the wall and the uncertainty associated to this observed position; this position is related to the origin of SLAM referential frame; $\overrightarrow{V} = ([vx, vy, vz], \Sigma_V)$ contains the normal vector which defines the wall direction and the uncertainty associated to that vector; $L = ([vl, vh], \Sigma_V)$ defines the length (height and width) of the wall and the uncertainty associated to the length; $S = [sl, sh]$ defines the shape curvature of the wall; e defines the existence of the plan, (0 when the wall does not exist, 0.5 a glass wall and 1 when the wall is a brick wall).

The visual signature parameter v_i is defined by:

$$v_i = (\mathcal{I}, X, La, t) \tag{3.4}$$

Where:$\mathcal{I}$ is a matrix containing the image/signature, $X = ([x,y,z],\Sigma_X)$ is the center of image/signature acquisition and the associated uncertainty described by a covariance matrix (Σ_X); La is the angular aperture ($L = [Lx,Ly]$); and t is time of acquisition.

The edges ($\mathcal{C} \subseteq \mathcal{P} \times \mathcal{P}$) are labeled with two attributes: semantic words and doorway set, as follows:

$$\{\mathcal{SD},\mathcal{D}\} \tag{3.5}$$

Where: $\mathcal{SD}$ is a semantic set of words labeling the edge, and $\mathcal{D}$ is the doorway definition.

The parameter door way $\mathcal{D}$ is defined by:

$$\mathcal{D} = (X,Ld,e,o,I) \tag{3.6}$$

Where: $X = ([x,y,z],\Sigma_X)$ defines the position of the mass center of the door and the uncertainty associated to this observed position, this position is related to the origin of SLAM referential frame (Σ_X); Ld defines the length of the plan ($L = [vl,vh]$); e defines the existence of the door (0 when the door is virtual, and 1 when the door is real); o is the last door state observed (open or closed); I stores a visual appearance of the door.

3.2.2 Topological engine description

The topological engine has seven components (*TopoParser*, *TopoMerge*, *PlaceMan*, *TopoVisualSignatures*, *TopoState*, *Topofeatures* and *Gr2To*) and one topological map (figure 3.9). This topological map is updated by these seven components using data provided by the human interaction, SLAM and robot vision. At the first stage, this topological engine will wait until the SLAM approach completes the mapping process of a predefined area or until a human describes the place. If there is an acceptable amount of area mapped into the gridmap, the *Gr2To* component is called and the first version of the augmented topological map is constructed. The details of *Gr2To* component are shown in chapter 4.

At this stage, the topological map does not contain any human word or visual signature associated to each segmented place. With this map, the *PlaceMan* component will notify the human-interaction layer to ask a human for the name of the place where the robot is, or ask to describe all the place with places names and connectivity. If the human tells only the name of that place, the *TopoParser* component will send that name to the *PlaceMan* component, which will associate the human word to the segmented place where the robot is placed. However, if the humans describes all the places, the human interaction should send this information to the *TopoParser* in the following format:

Robot, you are on the corridor. On your left you have the bathroom, and then room1, room2 and room3. You are on the corridor. On your right you have room4 and room5. Room room3 is also known as John's office. Room2 is 6 meters by 8 meters. Room4 is 6 meters by 6 meters. Room5 is 6 meters by 4 meters.

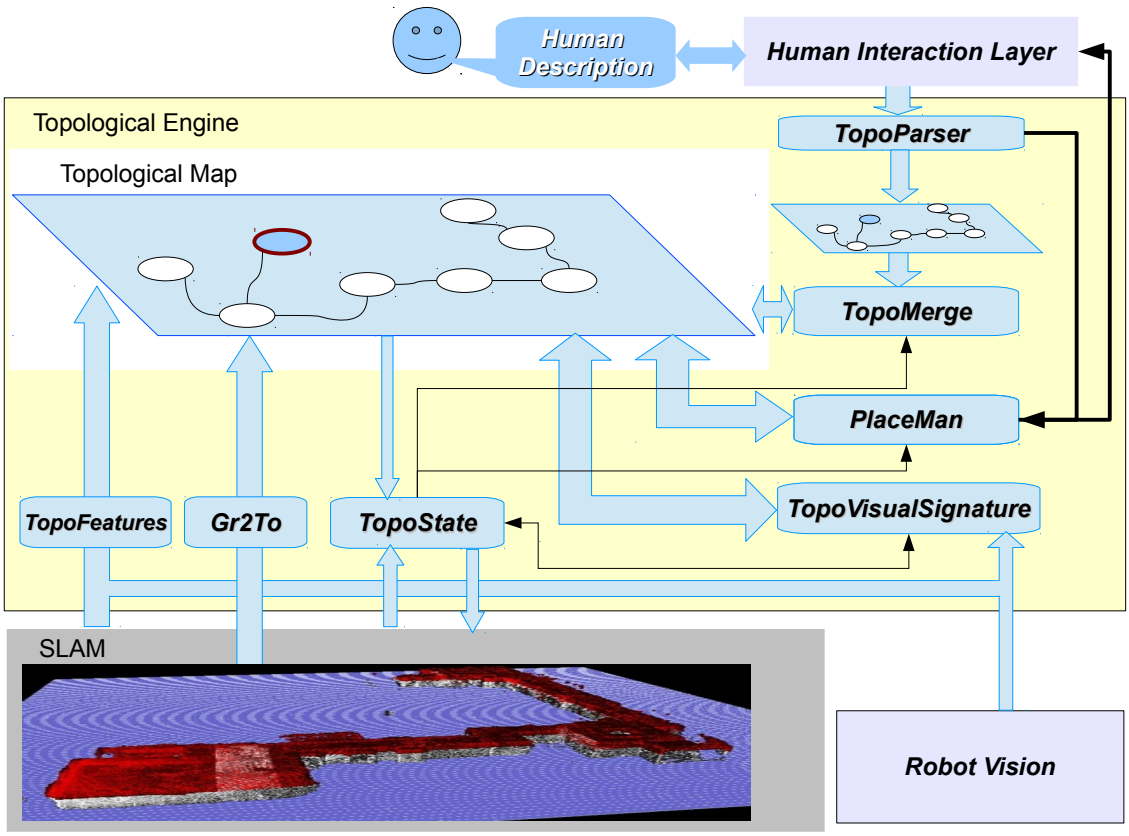

Figure 3.9: The topological engine has seven components: *TopoParser*, *TopoMerge*, *PlaceMan*, *TopoVisualSignatures*, *TopoState*, *Topofeatures* and *Gr2To*.

After, the *TopoParser* will convert this text to an augmented topological map. At this moment, there are two complementary topological maps, one obtained from SLAM and other from human-robot interaction. The first is more accurate from a geometric perspective and it is related to the gridmap. The second map contains the human place segmentation labeled with human words without any connectivity to the gridmap obtained by the robot. In this case, the *TopoMerge* is called and it will try to merge these two maps. The details of the *TopoMerge* component are shown in chapter 4.

The function of *PlaceMan* is to detect segmented places without semantic tags and notify the human-interaction layer to ask for human words for these places. Also, it should associate human words to segmented places, and it can be used to add virtual places to the topological map or virtual features as walls or doors to places. For example, when a human says " *Robot, you are in room3 also known as John office.*", this component will get the state of the robot in the topological map from *TopoState*, and it will get the semantic tags (e.g. room3 and John office) from *TopoParser*. With robot state, the *PlaceMan* component will add these semantic tags to $\mathcal{SP}$, which is an attribute of the place. In contrast, if the human says "*Robot, you are in a virtual place, with 1 meter of radius.*" the *PlaceMan* will add a new place (vertex) to the topological map, with virtual walls. After that, *PlaceMan* will ask for the name of this place. If the human says, "*Robot,*

you have a glass wall in your front.", the component will try to add a virtual wall and infer the size from the gridmap.

The *TopoParser* component receives structured sentences and converts them to a topological map, to be used by *TopoMerge*, or splits the sentences into parameters to be used by *PlaceMan* component. The *TopoState* reads the robot position from SLAM and updates the state of the robot in the topological map. It can work in reverse way, as when SLAM gets a higher uncertain in the position estimation or an invalid position, the *TopoState* can use the *TopoVisualSignatures* to guess where the robot is located. The *Topofeatures* uses the robot vision and SLAM observations to detect features of places, as doors and windows.

The function of the *TopoVisualSignatures* component is to classify and recognize places from an image acquired from the robot vision system and from visual signatures, which are stored in the topological map. During the robot life, this component will acquire visual perspectives and store them into the topological map. The main purpose of these stored images is to get an image database of the places. This database and topological map are going to be used by the classifier to get unique visual signatures for each segmented place. These unique visual signatures are going to be used to recognize places and detect the robot state in the topological engine. The process of place recognition can help to increase the recovery speed of SLAM (in a kidnapped or lost situation) and detect malfunctions in SLAM. For the scenario (place) to be recognized quickly, the perspective of Oliva and Torralba, presented in Oliva and Torralba (2006) can be used. They argue that fast scene recognition does not need to be built on top of object recognition but can be prompted by scene-centered mechanisms. They defend that position by pointing out behaviors on human vision: when provided with a glance of a shot a person can identify the meaning of that given shot or *gist of a scene* without remembering specific details. Other works (by Murphy et al. (2006), Linde and Lindeberg (2004a), Pronobis et al. (2009), and Ranganathan (2010)) can be adopted for place recognition.

3.2.3 Objects mapping layer

The objects mapping layer stores the robot knowledge in the object dictionary and in the objects map, figure 3.10. The object dictionary stores the generic definition for each object. The objects map stores the knowledge about objects detected in the external world and their spatial relation, in the form of attributed graph. The spatial relation is defined by a set of words $\{at, inside, in, on, left, right, top\}$. Each object stored in the objects map inherits object features from a generic object definition stored in the object dictionary and the object properties are filled based on robot sensor observations or on human description.

The objects map $\mathcal{M}_S$ is defined by an attributed graph :

$$\mathcal{M}_S = (\mathcal{O}, \mathcal{R}) \tag{3.7}$$

where: $\mathcal{O}$ is the object set ($\mathcal{O} = \{o_0, o_1, ..., o_{on}\}$) and $\mathcal{R}$ the edges ($\mathcal{R} \subseteq \mathcal{O} \times \mathcal{O}$).

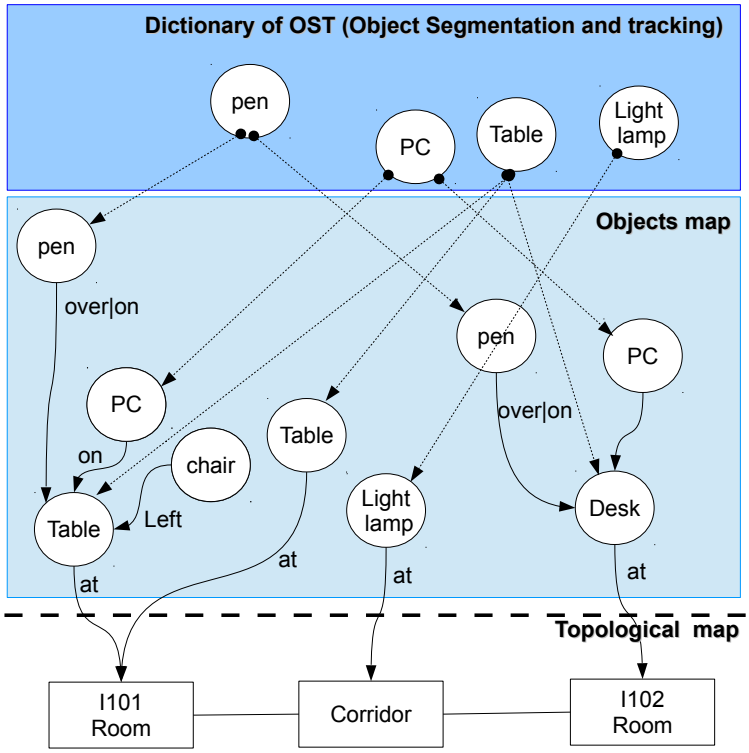

Figure 3.10: The objects map relating Object/place spatially. Each object is an instance of a generic object described in Object dictionary. The places are represented by rectangular symbols and objects are represented by circular symbols.

The edges are tagged with the relation between objects and also to the place. The relation is described by a triplet:

$$\mathcal{R}_i j = \{\mathcal{WR}, \mathcal{ER}, \mathcal{MR}\} \tag{3.8}$$

where: $\mathcal{WR}$ is the object semantic physical relation, given by a word of relation dictionary $\mathcal{RD} = \{on, over, in, inside, at\}$. When the object is related to a place it is always related with the semantic word *at*. $\mathcal{ER}$ is the euclidean vector relating the center of object-object/place. $\mathcal{MR}$ is a real number, $[0, 1]$, and defines if the object is movable (0 for fixed object and 1 for movable).

The object o is an instance of generic object $\mathcal{GO}_k$ with a specific property $\mathcal{OF}$. The $\mathcal{GO}_k$ is an element of the object dictionary $\mathcal{OD}_{go} = \{\mathcal{GO}_1, \mathcal{GO}_2, ..., \mathcal{GO}_n\}$. The generic object $\mathcal{GO}_k$ is defined by a quadruple:

$$\mathcal{GO}_k = \{\mathcal{OS}, \mathcal{OT}, \mathcal{OF}, \mathcal{OC}, \mathcal{OW}\} \tag{3.9}$$

where: $\mathcal{OS}$ defines the 3D object shape, $\mathcal{OT}$ defines the 3D object texture, $\mathcal{OF}$ defines the parameterizable features, $\mathcal{OC}$ defines the classifier and features for object detection, and $\mathcal{OW}$ is a generic object name.

3.2.4 Objects mapping engine

The objects mapping engine has four components, one object map and one object dictionary, as depicted in figure 3.11. The objects maps is updated by *FillMap* and *ObjectDetector*, the object dictionary is only updated by the *NewObject* component.

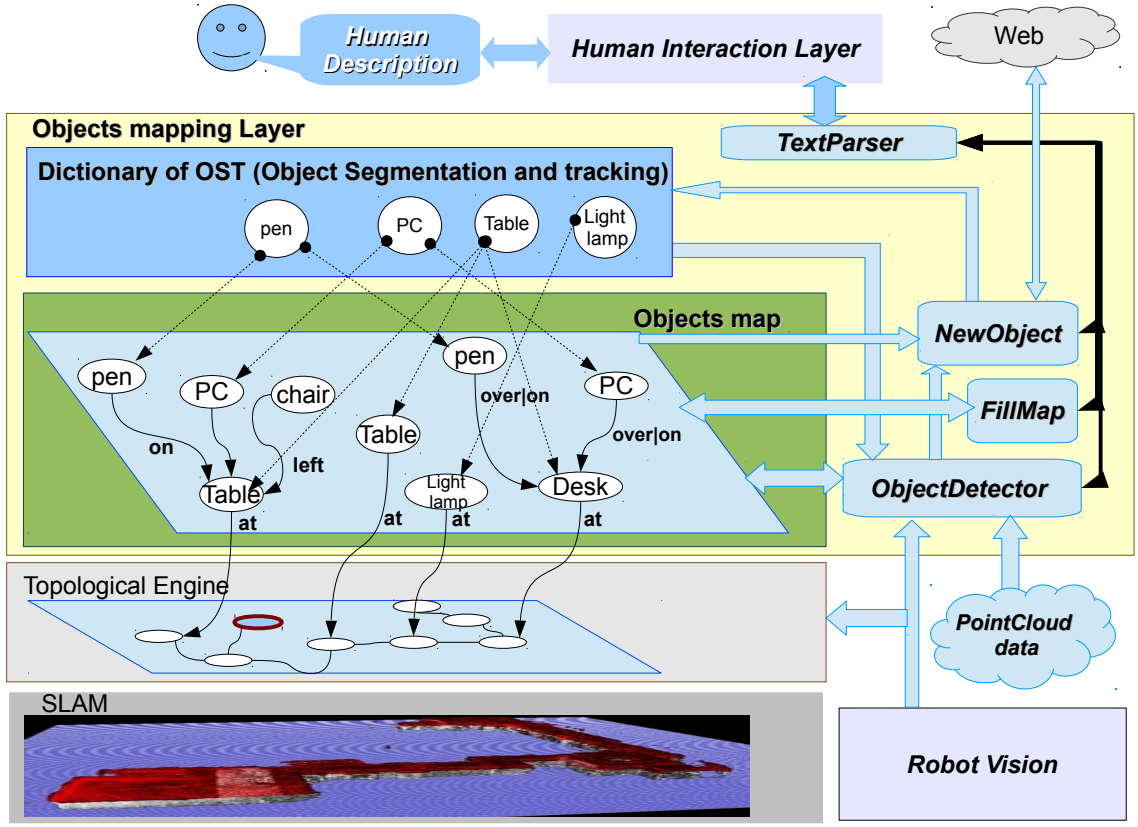

Figure 3.11: The objects mapping engine has four components: *TextParser*, *NewObject*, *FillMap*, and *ObjectDetector*

The update of the objects maps can happen when a human describes objects available in one place, as in this way: *Robot, in John's office there is a PC and one pen over the table. On the left of the table there is a chair.*. At this moment the *TextParser* will parse this sentence for the *FillMap*. The *FillMap* will see if any of these objects are stored in the objects maps related to that place. If they are not found, the component will fill the objects maps with this knowledge, as depicted in figure 3.11. If the object dictionary does not have a definition for these objects, the robot will not be able to recognize them in the external world, because *ObjectDetector* does not know how to identify them from the robot sensors information. Here, the *NewObject* component will identify objects in the objects maps which are not defined in the dictionary and component will try to add the object definition to the dictionary. There are two ways: one way is to search in the web for a description of the object, and the other way is to learn the new object with the help of human-robot interaction.

The object definition is dependent on the approach used for object recognition and classification implemented in *ObjectDetector*. However, a full functional, efficient and universal approach to object recognition and classification does not exist. The object recognition is a complex problem since, it includes identifying instances of a particular object (for example: a Ferrari car) or scene as well as generalizing to categorize instances at a basic level (for example: any car). Object recognition and classification is still a research field with a large amount of issues to solve, as is the case of efficient algorithms with higher efficiency and able to work in real time. So, the *ObjectDetector* component should be based on multiple approaches and the selection of these approaches, in each moment, should be based on component mode (learning, find and tracking) and on object definition. These approaches can include object learning and recognition using RGB-D sensors (by Rusu (2009); Shanming and Malik (2013); and Bo et al. (2012)) and the visual learning and recognition of 3-D objects from their appearance (by Murase and Nayar (1995); Leonardis and Bischof (2000); and Grauman and Leibe (2011a)). Although this component is able to create its own object dictionary, it should be able to acquire and reuse the knowledge acquired from other robots and shared in the cloud (by Waibel et al. (2011); and Quintas et al. (2011)).

Chapter 4

Towards the Extraction of Topological Maps from 2D and 3D Occupancy Grids

The next generation of robots which will cooperate with humans should be able to describe the world with place names and place connectivity. Also, these robots should infer from their knowledge and observations the limits of these places. This is will be essential for the robot to be able to understand sentences as *"Robot, go to the Room C"* or *"Robot, come to the Room D, which is near to C"*, or also *"Robot, the name of this place is Room T"*.

Instead of conventional SLAM (Simultaneous Localization and Mapping) methods which do not interpret sensor information other than at the geometric level, these new robot capabilities require an environment map representation closer to the human representation. Topological maps are one option to translate these geometric maps into a more abstract representation of the the world and to make the robot knowledge closer to the human perception. For these reasons, the HySeLaM framework defines a new augmented topological map on top of the conventional SLAM approaches which are based on occupancy grid maps.

The purpose of this section is to show an approach that translates a grid map, which was obtained from SLAM, into an augmented topological map which was defined in section 3.2.1. This approach is novel and it is capable of translating a 3D grid map into the augmented topological map. It was optimized to obtain similar results to those obtained when the task is performed by a human. Also, a novel feature of this approach is the augmentation of the topological map with features such as walls and doors.

Section 4.1 presents a global overview to the problem and different approaches to translate a grid map into a topological map. Section 4.2 presents the novel approach to translate a grid map into a topological map. Section 4.3 presents the obtained results using this novel approach. Section 4.4 presents the chapter conclusions and future directions.

4.1 Global overview

The well known Simultaneous Localization And Mapping (SLAM) problem, detailed in section 2.4, for mobile robots was widely explored by the robotics community in the last two decades. Several approaches were proposed and explored. Nowadays there are valuable solutions based on these approaches that make it possible the robots to operate in crowed places and in buildings without special requirements. Rhino by Burgard et al. (1999), Robox by Siegwart et al. (2003), Minerva by Thrun et al. (2000) and RoboVigil by Pinto et al. (2013a) are some of those robots that can create a map and use it to localize and navigate through the environment. These robots rely on 2D or 3D accurate metric representations of the environment, derived from SLAM techniques.

Using these SLAM approaches, the robot can build an accurate 2D or 3D grid map of the environment autonomously. However, considering that robots are moving from high tech factories to our homes, offices, public spaces and small factories, some of them will be required to work and cooperate alongside us. In this scenario, the information about cell occupation in the grid map is not enough to provide an association of a human word to a place name or even an object. Moreover, this knowledge, in the form of a grid map, is not easily communicable. The HySeLAM extension is one way to abstract this knowledge and make it more communicable. It creates a new layer over the grid map, the topological layer. Nevertheless, two questions emerge *How a human does the place segmentation from the occupancy gridmap* and *how this can be done by the robot and stored in to the augmented topological map*, as depicted in figure 4.1.

Firstly, an occupation grid map is a metric map (Elfes (1989),Chatila and Laumond) that discretizes of the environment into 2D or 3D cells. The grid map consists of empty cells, $m(x,y) = 0$, which represent free space and occupied cells, $m(x,y) = 1$, where an obstacle exists. 2D occupation gridmaps are the most commonly used in mobile robots, but 3D occupation gridmaps are growing in popularity due to the low cost of 3D acquisition systems, such as low cost RGB-D cameras and 3D laser range finder scan solutions. These 3D grid maps are extremely expensive in terms of memory size, so they are often stored in the form of octomaps.

Usually, a topological map (earlier used in robotics by Mataric (1990) and Kuipers and Byun (1991)) describes the world using a graph, which is composed by a set of vertices and edges. In a topological map, the set of vertices is related to the significant spaces/places in the environment, and the edges supply information about the connectivity between vertices. In the HySeLAM framework, the edges represent a real or virtual door and these vertices represent a room or a distinctive place and are augmented with real and virtual delimitations. For these reasons, the topological map is more compact and widely used for global path planning.

Thrun (1998), among other researchers (as Joo et al. (2010); Fabrizi and Saffiotti; and Myung et al. (2009)), has extracted topological models from grid maps to solve global path planning and to help performing navigation and localization in local areas.

One way to get a topological map is to extract the Voronoi diagram from the grid map, as depicted in figure 4.2. A Voronoi diagram is a way of dividing space into a number of regions which are delimited by the Voronoi segments. These are all the points in the plane that are equidistant to

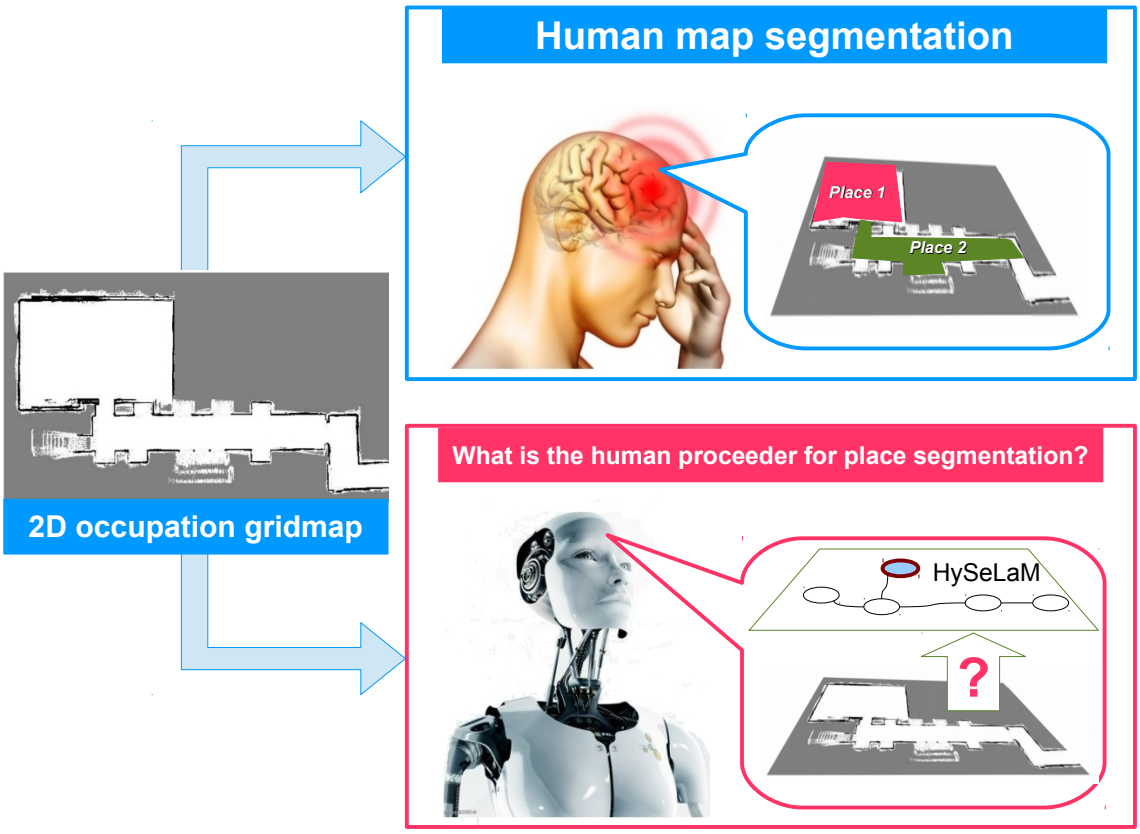

Figure 4.1: How do humans do place segmentation from a gridmap?

the two nearest sites. The Voronoi vertices (vertices) are the points equidistant to three (or more) sites. In Lau et al. (2010) we found an approach to extract the Voronoi diagram from a dynamic grid map; this approach was been optimized for online Voronoi diagram extraction.

Thrun (1998) extracts the Voronoi diagram and critical points from the grid map. Critical points are used to delimit regions and the Voronoi edges are used to relate region connectivity. These regions and edges are used to build the topological map. However, these topological maps extracted for global path planning are oversegmented and they do not extract the boundaries of each vertex.

In contrast to these works, Brunskill et al. (2007) and Zivkovic et al. (2006) use graph partitioning methods to divide a grid map into several vertices. Buschka and Saffiotti (2002) suggest an approach where room-like spaces are extracted from grid maps using fuzzy morphological opening and watershed segmentation. Even though those methods show successful topology extraction from grid maps, they are not easy to apply directly in home environments because they are suitable for corridor environments or considers only narrow passages to extract a topological model.

Choi et al. (2011) suggest an approach in which the topological modeling is based on low-cost sonar sensors. The proposed method constructs a topological model using a sonar grid map by extracting subregions incrementally. A confidence for each occupied grid is evaluated to obtain

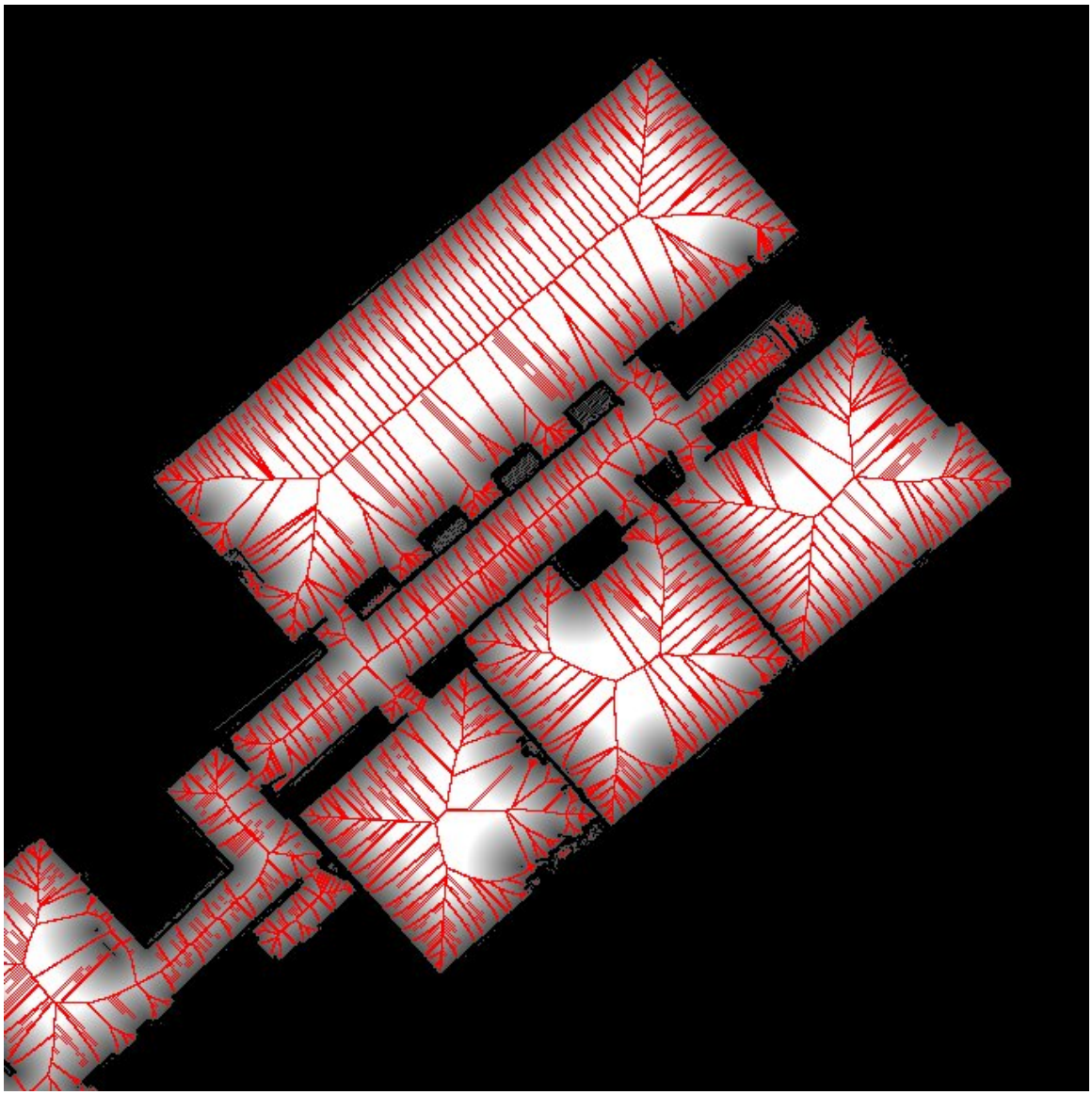

Figure 4.2: The Vonoroi diagram (red) for this gridmap was obtained using the Dynamic Voronoi algorithm described in Lau et al. (2010). The grid map was obtained with RobVigil and Hector SLAM in the first floor of the building I of the Faculty of Engineering of the University of Porto.

reliable regions in a local grid map, and a convexity measure is used to extract subregions automatically. In contrast to this work, Jae Gon and Heung Seok proposes an approach based on the following rule: finding the largest free rectangular area recursively in a given area of the occupancy gridmap. The output result is a topological map called R-Map, which represents the environment with a set of rectangles. This method has slight problems in nonrectangular divisions.

In another direction, Choi et al. (2009) and Joo et al. (2010) suggest another approach based on the central concept of a virtual door, depicted in figure 4.3. A virtual door is defined as the candidate of a real door. In the approach suggested by Joo et al. (2010), virtual doors are detected as edges of the topological map by extracting corner features from the occupancy grid-map; using

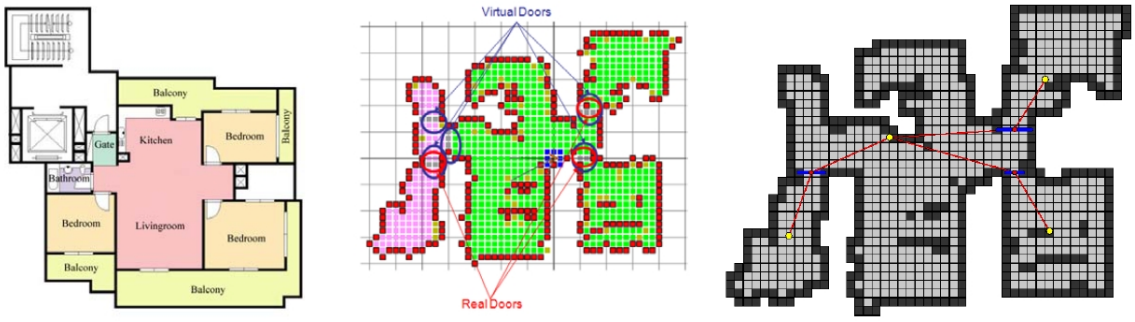

Figure 4.3: Two approaches for door detection. From left to right: Real environment, Occupancy gridmap with virtual doors detected by the approach of Choi et al. (2009), and Occupancy gridmap with Virtual doors detected by the approach of Joo et al. (2010)

this method, an initial topological map is generated, which consists of vertices and edges. The final topological map is generated using a genetic algorithm to merge the vertices and reduce the edges. As a result, the generated topological map consists of vertices divided by virtual doors and edges located in real doors. The proposed methods provide a topological map for user interaction and for the cleaning task planning, and the topological map can be used to plan more efficient motion including room-to-room path planning and covering each room. Although this approach shows a good performance for gripmaps with standard door size, it fails with non standard.

In the problem of place segmentation from an occupancy gridmap, the concept of virtual doors, proposed by Myung et al. (2009) and by Joo et al. (2010), has proved to be useful for correct segmentation. However, it considers that all places are connected by doors, which is not always true and leads to the detection of fewer places than exist in reality. Other approaches based on Vonoroi maps, proposed by Thrun (1998), have proved to be useful to extract the topological map from the gridmap for global path planning. However, the use of interception points and critical points (narrow passages) obtained from the Vonoroi maps, normally results in a higher number of detected places when compared to reality. The noise (furniture and other kinds of objects) present in the occupancy gridmap increases the difficulty of the place segmentation problem, and none of these approaches have tried to solve this problem.

4.2 Gr2To - 3D grid map to Topological map conversion

In the HySeLaM framework, the topological layer is created to store the place segmentation and the symbol grounding knowledge. However, a process is required to infer the augmented topological map from the occupation gridmap, which was obtained using a SLAM approach. The SLAM algorithm will build and maintain the grip-map of the environment. When this map is considered completed by the robot, the topological HySeLaM engine will start the extraction of the topological Map M_t from the occupation gridmap.

From the literature review, there are two approaches that should be considered for this topological map extraction. For this task, Thrun (1998) suggests the use of Voronoi maps and Joo et al.

(2010) suggests the use of the virtual door concept. Both approaches show a good performance in their domain of application, although the final result has higher or lower number of detected places when this number is compared to the one obtained by the task done by a human, figure 4.1. Both approaches are complementary and they should be considered.

The noise (furniture and other kinds of objects) present in the occupancy gridmap increases the difficulty of the place segmentation problem. However, none of those approaches have tried to solve this problem. When the input is a 3D occupancy gridmap, it is possible to compress this 3D occupancy gridmap into a 2D occupancy gridmap and probably remove the noise; this concept is explored in the following approach.

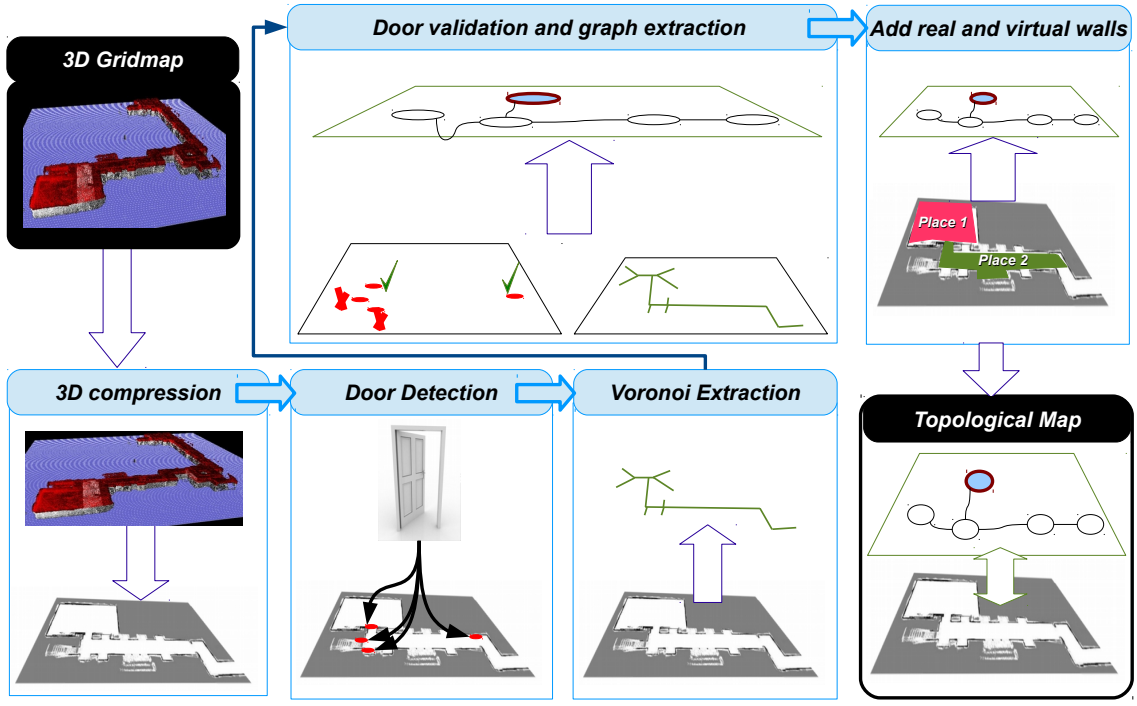

Figure 4.4: The Gr2To algorithm extracts the augmented topological map from an 3D occupation gridmap using five main stages.

Gr2To stands for Gridmap to Topological and is the name of the proposed approach. There were three main challenges to this approach:

- removing the noise (furniture and other kinds of objects) from the occupancy gridmap, when a 3D occupancy gridmap is available;
- detecting locations in the occupancy gridmap with higher probability of door existence;
- estimating the place segmentation and delimitation based on Voronoi maps, where the number of detected places and doors should be closer to the number detected by a human.

The Gr2To Algorithm defines our approach to convert the grid map into an augmented topological map, depicted in figure 4.4. Gr2To is divided into five stages: Compression from 3D grid

map to 2Dgrid map and Filtering; Door detection; Voronoi Graph Diagram extraction; Topological map construction and door validation; and, Optimization space delimitation and topological map augmentation with real and virtual walls. The source code of this algorithm can be found at http://www.fbnsantos.com/hyselam .

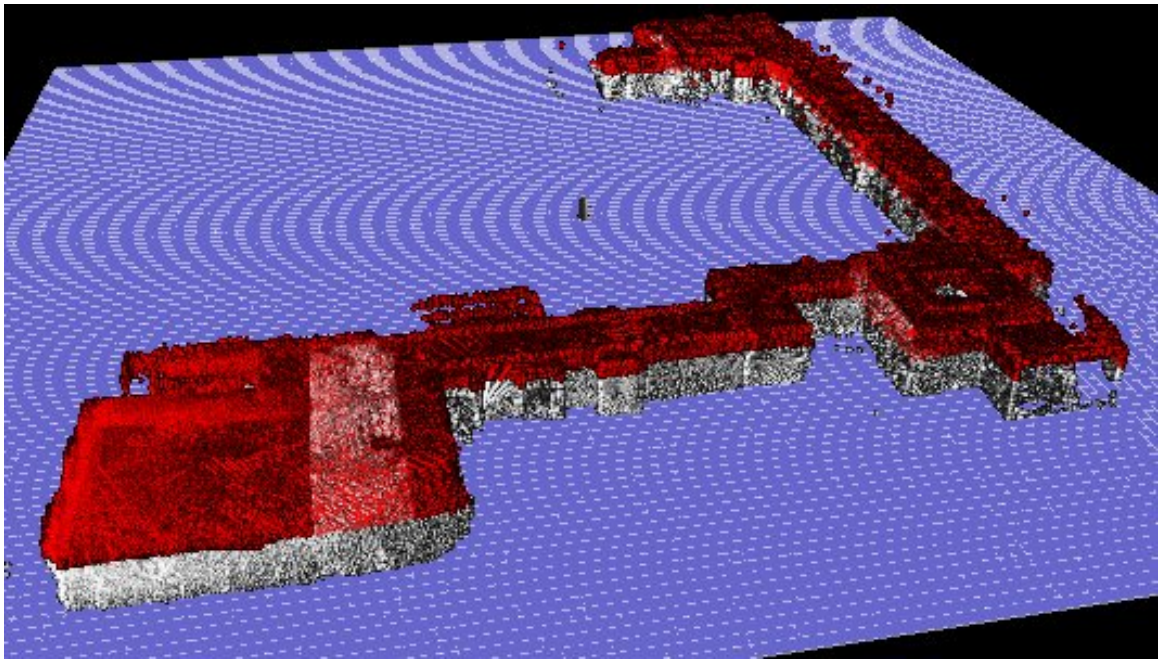

Figure 4.5: The 3D grid map obtained by the RobVigil Pinto et al. (2013a). The red points are occupied cells that are higher than 1.80 meters. The white points are occupied cells that are bellow. This map was obtained in the first floor of the building I of Engineering Faculty From Porto University.

First stage: Compression from 3D grid map to 2Dgrid map and Filtering. This stage have been designed to be compatible to 3D SLAM algorithms and to take some advantages of 3D maps. One of these advantages is the possibility of removing the noise present in the environment from the final map. Furniture or other kinds of objects present in the environment are considered noise to our algorithm because they do not define the boundaries of a room. When the input is a 3D map, figure 4.5, this map is compressed into a 2D map, figure 4.6. With this compression it is possible to remove the furniture/noise present in the environment from the final map, as shown in figure 4.6.

The compression algorithm compresses all cells in Z-axis into a single cell, as follows:

$$map_{2D}(x,y) = \begin{cases} 0, & \text{if } NO > NO_{min} \text{and } NF < NF_{max} \\ 1, & \text{if } NF > NF_{max} \\ 0.5, & \text{otherwise} \end{cases} \quad (4.1)$$

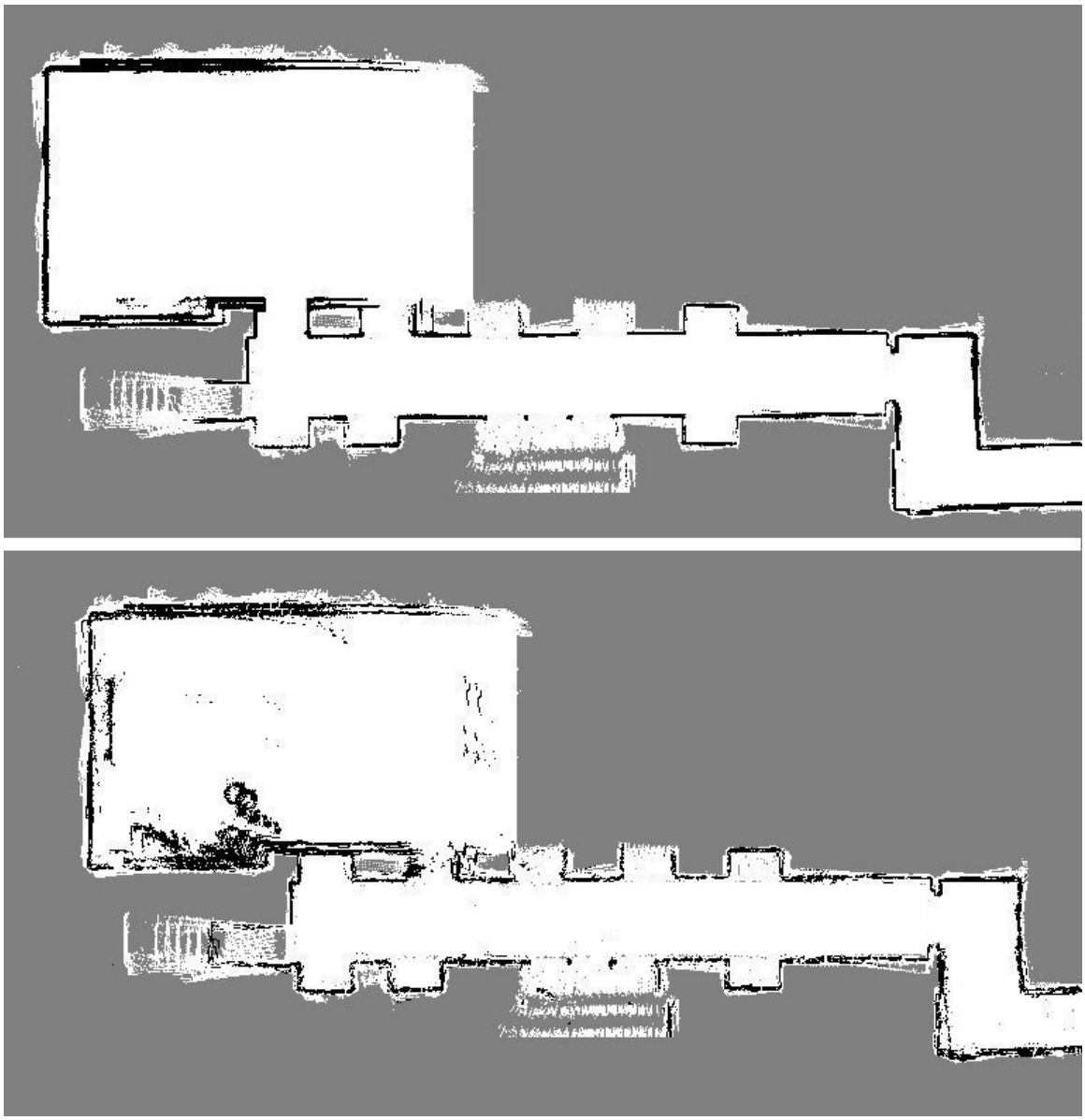

Figure 4.6: The top 2D grid map was obtained in the first stage of Gr2To using the 3D map, figure 4.5. The bottom 2D grid map is a map obtained using a 2D SLAM with the acquisition system at 1.20 meters from the floor. The top map has filtered the furniture present in the bottom grid map. (First floor, building I, FEUP)

where, NO is the function that returns the number of occupied cells from $map_{3D}(x,y,z_{min})$ to $map_{3D}(x,y,z_{max})$, NF is the function that returns the number of empty cells from $map_{3D}(x,y,z_{min})$ to $map_{3D}(x,y,z_{max})$. The compression algorithm requires six parameters, $(z_{min},z_{max},NF_{max},NO_{min},P_{Max},P_{Min})$. Where, z_{min} and z_{max} defines the interval of the 3D map to be compressed into a 2D map. NF_{max} and NO_{min} defines the maximum number of free cells and minimum number of occupied cells to consider a vertical occupied cell, P_{Max} defines the maximum probability for a cell to be considered empty cell and P_{Min} defines the minimum probability for a cell to be considered occupied.

After this compression, there is the map filtering step, where all occupied cells that do not have any neighboring occupied cells are removed and all empty cells that do not have any neighboring empty cells.

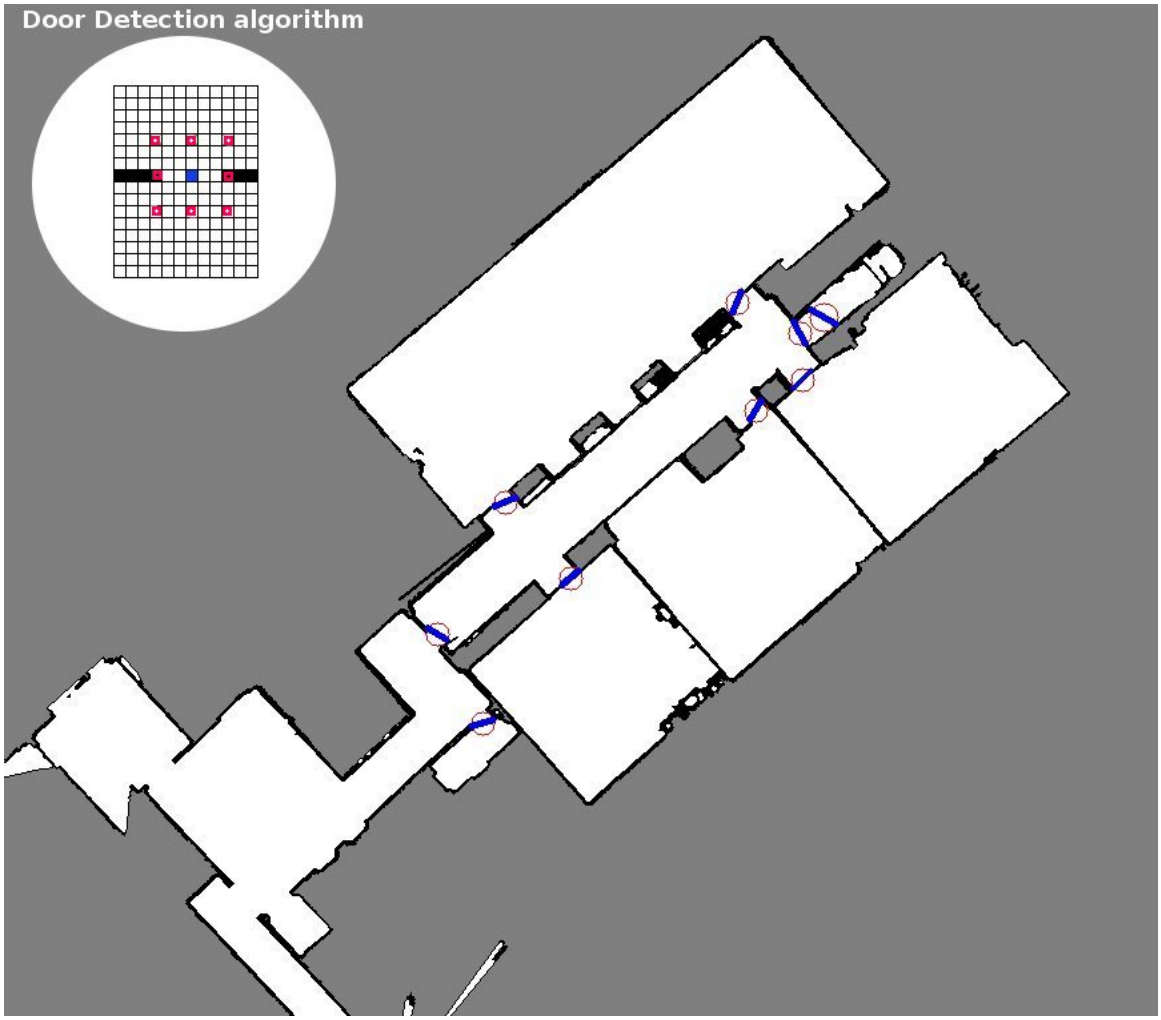

Figure 4.7: Red circles are places with higher probability of door existence and they were identified by the door detection algorithm. Blue lines represent the door in a closed state. On the left corner, the principle for door searching represented if the blue cell and the red cells satisfy the requirements then the blue cell is considered to have higher probability of door existence. (First floor, building I, FEUP)

Second stage: Door detection In this stage a map of distances is created, map_{dist} , from the filtered map. This map of distances contains in each cell the euclidean distance to the nearest occupied cell. Afterwards, the door detection algorithm search for locations with probability of door existence, figure 4.7. This search has two parameters $Door_{min}$ and $Door_{max}$, which defines the minimum and maximum door size, in meters. These values are converted to pixel units,

$Door_{minPixel} = \frac{Door_{min}}{2 \times Gridmap_{resolution}}$ and $Door_{maxPixel} = \frac{Door_{max}}{2 \times Gridmap_{resolution}}$. The search finds in the distance map for cells that satisfy the condition $Door_{minPixel} < map_{dist}(x,y) < Door_{maxPixel}$. For the cells that satisfy this condition, the algorithm takes eight samples from distance map, using this formula:

$$v(i) = map_{dis}\left(x + d \times cos\left(\frac{2\pi i}{8}\right), y + d \times sin\left(\frac{2\pi i}{8}\right)\right) \tag{4.2}$$

Where, d is the distance parameter, in our tests this parameter takes the value of $Door_{minPixel}$; i is an integer number between 0 and 7. If two samples satisfy the condition $v(i) > map_{dist}(x,y) \wedge v(j) > map_{dist}(x,y)$ with $2 <\mid i-j \mid< 6$, this place is considered to have higher probability of door existence.

The closer cells with higher probability of door existence are used to estimate a central location. This central location is stored in a vector of door locations. Closer cells are those with a distance under $Door_{minPixel}$ pixels.

Third stage: Voronoi Graph Diagram extraction In this stage the Voronoi Graph diagram is constructed from the distance map, figure 4.8. The development of this algorithm was based on previous works, as Lau et al. (2010).

Fourth stage: Topological map construction and door validation To construct the topological map we have considered the definition of critical point from previous works of Thrun Thrun (1998). In this stage, the critical point is used to validate the doors found in the second stage. After this validation, the algorithm travels by the cells that belong to the Voronoi diagram, 4.8. In each these cells, the algorithm gets three parameters to define a circle. The circle location $\overrightarrow{r}_c = (x_c, y_c)$ which is equivalent to cell location, and circle radius $r_c = map_{dist}(x,y)$. With this circle definition the algorithm searches all stored circles to know if the condition $r_c(i) + r_c < \sqrt{(x_c - x_c(i))^2 + (y_c - y_c(i))^2}$ is satisfied. If this condition is not satisfied the circle is stored; if the condition is satisfied, the algorithm selects the circle with bigger radius for the stored circles and drops the circle with smaller radius. Afterwards, the algorithms verify the next conditions for all circles:

$$\begin{cases} \sqrt{2}r_c(i) + \sqrt{2}r_c(j) < \sqrt{(x_c(j) - x_c(i))^2 + (y_c(j) - y_c(i))^2} \\ min(r_c(i), r_c(j)) > map_{dist}\left(\frac{x_c(j)-x_c(i)}{2}, \frac{y_c(j)-y_c(i)}{2}\right) \end{cases} \tag{4.3}$$

If these two conditions are satisfied, the algorithm drops the circle with smaller radius. After that, the algorithm finds the circle connections using the Voronoi Diagram Graph; the final result is in figure 4.10. With stored circles, circles connections and door places the algorithm builds up the topological map and stores it in an XML (eXtensible Markup Language) file.

Fifth stage: Optimization of space delimitation and topological map augmentation with real and virtual walls. In this stage the algorithm draws a closed door in all places that have a confirmed door. Then, one polygon of eight vertices is associated to each stored circle. The eight

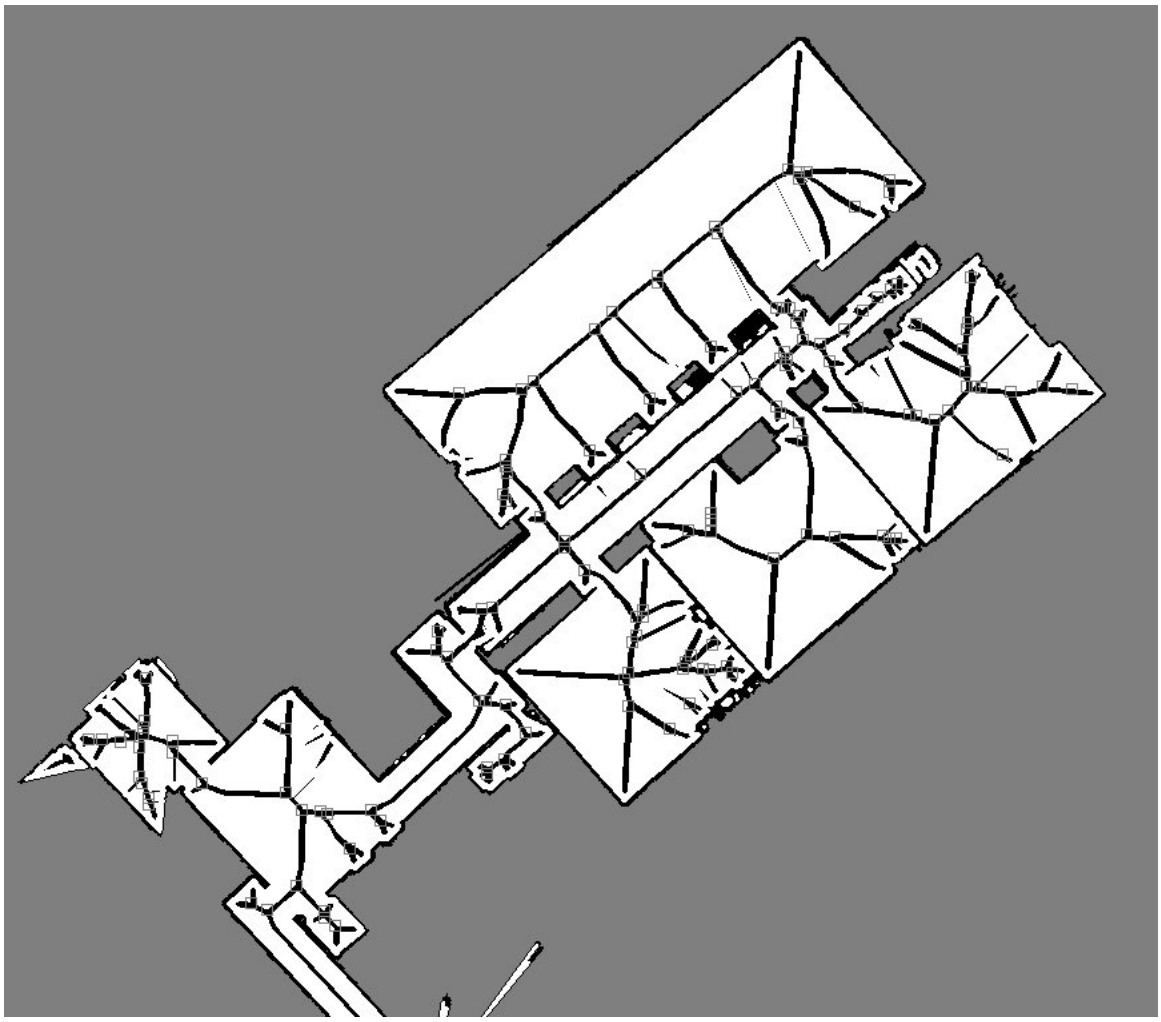

Figure 4.8: The obtained Voronoi graph diagram with nodes and edges numbered. The squares represents the nodes.(First floor, building I, FEUP)

vertices are placed over the circle edge equally spaced. Each vertex of the polygon is updated to the farthest occupied cell, left corner of figure 4.9. This update occurs inside one beam, the beam origin is the circle center and it have an aperture of $\frac{2\pi}{8}$ radians and is aligned to the vertex. This update is limited to a maximum distance from the center circle, defined by:

$$dist = \frac{r_c(i)}{r_c(i)+r_c(j)}\sqrt{(x_c(j)-x_c(i))^2+(y_c(j)-y_c(i))^2} \tag{4.4}$$

Where, i is the index of the actual circle and j is the index of the closest circle that is inside an actual beam. Then, the algorithm tests each polygon edge, and if there are over 80% occupied cells the place definition is updated in the topological map with a real wall defined by the polygon edge. If the condition is not verified the place definition is updated with a virtual wall.

Figure 4.9: This is the intermediate topological map obtained in the fourth stage. The red circles represent the main places (vertices), the cyan circles represent the location of the doors, and the black lines represent the connection between the main places. The yellow circles are the critical points. The green lines are a boundary defined by the critical points. (First floor, building I, FEUP)

4.3 Experimental results

In this approach Gr2To was tested using three grid-maps obtained in two real scenarios and in one virtual scenario. The output results were compared to a human segmentation. The three grid-maps, input of Gr2To, were printed and given to eleven persons. It was asked, to each person individually, to place a mark in each door, room and corridor of the map. The number of identified items and task completion time are shown on the table 4.1. Also, these three maps were processed by Gr2To in a computer with an Intel Pentium Dual Core T4300 processor at 2,16 GHz and with 2GB of memory.

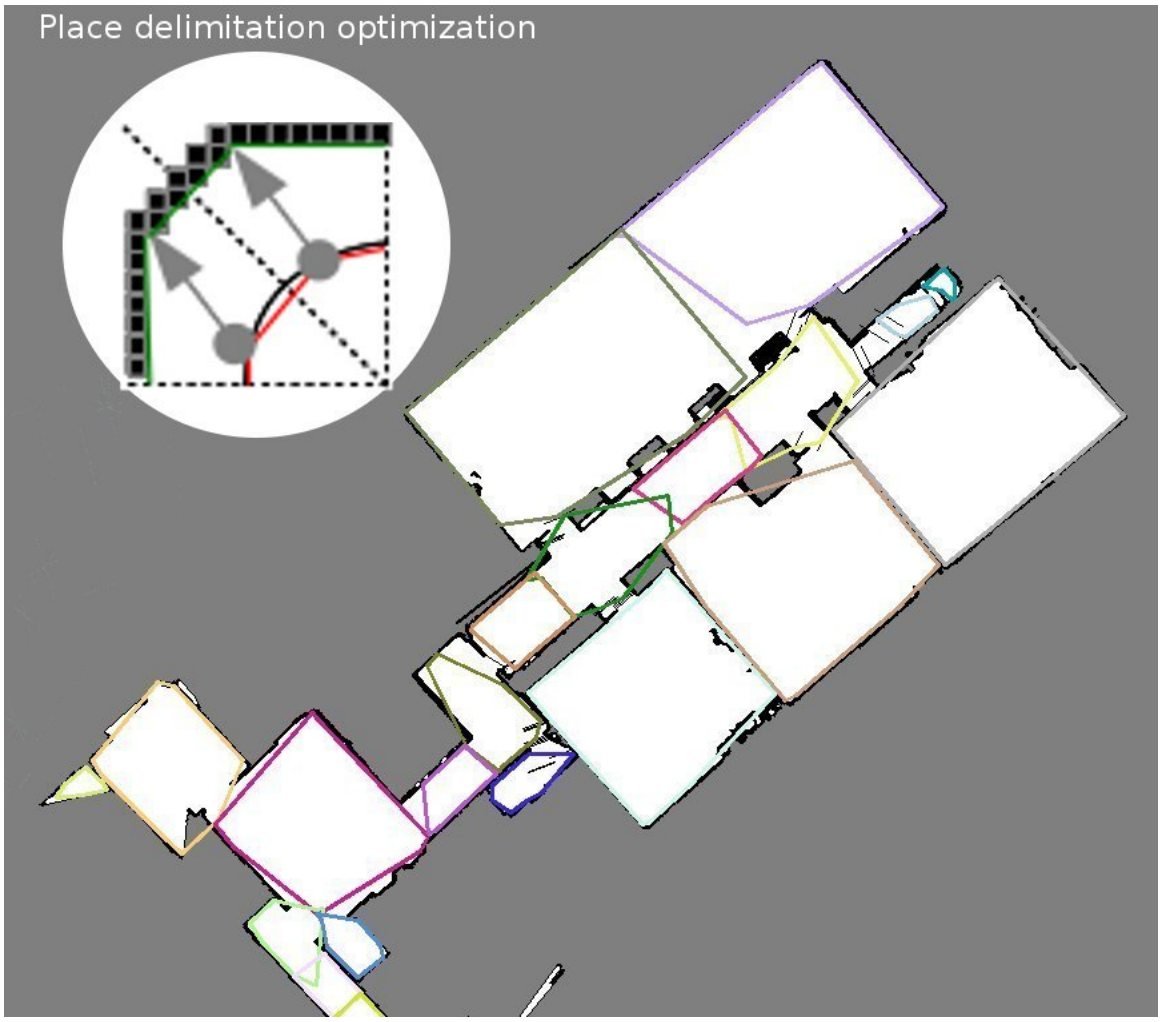

Figure 4.10: Segmented places over the grid map identified with random colors. (First floor, building I, FEUP)

The first grid-map was obtained from the robot RoboVigil, using a 3D SLAM approach proposed by Pinto et al. (2013a), in the ground floor of building I of the Faculty of Engineering of University of Porto (FEUP). In figure 4.7 and 4.10 intermediate steps and final place segmentation done by Gr2To can be seen.

The second grid-map was obtained in TRACLabs facility, which is available in Kortenkamp (2012). The third grid-map was obtained using the gazebo simulator with a turtlebot robot, Hector SLAM, and a virtual building. In figures 4.11 and 4.12 intermediate steps and final place segmentation done by Gr2To can be seen.

Table 4.1 summarizes the number of rooms, corridors and doors counted/identified by eleven persons and by Gr2To. It also shows the time taken to complete this task by humans and Gr2TO. Regarding human results, the three values in each cell are: the mean value of the samples rounded to the nearest integer, and the minimum and maximum values of the samples (inside of brackets). As for Gr2To results, in the corridor row there are two values: the number of corridors detected

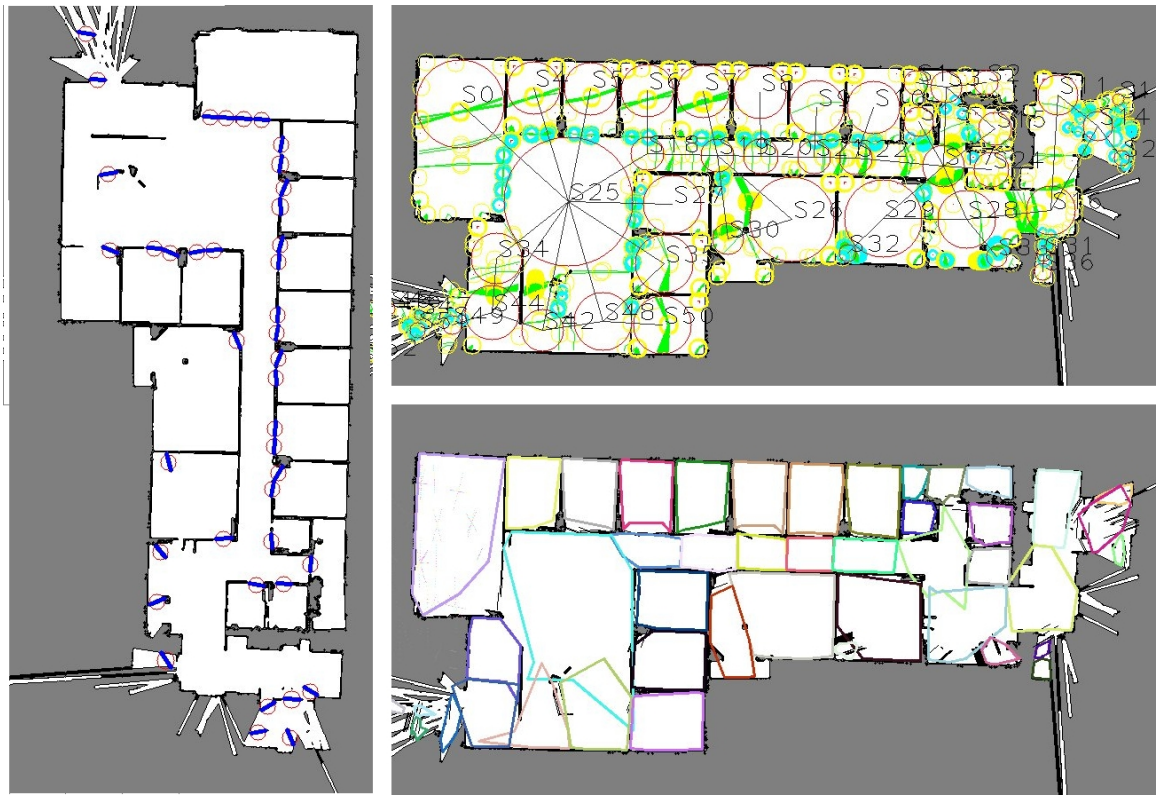

Figure 4.11: Result obtained with the gridmap of the TRACLabs facility, available in Kortenkamp (2012). In the left map, Gr2To has marked with red circles locations with higher probability of door existence. In the right top map, Gr2To draws red circles in the identified places, cyan circles in critical points with high probability for of door existence and black lines the connectivity between vertex. In the right bottom map, Gr2To draws places delimitation with random colors.

Table 4.1: Human segmentation Versus Gr2To approach segmentation

Grid Map	Segmentation by Human				Segmentation by Gr2To			
	Rooms	Corridors	Doors	t(s)	Rooms	Corridors	Doors	t(s)
FEUP	7 [5,8]	3 [2,4]	9 [5,10]	34 [26,47]	11	2 (7)	9 (2,2)	8.7
TRACLabs	17 [16,21]	2 [1,3]	18 [15,30]	43 [37,54]	29	1 (5)	39 (3,7)	8.4
Virtual	18 [17,21]	1 [1,2]	18 [18,20]	41 [33,53]	21	1 (9)	18 (0,0)	8.9

and the number of vertices merged (inside brackets). In the door column there are three values: the number of doors detected, the number of missed doors, and the number of wrong door detection.

From table 4.1 it is possible to see that in the virtual scenario the algorithm has detected the same number of doors, rooms and corridor as humans, while taking less time to process. In real scenarios, the Gr2To has failed to detect the doors with non-standard width. To make these doors detectable by Gr2To the value of the $Door_{max}$ parameter had to be increased. However, this has also increased the number of outliers in door detection. These outliers can be removed using visual door detectors. The number of doors detected by Gr2To in the TRACLabs is higher than

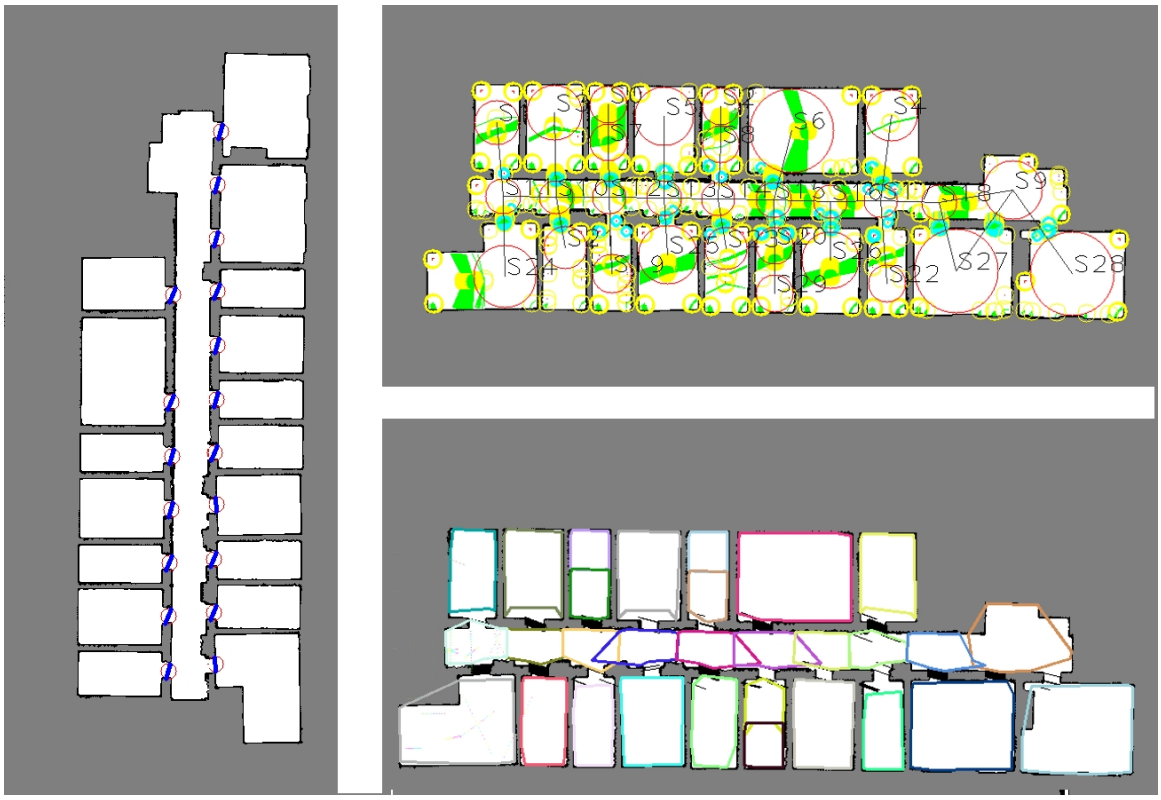

Figure 4.12: Intermediate steps and final place segmentation done by Gr2To with grid map obtained from virtual scenario.

the human average, which happens because humans have counted several pairs of doors as one door.

In the two real scenarios, the number of rooms detected by Gr2To is higher than the number of rooms detected by the average human. One approach to reduce the number of segmented places and approximate it to human input is to merge all vertices that exist between a door and a terminal vertex or door. However, from RoboVigil experience we found that over segmentation appears in long rectangular places, which are sometimes divided by humans in two distinctive places. So, if the Gr2To can fail to detect a door and these rectangular places are sometimes divided by humans, the merging of the vertices should happen only when the robot has more information about the place. The augmented topological map in the HySeLAM framework is a dynamic map and it is updated during the human robot interaction (voice), so the robot should only infer that a set of vertices corresponds to the same place when it gets the same human word for all vertices.

Gr2To was also tested with two more occupancy gridmaps, obtained in two distinctive real scenarios. These two occupancy gridmaps are made available by Roy and Nicholas (2003) in the robotics data set repository. These two scenarios are more complex, because they have more noise, are incomplete in some places and have corridors with curved sections. The results are shown in figures 4.13 and 4.14. Without knowing how these places are divided, in the first map, figure 4.13, it seems that Gr2To has performed an acceptable division of the place. However, in the second

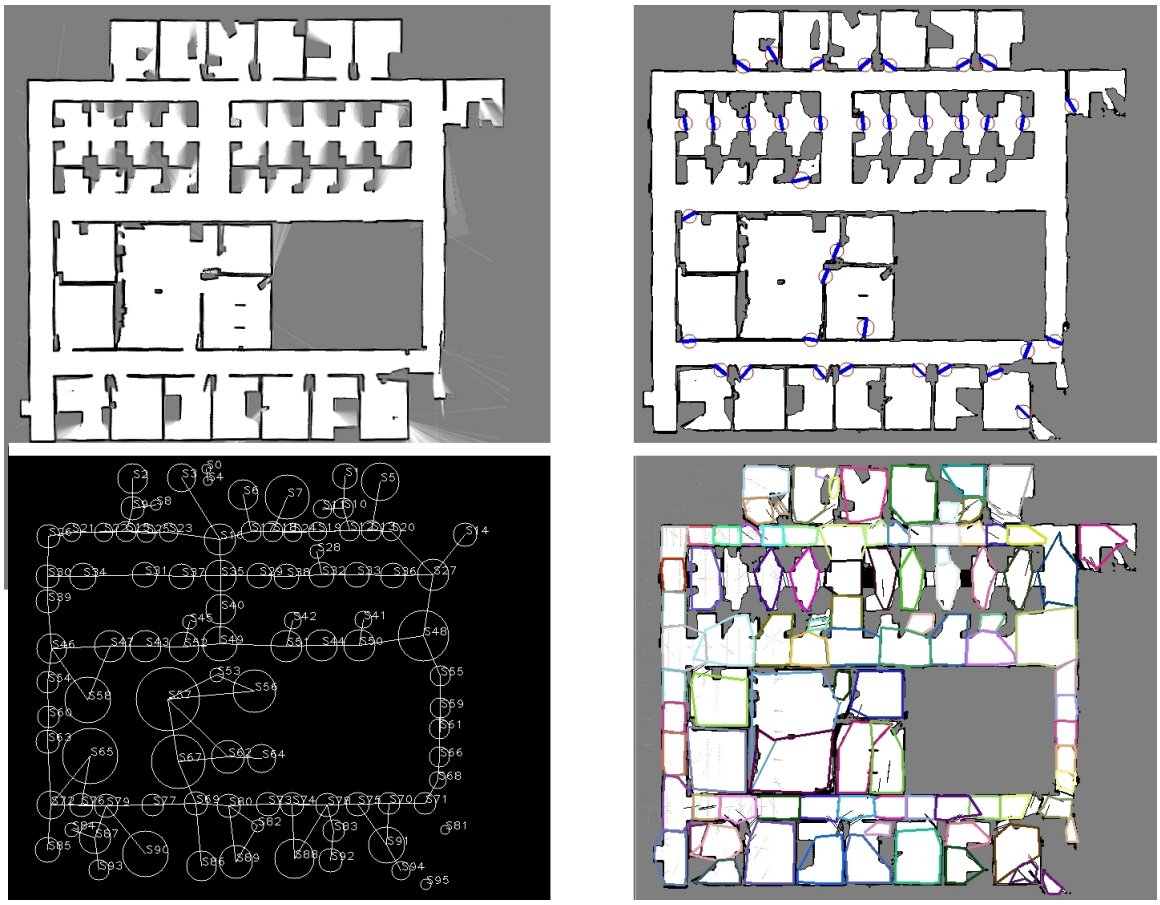

Figure 4.13: The intermediate steps and final place segmentation done by Gr2To with the occupancy grid map obtained by DP-SLAM, in SDR, on the site B of University of Southern California.

map, figure 4.14, using common sense it is possible to say that the final result has a higher number of segmented places. This happens because there is noise (furniture, objects and pillars) in the map in the form of occupied cells making it impossible for any algorithm to infer whether these occupied cells are relevant for the place segmentation without further information. Despite the higher number of segmented places, the algorithm shows good results delimiting the corridors with curved sections.

4.4 Conclusions and Future directions

The contribute made in this chapter was the development of a novel approach to translate a grid map into a topological map. This approach is able to compress a 3D grid map into a 2D grid map and filter noise (furniture and objects) present in the environment from the map, as shown in figure 4.6.

Another contribute of this work was the optimization of Gr2TO in order to obtain for the same task similar results to those obtained by a human, as shown in table 4.1. This was made possible

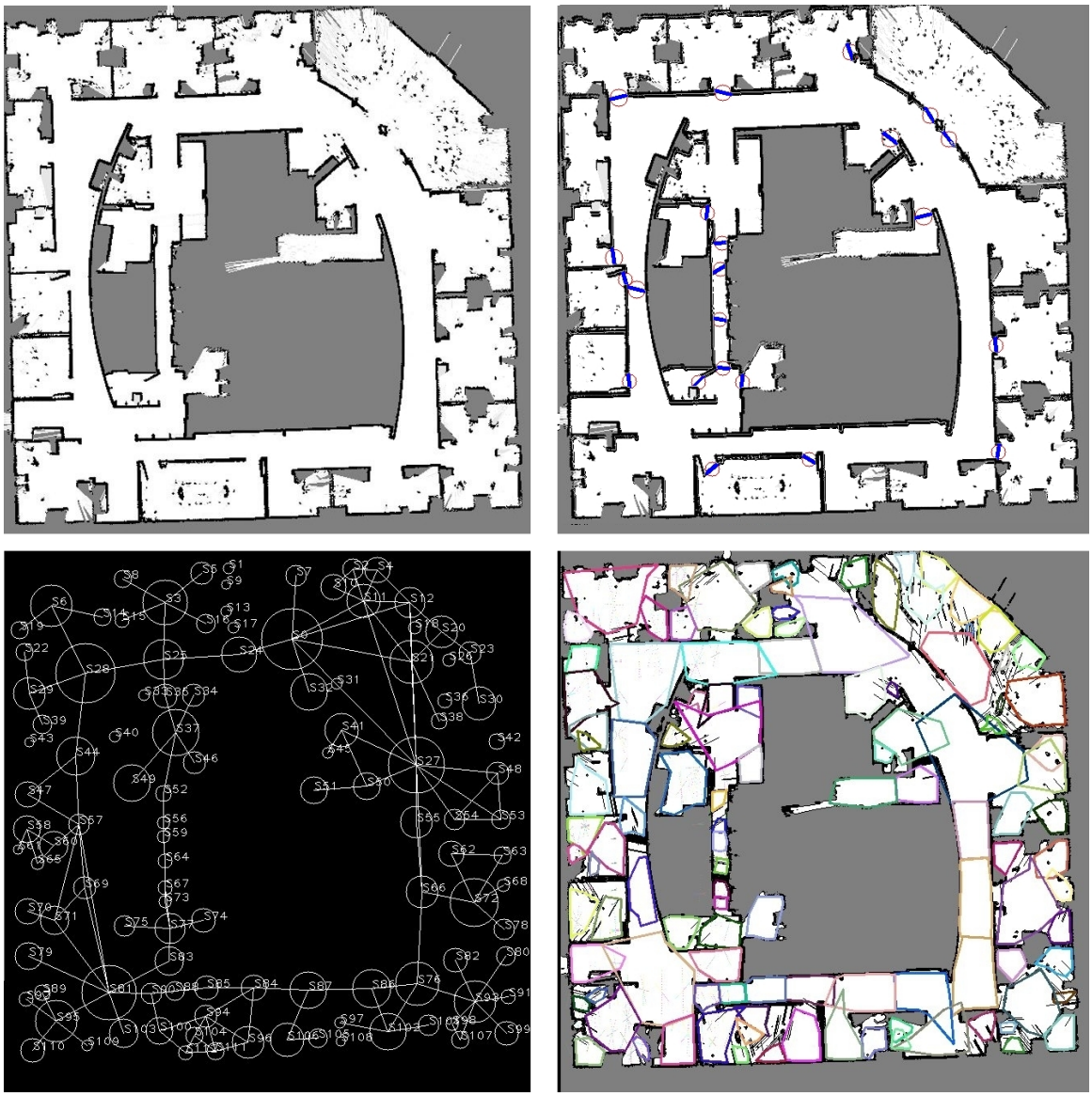

Figure 4.14: The intermediate steps and final place segmentation done by Gr2To with the occupancy grid map obtained by DP-SLAM inside the Intel Research Lab in Seattle.

using concepts proposed by Thrun (1998) and by Joo et al. (2010) – as virtual doors and Voronoi maps. Using both concepts it was possible to build an algorithm that segments the place with a number of places between the output number of the two approaches proposed by these authors.

Another important feature of this Gr2TO algorithm is the augmentation of the topological map with higher level features, as real and virtual doors and walls. This can have an important rule in point cloud data segmentation and can make it possible to share this topological map to other robots which do not know the environment map and do not have the same sensor configuration. In this scenario, the robot will achieve a higher description of the environment and this will help defining the best way to move through the it in order to obtain the occupation grid map in the

shorter time period

This Gr2TO algorithm shows that it is possible to translate the robot knowledge, stored in the form of an occupation grid map, and approximate the translation to the human perception. Indeed, RoboVigil with Gr2To was able to segment the obtained gridmap and ask in each place for the human word that tagged it. Gr2To simplifies the human-robot interaction and makes it possible for the robot to understand simple missions, as " Robot go to Sam's office and then go to robot charger place".

In spite of the good performance of the Gr2TO algorithm in all tested scenarios, there are components that should be integrated into the topological layer to improve the overall performance of the HySeLAM Topological Layer, especially when a 2D SLAM approach is used. The Gr2TO test which has used a 2D occupancy grid map obtained by DP-SLAM inside the Intel Research Lab in Seattle, figure 4.14, has led us conclude that other components in the topological layer are required in order to:

- Observe the external world and detect locations of higher level features, as doors and walls.
- Create an intermediate occupancy gridmap without noise (furniture and other objects).

A component to detect doors is the most important, because it can improve the Gr2TO algorithm results, it can help improving the performance of other algorithms of the robot and it can help the robot solving the complex task of detecting and opening a door. For example, the state of a door (open, semi-closed, and closed) can reduce accuracy of the SLAM approach because it can turn the occupancy gridmap into an incorrect representation of the environment. So, if there is a component capable to detect doors, it is possible to add mechanisms to the SLAM approach that will solve this problem, promoting overall efficiency improvement of Gr2To in delimiting spaces.

The problem of door detection has been studied numerous times in the past. The existing approaches differ in the type of sensors used and the variability of the environment/images which are considered. There are several works presenting techniques for door detection, the most promising ones being vision based and most of them employing generic preprocessing procedures such as Canny edge detection and the Hough Transform algorithm for extracting straight line segments. Shi and Samarabandu (2006) suggests a methodology for detecting corridor and door structures in an indoor environment, where a feedback mechanism based in the hypothesis generation and verification (HGV) method is used to detect corridor and door structures using low level line features in video images. They achieve a 5% of false positives and miss 6% of doors. Birchfield (2008) and Chen et al. (2011) present a vision-based door detection algorithm based on the LineDetection algorithm from Kovesi. To achieve robustness, the features, which include color, texture, and edges intensity, are combined using Adaboost. Tested on a large database exhibiting a wide variety of environmental conditions and viewpoints, the algorithm achieves more than 90% detection rate for only closed doors. Zhang et al. (2009) suggests the use of a stereo vision based algorithm to detect doors in order to generate a series of goal points. The door detection is restricted to a unique door color, and is performed in six steps: Color filtering, Morphological processing, Edge detection, Line extraction, Validation and Locating the door. They present some good results, but this works

for a unique door color which must be distinctive from the walls. Posada et al. (2009) suggests the use of an omnidirectional camera to address the problem of door detection and tracking, where the process is divided in three steps: image processing through canny edge, line processing and door frame recognition. Door detection is based on the matching of detected line segments with prototypical door patterns. In the performed test positives in single images for doors amount to 3% and false negatives occur 5% of the time. Andreopoulos and Tsotsos (2008) propose an active step-by-step approach, using exclusively a stereo vision system in a wheelchair. This is done using the typically Canny edge detection and the Hough transform algorithm. In contrast, Murillo et al. (2008) suggests an approach to capture both the shape and appearance of the door. This is learned from a few training examples, exploiting additional assumptions about the structure of indoor environments. After the learning stage, a hypothesis generation process and several approaches to evaluate the likelihood of the generated hypotheses are described. The approach is tested on numerous examples of indoor environments, showing good performance providing the door extent in images is sufficiently large and well supported by low level feature measurements.

The problem of door detection is a complex problem and there is not any approach capable to detect all doors in all scenarios. However, integrating some of these approaches in the component *TopoFeatures* for door detection will improve the overall performance of HySeLAM. Nevertheless, other strategies should be tested in order to improve door detection performance, as the use of critical points extracted from Voronoi maps to help localizing places with higher probability of door existence.

Chapter 5

Merging the Human description into the HySeLAM Topological map

The next generation of robots which will cooperate with humans should be able to understand a place description given by a human and to store that description in their knowledge structure. For example, if the robot receives from a human the following place description -: *Robot, this house is divided into six rooms. You are on the corridor and on your left there are two rooms, room A and room B. On your right there is the kitchen and the living room. In front of you there is the bathroom.*, it should be able to infer the correct name of each place mapped in its internal map from the human description.

Usually this map is stored in the form of an occupancy grid map, which is a metric description of the place and is very different from the human description. However, the HySeLaM framework makes it possible to translate this occupancy grid map to an augmented topological map, which is an abstraction of the occupancy grid map and produces a map representation much more similar to the human description. The procedure for this translation is detailed in the chapters 3.2 and 4. Although this augmented topological map is more similar to the human description, the place description given by a human must be translated into an augmented topological map. The merging of this augmented topological map with the internal augmented topological map stored into the HySeLAM extension is also required, as depicted in figure 5.1.

In the previous chapters, the way how robot senses the world was described, as well as the way these observations can be abstracted into knowledge representation that is closer to a human description of the world. In this chapter these fundamental questions will be addressed: *How to translate a place description given by a human into an augmented topological map* and *How to merge these two augmented topological maps into a single one*. Section 5.1 presents a global overview of the problem. Section 5.2 presents a global overview of the graph matching problem. Section 5.3 presents our approach to merging the two topological maps.

5.1 Global overview

When a robot that uses the HySeLAM extension, described in chapter 3, arrives at a new environment, it will start building the occupancy grid map by using a navigation strategy and the SLAM approach. When the grid map is considered complete by the robot, the topological HySeLaM layer will call the Gr2To component, which is described in chapter 4. The Gr2To will make the segmentation of the grid map and build the first version of the topological map M_t. This map M_t is not yet complete because there is no human word associated to each segmented place. However, at this moment, the robot is able to ask a human the correct name of each segmented place. This is done by the *PlaceMan* component using a simple procedure *TopoAskPlace*, algorithm 1, which signals the human interaction interface of the robot to ask a human for the place name.

Algorithm 1 HySeLAM - TopoAskPlace

```
while Robot_Live OR Non_Empty_Words_TopoVertex do
    if Topo_Get_Human_Word(X_t^Topo) = Null then
        Signal_Ask_Human_Word_For_Place()
        while X_t^Topo = X_{t-1}^Topo do
            if TopoParser_Word_exists() then
                SP_pX ⟵ TopoParser_Words()
            end if
        end while
        Cancel_Signal_Ask_Human_Word_For_Place()
    end if
end while
```

Where the *Robot_Live* represents the general state of the robot (0 when the robot is disconnected or 1 when it is connected), *Non_Empty_Words_TopoVertex* is the variable that represents the completeness of the topological map (when there are no human words related to all segmented places this takes the value 0), X_t^{Topo} is the current state of the robot in the topological map, SP_{p_X} is a semantic set of words labeling the place p_X, defined in chapter 3.2.1.

The robot with the *TopoAskPlace* and with a topological map is able to interact with a human in order to obtain the human word which tags the place. However, the human will not be available in all places all the time and may not want to follow the robot through the environment in order to tell the name of one place.

Therefore, one way of solving this problem is making it possible for the robot to understand sentences which describe the environment, for example: *Robot, this house is divided into six rooms. You are in the corridor and on your left there are two rooms, room A and room B. On your right you have the kitchen and the living room. The bathroom is in front of you.*

Analyzing a description of the place given by a human, it is possible to find that the human description has a similar structure to a topological map. Therefore, when a human describes the environment the robot should try to translate this description into an augmented graph. If this translation is possible, at some point there will be an augmented graph (human description) and

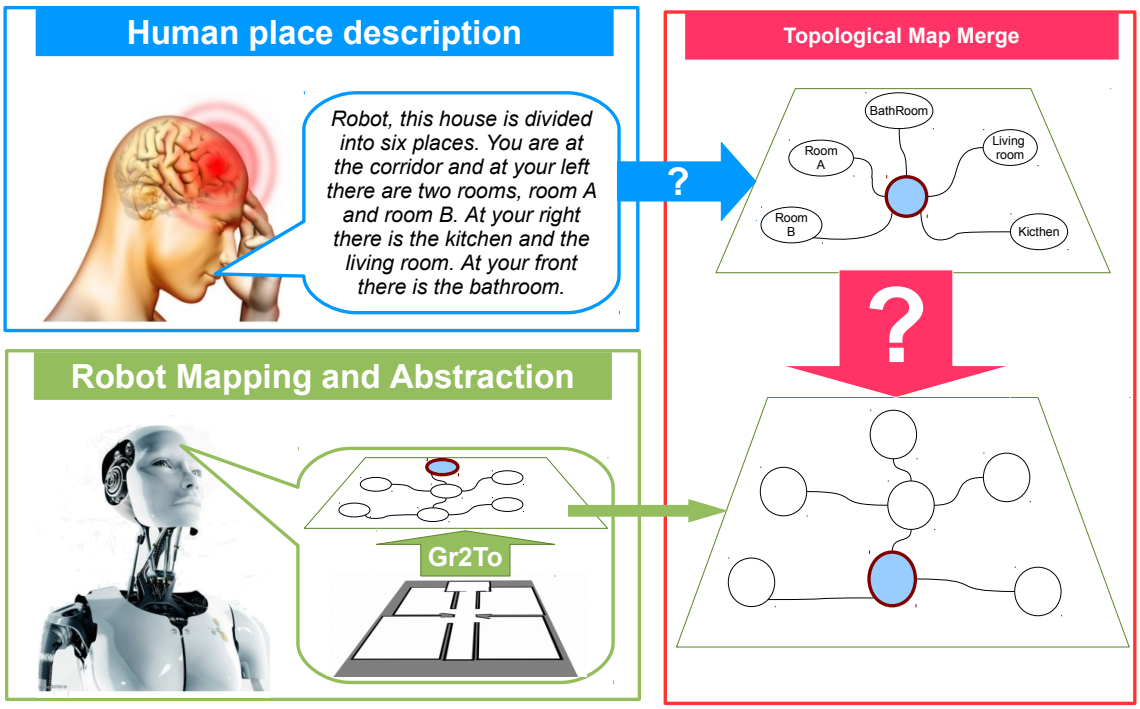

Figure 5.1: Translating the place description given by a human into an augmented topological map and merging this map with the robot's internal knowledge.

a topological map (HySeLAM) that should be merged, as depicted in figure 5.1. From here two fundamental questions emerge that must be answered:

- *How is it possible to translate the place description given by a human into an augmented graph?*
- *How is it possible to merge these two augmented topological maps to form a single map?*

In order to translate a place description given by a human into a graph, it is necessary to execute at least two sequential process: Speech recognition and Natural language processing. These processes are two research fields in computer science and linguistics with very extensive amount of work and also with a great number of challenges to be solved. Speech recognition (SR) is the field which studies the problem of translating spoken words into text, and it is also known by *automatic speech recognition* or simply *speech to text*. Natural language processing (NLP) is the field which studies the problems concerned with the interactions between computers and human (natural) languages. Many challenges in NLP involve natural language understanding to allow computers to derive meaning from human or natural language input. In this work, the NLP must translate a textual description of the place into an augmented topological map.

Speech recognition is a process which occurs outside the HySeLAM extension in the human interaction interface. Beigi (2009) and Pieraccini (2012) presented an in-depth source for updated details on speech recognition theory and practice. Duan et al. (2012) states that the techniques for

automatic speech recognition are classified into three groups: acoustic phonetic approach, pattern recognition approach, and artificial intelligence approach.

Acoustic phonetic approach : The basis of this approach is the postulate that there are finite, distinctive phonetic units (phonemes) in spoken language and that these units are broadly characterized by a set of acoustic properties that are manifested in the speech signal over time.

Pattern recognition approach : The pattern-matching approach involves pattern training and comparison. The essential feature of this approach is the use of a well formulated mathematical framework and the establishment of consistent speech pattern representations,for reliable pattern comparison, from a set of labeled training samples using a formal training algorithm. A speech pattern representation can be in the form of a speech template or a statistical model and can be applied to a sound (smaller than a word), a word, or a phrase. In the pattern-comparison stage of the approach, a direct comparison is made between the unknown speeches (the speech to be recognized) with each possible pattern learned in the training stage in order to determine the identity of the unknown speech according to the goodness of match of the patterns. Usually, pattern recognition approaches are model based, such as the Hidden Markov Model (HMM), the Artificial Neural Networks (ANN), Support Vector Machine (SVM), Vector Quantization (VQ) and Dynamic Time Warping (DTW).

Artificial intelligence approach : This approach is a hybrid of the acoustic phonetics approach and the pattern recognition approach. It exploits the ideas and concepts of acoustic phonetics and pattern recognition methods. Knowledge based approaches use linguistics, phonetics and spectrogram information.

These techniques are the basis of open source tools for speech recognition. CMU Sphinx, Julius, Kaldi, Simon (based on Julius and HTK), iATROS-speech, RWTH ASR, and Zanzibar OpenIVR are some of these open source tools. From a literature review, CMU Sphinx and Julius have the best performance and are the most widely used and based on the pattern recognition approach, which is the most accurate technique for speech recognition.

After the speech recognition process has translated the spoken words describing the place into text, this text must be translated into an augmented graph (topological map). This translation is done by the *TopoParser* component using a natural language processing (NLP) technique, as depicted in figure 5.2.

The process of extracting information from texts using NLP techniques makes it possible to convert human descriptions into written representations manipulated by machines. This process is recognized in numerous applications; some of these applications are outlined by Baeza-Yates (2004): search engines, information retrieval, relationship extraction, automatic translation tools and generation of summaries. The graph extraction from textual description has the same formulation as the information retrieval applications. All approaches for information retrieval (IR) have always used basic natural language processing techniques, as described by Baeza-Yates and

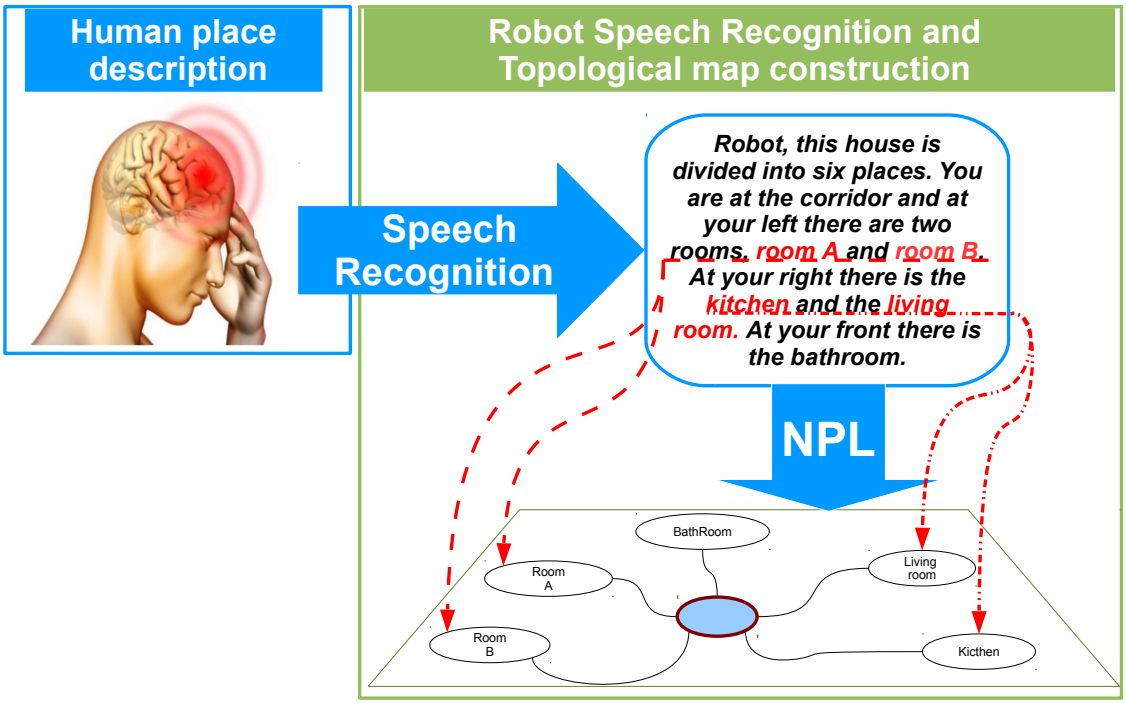

Figure 5.2: Speech recognition and natural language processing for graph extraction.

Ribeiro-Neto (1999). Indeed, Baeza-Yates (2004) outlines and describes the main techniques used in IR, such as tokenization, stopword removal, stemming, and other text normalization tasks which support the approach where every document is viewed as "a bag of words."

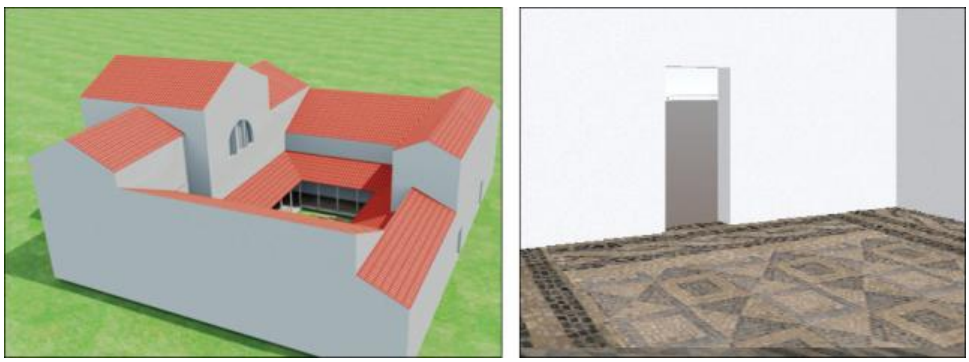

Figure 5.3: Rule-based generation of ancient roman houses, with the outside appearance on the left, and the interior facade on the right. By Adão et al. (2012)

Adão et al. (2012) presents an approach for 3D virtual reconstruction from textual monument descriptions found in books and historical documents; the final result is depicted in figure 5.3. The authors used the Nooj tool to retrieve the information. Nooj is a freeware, linguistic-engineering development environment for formalizing various types of textual phenomena (orthography, lexical and productive morphology, local, structural and transformational syntax). They chose the

Nooj as opposed to other tools, such as the GATE (General Architecture for Text Engineering) by Bontcheva et al. (2002) and the NLTK (Natural Language Toolkit) by Garrette and Klein (2009), due to its characteristics: it is multilingual, it has a large linguistic, lexical and syntactic resource coverage, it processes corpora in several file formats and follows the corpora to build custom grammars with textual patterns which help recognize certain sequences used in annotation systems, as stated by Silberztein (2005) . Other distinguishing factors are the use of graphical grammar design, the application of independent run-time and the use of annotation systems to build the optimized output file.

Based on previous experiences using the Nooj tool for information retrieval, the process of converting a textual description into a topological map, in the HySeLAM extension, is performed using the Nooj tool, which is required by the *TopoParser* component. Grammars are created in Nooj using a graphical editor where text patterns were identified and notes were added to complement the information extracted. The human description of the places represents a text which can be applied to all grammars. Figure 5.4 shows an example of one grammar, which includes text patterns and annotations from the output file.

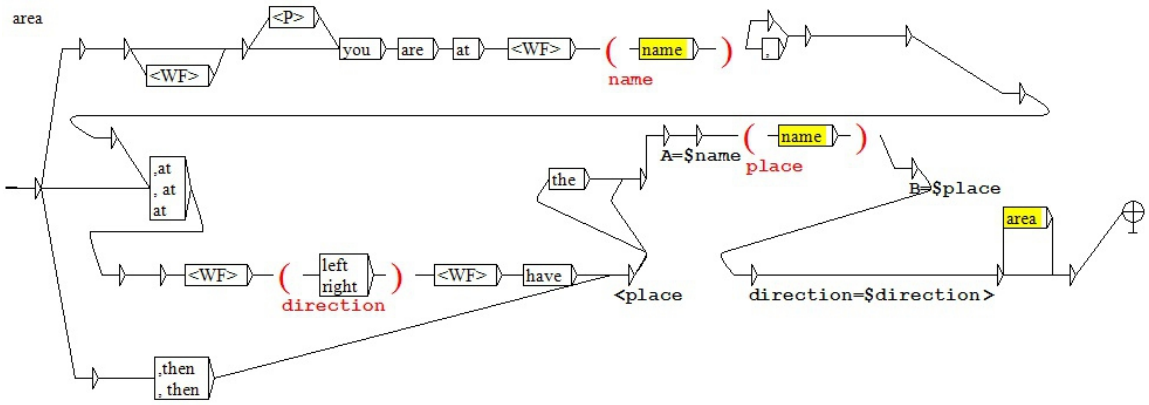

Figure 5.4: Grammar example with defined sequences of texts and annotations for the recognition area, which are used on the recursive method calling the grammar itself.

Therefore, when the robot receives the spoken description of the place, it translates this spoken description into a textual description. This textual description is sent to the *TopoParser* component which will use the Nooj tool with the implemented grammar, as shown in figure 5.4. The final result of the Nooj is an augmented topological map in the form of an XML file.

In order to test the grammar and the performance of the Nooj, the following textual place description was given to the *TopoParser* component : *Robot, you are in the corridor. On your left you have room I-109, then i-110 and i-111. You are in the corridor. On your right you have room i-108 and then i-110. Room i-108 is also known as the industrial robotics lab. Room i-109 is 6 by 8 meters. Room i-110 is 6 by 6 meters. Finally, room i-111 is 6 meters by 4 meters..* The final result was an XML file, figure 5.5, which represents the topological map given by the human description. In this XML file, each tag *place* represents the definition of a new place and in

the graph formalism it represents a vertex augmented with human words (for symbol grounding) and the physical dimensions of the place. Each tag *door* represents a new definition of a door definition and in the graph formalism it is an edge augmented with place connections and physical orientation. The tag textit location stores the state of the robot in the topological map described by the human and at the moment of the human description.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<HumanMAP>
 <!--This Topological Map is HySeLAM Based-->
        <location place="corridor">
        <door A="corridor" B="i-109" direction="left">
        <door A="corridor" B="i-110" direction="left">
        <door A="corridor" B="i-111" direction="left">
        <door A="corridor" B="i-108" direction="right">
        <door A="corridor" B="i-110" direction="right">
        <place semantic="i-108" othersemantic="industrial robotic lab">
        <place semantic="i-109" w="6" h="8">
        <place semantic="i-110" w="6" h="6">
        <place semantic="i-111" w="6" h="4">
</HumanMAP>
```

Figure 5.5: Nooj output file for the description of the place used in the testing scenario.

At this moment, the HySeLaM extension has two topological maps, one connected to the occupancy grid map and extracted by the Gr2To algorithm and another obtained from a human description. These two topological maps are complementary and they should be merged. The first contains the real physical delimitations of the places and the correct location of the places in the robot's navigation framework. The second contains the human names for the place and their connections, and sometimes their physical dimensions and spatial orientation. To solve this problem, it is necessary to analyze the graph matching theory, which is done in section 5.2. The proposed approach to merge these two graphs is described in detail in section 5.3.

5.2 Graph matching

In order to solve the question *How can the two augmented topological maps be merged to form a single map*, it is necessary to analyze the graph theory. Graphs are widely studied in mathematics and computer science and are widely used to model or represent processes or knowledge. In computer vision, graphs have proved to be an effective way of representing objects, by Eshera and Fu (1986). Graph matching is considered one of the most complex problems in object recognition in computer vision, and as state by Bienenstock and von der Malsburg (1987) its complexity comes from its combinatorial nature.

This section explains the graph matching problem, firstly by introducing an amount of notation and terminology, and then by classifying the different graph matching types.

5.2.1 Basic notation and terminology

In its basic form, a graph $G = (V,E)$ is composed of vertices and edges. V is the set of vertices (also called nodes or points) and $E \subset V \times V$ (also defined as $E \subset [V]$ in the literature) is the set of edges (also known as arcs or lines) of graph G. A graph G and its set of vertices V are not always strictly differentiated, and commonly a vertex u is said to be in G when it should be said to be in V.

The order (or size) of a graph G is defined as the number of vertices of G and it is represented as $|V|$ and the number of edges as $|E|$.

If two vertices in G, say u, $v \in V$, are connected by an edge $e \in E$, this is denoted by $e = (u,v)$ and the two vertices are said to be adjacent or neighbors. Edges are said to be undirected when they have no direction, and a graph G containing only such types of graphs is called undirected. When all edges have directions and therefore (u,v) and (v,u) can be distinguished, the graph is said to be directed. Usually, the term arc is used when the graph is directed, and the term edge is used when it is undirected. This section thesis, will mainly use undirected graphs, although graph matching can also be applied to directed graphs. In addition, a directed graph $G = (V,E)$ is called complete when there is always an edge $(u,u) \in E = V \times V$ between any two vertices u, u' in the graph.

Graph vertices and edges can also contain information. When this information is a simple label (for instance, a name or a number) the graph is called *labelled graph*. Other times, vertices and edges contain some more information. These are called vertex and edge *attributes*, and the graph is called *attributed graph*. More frequently, this concept is specified by differentiating *vertex-attributed* (or *weighted graphs*) and *edge-attributed graphs*.

A path between any two vertices $u,u' \in V$ is a non-empty sequence of k different vertices $< v0,v1,\ldots,vk >$ where $u = v_0, u' = v_k$ and $(v_{i-1},v_i) \in E, i = 1,2,\ldots,k$. Finally, a graph G is said to be acyclic when there are no cycles between its edges, regardless of whether the graph G is directed or not.

5.2.2 Definition and classification of graph matching problems

When graphs are used to represent objects or images, vertices usually represent regions (or features) of the object or image, and the edges between them represent the relations between regions. In the augmented topological map, formalized in HySeLAM, vertices represent a physical delimited region, and the edges between them represent the connections between regions (such as real or virtual doors).

Graphs can be used to represent objects or general knowledge, and they can be either directed or undirected. When edges are undirected, they simply indicate that there is a relation between two vertices. On the other hand, directed edges are used when relations between vertices are considered in a non symmetric way.

Two graphs are given in model-based pattern recognition problems: the model graph G_M and the data graph G_D. The comparison involves verifying whether they are similar or not. Generally

speaking, it is possible to define the graph matching problem as follows: given two graphs $G_M = (V_M, E_M)$ and $G_D = (V_D, E_D)$, with $|V_M| = |V_D|$, the problem is to find a one-to-one mapping $f : V_D \rightarrow V_M$ such that $(u,v) \in E_D \; iff \; (f(u), f(v)) \in E_M$. When such a mapping f exists, this is called an *isomorphism*, and G_D is said to be isomorphic to G_M. This type of problem is called exact *graph matching*. The term *inexact* applied to some graph matching problems means that it is not possible to find an isomorphism between the two graphs to be matched. This is the case when the number of vertices is different in both the model and data graphs.

In the cases where no isomorphism can be expected between both graphs, the graph matching problem is defined as finding the best match between them. This leads to a class of problems known as inexact graph matching. In that case, the matching aims at finding a non-bijective correspondence between a data graph and a model graph. Next, $|V_M| < |V_D|$ will be used.

The best correspondence of a graph matching problem is defined as the optimum of some objective function that measures the similarity between matched vertices and edges. This objective function is also called *fitness function* (also known as *energy function*).

An inexact graph matching problem contains $|V_M| < |V_D|$, and the goal is to find a mapping $f' : V_D \rightarrow V_M$ such that $(u,v) \in E_D \; iff \; (f(u), f(v)) \in E_M$. This corresponds to the search for a small graph within a larger graph. An important sub-type of these problems are *sub-graph matching problems*, in which there are two graphs $G = (V,E)$ and $G' = (V',E')$, where $V' \subseteq V$ and $E' \subseteq E$, and in this case the aim is to find a mapping $f' : V' \rightarrow V$ such that $(u,v) \in E' \; iff \; (f(u), f(v)) \in E$. When such a mapping exists, this is called *subgraph matching* or *subgraph isomorphism*.

Exact and *inexact graph matching* will be the terms used here, to differentiate these two basic types of graph matching problems. However, in the literature this type of graph matching problem is also called *isomorphic* and *homomorphic* graph matching problems respectively.

In some inexact graph matching problems, the problem is finding a one-to-one correspondence, with the exception of vertices in the data graph which have no correspondence whatsoever. More formally, with two graphs $G_M = (V_M, E_M)$ and $G_D = (V_D, E_D)$, the problem consists of searching for a *homomorphism* $h : V_D \rightarrow V_M S \cup \{\varnothing\}$, where $\{\varnothing\}$ represents the null value, meaning that when for a vertex $a \in V_D$ we have $h(a) = \varnothing$ there is no correspondence in the model graph for vertex a in the data graph. This value $\{\varnothing\}$ is known in the literature as the null or dummy vertex.

Note that in these cases the fitness function that measures the fitness of homomorphism h has to be designed taking into account that it should encourage the graph matching algorithm to reduce the number of vertices $a1, a2, \ldots, a_n \in V_D$ meeting the condition $h(a_i) = \varnothing, i = 1,2,\ldots,n$. In other words, nothing prevents the homomorphism h for which $\forall a_i \in V_D h(a_i) = \varnothing$ is valid, but this solution would not represent any satisfactory result.

Some other graph matching problems allow many-to-many matches, that is, with two graphs $G_M = (V_M, E_M)$ and $G_D = (V_D, E_D)$, the problem consists of searching for a homomorphism $f : V_D \rightarrow W$ where $W \in P(V_M) \setminus \{\varnothing\}$ and $W \subseteq V_M$. If also dummy vertices are also used, W can take the value $\{\varnothing\}$, and therefore $W \in P(V_M)$. This type of graph matching problem is more

difficult to solve, as the complexity of the search for the best homomorphism contains various N_c combinations and, therefore, the search space of the graph matching algorithm is much larger.

$$N_c = \left(2^{|V_M|}\right)^{|V_D|} \tag{5.1}$$

The total number of these combinations is given by equation 5.1. For example, for the problem of matching two graphs with four vertices each $|V_M| = |V_D| = 4$, the number of possible combinations is 65.535, and for $|V_M| = |V_D| = 5$ there are 33.554.432 total combinations. However, this amount of total combinations can be significantly reduced by considering several constraints in the matching process.

5.2.3 Approaches to graph matching

The problem of graph matching has widely studied over the last forty years. Conte et al. (2004) presents a detailed literature review since 1970 focusing on the graph matching techniques applied to different application areas, such as 2D & 3D Image Analysis, Document Processing, Biometric Identification, Image Databases, Video Analysis, and Biomedicine and Biology.

In many applications, a crucial operation is comparing two objects or an object and a model to which the object could be related. When structured information is represented by graphs this comparison is made using some form of graph matching. As previously mentioned, graph matching is the process of finding a correspondence between the vertices and the edges of two graphs that meets some more or less stringent constraints, ensuring that similar substructures in one graph are mapped to similar substructures in the other.

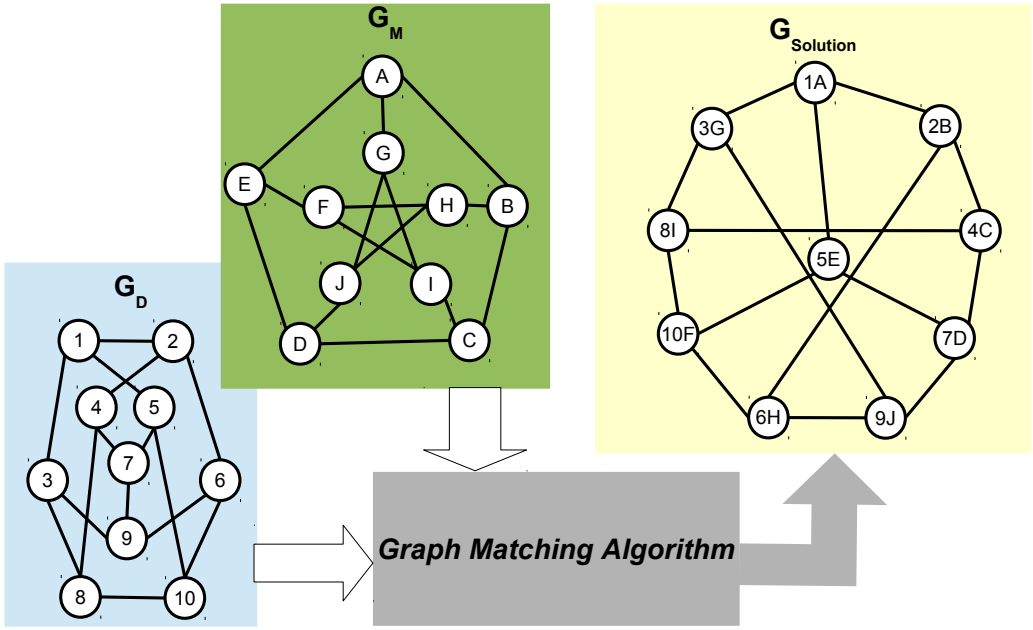

Figure 5.6: The graph matching problem.

Exact graph matching is characterized by the fact that the mapping between the vertices of the two graphs must be edge-preserving, in the sense that if two vertices in the first graph are linked by an edge, they are mapped to two vertices in the second graph which are also linked by an edge, as depicted in figure 5.6 . When exact graph matching is possible or required, tree search is the most widely used concept to develop algorithms. For example, Ullmann (1976), Messmer and Bunke (1998), Cordella et al. (2001), and, Dijkman et al. (2009) propose algorithms based on the tree search concept. These algorithms are based on some form of tree search with backtracking. These algorithms start with a partial match, which is initially empty, and iteratively expanded by adding new pairs of matched vertices. The pair is chosen using not only the necessary conditions to ensure compatibility with the constraints imposed by the matching type with regard to the vertices mapped so far, but also heuristic conditions to prune unfruitful search paths as early as possible. At some point, the algorithm reaches the complete matching solution or a state where it cannot add new pairs of matched vertices, which meets the constraints. In the latter case, the algorithm backtracks, which means that it undoes the last additions and tries alternative paths for the matching solution. This backtrack step will happen until it is not possible to add new pairs of matched vertices or until all possible mappings that meet the constraints have already been tested. Although there are several different implementations for this kind of algorithm, the main differences are found in the strategies for selecting paths and in the backtrack step. Depth-first search, or branch and bound, is the simplest strategy in that it requires less memory than others and lends itself very well to a recursive formulation.

Other concepts are found in literature, contrasting with tree search concept. Nauty, in McKay (1981), is an interesting matching algorithm for exact graph matching not based on tree search. Nauty is based on the group theory. In particular, it uses results from this theoretical framework to efficiently construct the automorphism group of each of the input graphs. A canonical labeling is derived from the automorphism group, which introduces a node ordering that is uniquely defined for each equivalence class of isomorphic graphs. Therefore, two graphs can be checked for isomorphism by simply verifying the equality of the adjacency matrices of their canonical forms. However, this algorithm deals only with the isomorphism problem and under particular conditions it can be outperformed by other tree search based algorithms, in Santo et al. (2003).

In some circumstances, the stringent constraints imposed by exact matching are too rigid to be applicable in real applications. In many applications, the observed graphs are subjected to deformations due to several causes; intrinsic variability of the patterns, noise in the acquisition process, and the presence of nondeterministic elements in the processing steps leading to the graph representation, are among the possible reasons for having actual graphs differing from their ideal models. Therefore, the matching process must be tolerant and accommodate the differences by relaxing, to some extent, the constraints that define the matching type. This can be useful even when no deformation is expected. It is for these reasons that the inexact graph matching was also widely explored and algorithms for this problem were developed. Usually, in these algorithms the matching is note forbidden between two vertices that do not satisfy the edge-preservation requirements of the matching type. Instead, it is penalized by assigning a cost that may take into

account other differences, for example among the corresponding node/edge attributes. Therefore the algorithm must find a mapping that minimizes the matching cost.

In inexact graph matching there are two types of algorithms, optimal and approximate (or suboptimal). Optimal algorithms will always find a solution that represents the global minimum of the matching cost. On the other hand, approximate matching algorithms only ensure that a local minimum of the matching cost is found. For some algorithms, the definition of the matching cost is based on an explicit model of the errors (deformations) that may occur, such as missing vertices,or assigning a possibly different cost to each kind of error. These algorithms are often denoted as error-correcting or error-tolerant. Another way of defining a matching cost is by introducing a set of graph edit operations, such as node insertion, node deletion, and assigning a cost to each operation, and the cheapest sequence of operations required to turn one of the two graphs into the other is computed. The cost of this sequence is called graph edit cost.

The tree search concept is also widely used to solve the inexact graph matching. However, in these problems the tree search algorithm is driven by the cost of the partial matching obtained so far, and by a heuristic estimate of the matching cost for the remaining vertices. This information can be used either to prune unfruitful paths in a branch and bound algorithm, or to determine the order in which the search tree must be traversed, as in the A* algorithm. Tsai and Fu (1979) proposes one of the first tree based inexact algorithms. The work introduces a formal definition of error-correcting graph matching of Attributed Relational Graphs (ARG), based on the introduction of a graph edit cost. The proposed algorithm only takes into account the operations of node and edge substitution, omitting insertion and deletion. Hence, the graphs being matched must be structurally isomorphic. The proposed heuristic is based on the computation of the future node matching cost by disregarding the constraint where the mapping has to be injective, and the search method makes it possible to find the optimal solution. Tsai and Fu (1983) proposes an upgrade to the method by considering the insertion and deletion of vertices and edges, and Wong et al. (1990) suggest an approach wich improves the heuristic for error-correcting monomorphism, also taking into account the future cost of edge matching.

The matching methods based on tree search rely on a formulation of the matching problems directly in terms of graphs. A radically different approach is to cast graph matching, which is inherently a discrete optimization problem, into a continuous, nonlinear optimization problem. Then, there are many optimization algorithms that can be used to find a solution to this problem. These algorithms do not ensure the optimality of the solution, even if the most sophisticated includes techniques to avoid trivial local optima. Furthermore, the solution found needs to be converted back from the continuous domain into the initial discrete problem by a process that may introduce an additional level of approximation. These methods are proposed by Fischler and Elschlager (1973), Wilson and Hancock (1997), and by Torsello and Hancock (2003).

The spectral method is another popular approach widely used for the inexact graph matching problem. Spectral methods are based on the following observation: the eigenvalues and the eigenvectors of the adjacency matrix of a graph are invariant with regard to node permutations. Hence, if two graphs are isomorphic, their adjacency matrices will have the same eigenvalues and

eigenvectors; however, the inverse is not true. An important limitation of these methods is that they are purely structural, in the sense that they are not able to exploit node or edge attributes, which often convey very relevant information for the matching process. Furthermore, some of the current spectral methods are capable of dealing only with real weights assigned to edges by using an adjacency matrix with real valued elements. These methods are proposed by Umeyama (1988), Shapiro and Brady (1992), Xu and King (2001), and by Duchenne et al. (2011).

As noted earlier, the graph matching literature is extensive, and many different types of approaches have been proposed. An incomplete list includes:

- Tree search, used in Tsai and Fu (1979), Tsai and Fu (1983), Wong et al. (1990), Messmer and Bunke (1998), Cordella et al. (2001), and, Dijkman et al. (2009).
- Spectral methods, used in Shapiro and Brady (1992), Leordeanu and Hebert (2005), Zaslavskiy et al. (2009), Duchenne et al. (2011), and Shokoufandeh et al. (2012).
- Relaxation labeling and probabilistic approaches, used in Wilson and Hancock (1997) and Caetano et al. (2004);
- Semidefinite relaxations, used in Schellewald (2005).
- Replicator equations, used in Pelillo (1999).
- Graduated assignment, used in Gold and Rangarajan (1996).
- RKHS methods, used in Wyk (2002).
- Factorized Graph Matching, used in Zhou and Torre (2012).
- Learning graph matching, used in Caetano and McAuley (2009).

From this literature review, the matching methods based on tree search are the simplest and can be very easily adapted to take into account the attributes of vertices and edges in constraining the desired matching, with no limitations on the kinds of attributes that can be used. This approach is very important for a large set of applications where attributes often play a key role in reducing matching computational time.

To solve a generic graph matching problem, several open-source implementations are available, such as the NAUTY by Brendan D. McKay, the VFLib graph matching library, from the University of Naples, the GraphBlast or GraphGrep, the Graph Matching tool, by Giugno and Shasha (2002), the Combinatorica, implemented for Mathematica, the LEMON the Library for Efficient Modeling and Optimization in Networks, the GMTE the Graph Matching and Transformation Engine, and the Factorized Graph Matching by Zhou and la Torre (2013).

5.3 The TopoMerg Approach

When the robot arrives at a new environment, it will start building the grid map using the SLAM algorithm. When the grid map is considered to be completed by the robot, the topological HySeLaM layer will initiate the Gr2To algorithm. Gr2To segments the grid map and builds the first version of the topological map M_t. This map M_t is not complete because there is no any human word associated to each place.

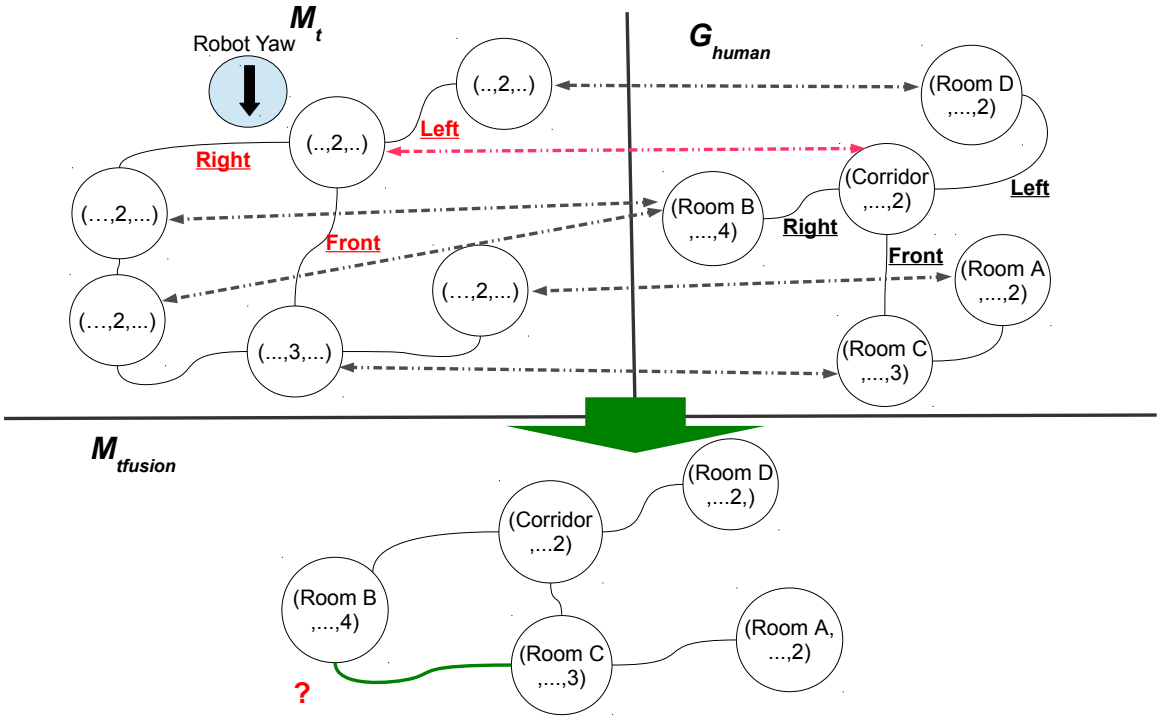

Figure 5.7: Merging map illustrations: M_t represents the topological map obtained from the grid-map and G_{human} represents the topological map obtained from the natural language (human). The red arrow is the partial a priori knowledge inferred from the human interaction and it is a constraint that simplifies the matching problem.

At some moment, the robot will ask a human to describe the environment with the names of the places and their relation. From this natural language, the robot will extract another topological map G_{human}. From a geometric perspective, this topological map is less detailed and precise than M_t. These maps are complementary and should be merged, as depicted in figure 5.7.

The HySeLaM layer will obtain two topological maps, one from the SLAM gridmap abstraction and another from the human description. These maps can be classified as attributed graphs.

The attributes of vertices in G_{human} are the human words given to the place and an approximate dimension of the place. The attributes of vertices in M_t are the real and virtual walls that delimit the place and the position of the central point in the place. The edges in G_{human} provide information about the connections and orientation between the places. The edges in M_t are the predicted doors with their location.

These maps are complementary and should be merged. For that , it is necessary to use approaches related to inexact graph matching problems. From the conclusions of section 4.4, it is possible to state that the Gr2To will provide an over-segmentation of the place, which will lead to a typical graph matching problem, allowing one-to-many matches. This means that, with two graphs $M_t = (V_M, E_M)$ and $G_{human} = (V_G, E_G)$, the problem consists of searching for a homomorphism $h : V_G \rightarrow W$, where $W \in P(V_M) \setminus \{\varnothing\}$, $W \subseteq V_M$, and minimizing the fitness function f. f is the function that measures the fitness of the homomorphism h. If dummy vertices are also used, W can take the value $\{\varnothing\}$, and therefore $W \in P(V_M)$. It is difficult to solve this type of graph matching problem, as the complexity of the search for the best homomorphism presents various solution combinations, and therefore the search space of the graph matching algorithm is much bigger than other kinds of graph matching problems. In the literature this is sometimes referred to as NP-complete problem (see Lovász and Plummer (1986)).

Partial a priori knowledge is inferred from the human description in the HySeLaM framework. This knowledge contains the correspondence between two vertices $h_{GM} : v_G \rightarrow v_M$. This correspondence appears during the human-robot interaction and relates the current robot location and the human location, during the environment description. This partial a priori knowledge is a constraint to the inexact graph matching problem, which reduces the problem's complexity.

5.3.1 The fitness function

In order to quantify the matching solution, the fitness function $f(M_w, G_{human})$ was created taking into account the matching between vertex features (place area/dimension), corresponding a vertex of G_{human} to a vertex of M_w, and edge features (connection and orientation) from G_{human} to M_w. M_w is an intermediate graph constructed from M_t (topological map obtained from SALM) and from homomorphism h. The quantification of matching quality is given by a real number, between 0 and 1, $f(h, M_t, G_{human}) \in [0,1] \in \Re$, where 1 represents a perfect match between G_{human} and M_w.

$$\begin{aligned} f(M_w, G_{human}) = & \gamma_v \frac{1}{n_v^G} \sum_{i=0}^{n_v^G} (Eh_n(i)) + \gamma_a \frac{1}{n_{va}^G} \sum_{i=0}^{n_v^G} (Ah_n(i)) \\ & + \gamma_e \frac{1}{N_e^G} \sum_{i=1}^{n_e^G} \sum_{j=0}^{n_e^G} (Ch_n(i,j)) + \gamma_d \frac{1}{N_{ea}^G} \sum_{i=1}^{n_e^G} \sum_{j=0}^{n_e^G} (Dh_n(i,j)) \end{aligned} \tag{5.2}$$

Where: n_v^G is the number of vertices in G_{human}; n_{va}^G is the number of vertex in G_{human} with a atributed value in the area parameter; N_{ea}^G is the total number of edges possible for the number of edges in G_{human}, so $N_{ea}^G \sum_{j=1}^{n_v^G - 1} j$; N_{ea}^G is the number of edges in G_{human} with a attributed value in the orientation parameter (for example, left, right and front); γ_v, γ_a, γ_e, and γ_d are weight factors for equation normalization and they are parameterizable, with the constraint $\gamma_v + \gamma_a + \gamma_e + \gamma_d = 1$; $Eh_n(i)$ is a function that returns 1 if the function h relates the vertex v_G^i (from G_{human}) to any vertex of the M_w graph; $Ah_n(i)$ is a function that returns the matching quantification based on the area

feature in the v_G^i (from G_{human}) and v_M^i (from M_w); $Ch_n(i,j)$ is a function that returns 1 if there is the same existence connection in M_w and in G_{human}, for the vertex i and j, and 0 otherwise; $Dh_n(i,j)$ is a function that returns the matching quantification based on the edge orientation for the vertex i and j taking into account the referential orientation ψ_i (obtained at the moment of robot-human interaction).

$$Ah_n(i) = 1 - \frac{|A(v_G^i) - A(v_M^i)|}{max(A(v_G^i), A(v_M^i))} \tag{5.3}$$

$$Dh_n(i,j) = 1 - \frac{Ne(i,j)}{\pi} \in [0,1] \tag{5.4}$$

where $A(v_D^i)$ is a function that returns the area size in the vertices i from graph D. $Ne(e_M^i, e_G^i)$ is function that returns a positive and normalized difference from two edges, taking into account the edges that connect the vertex i to vertex j in G_{human} and in M_w.

5.3.2 Fitness function validation

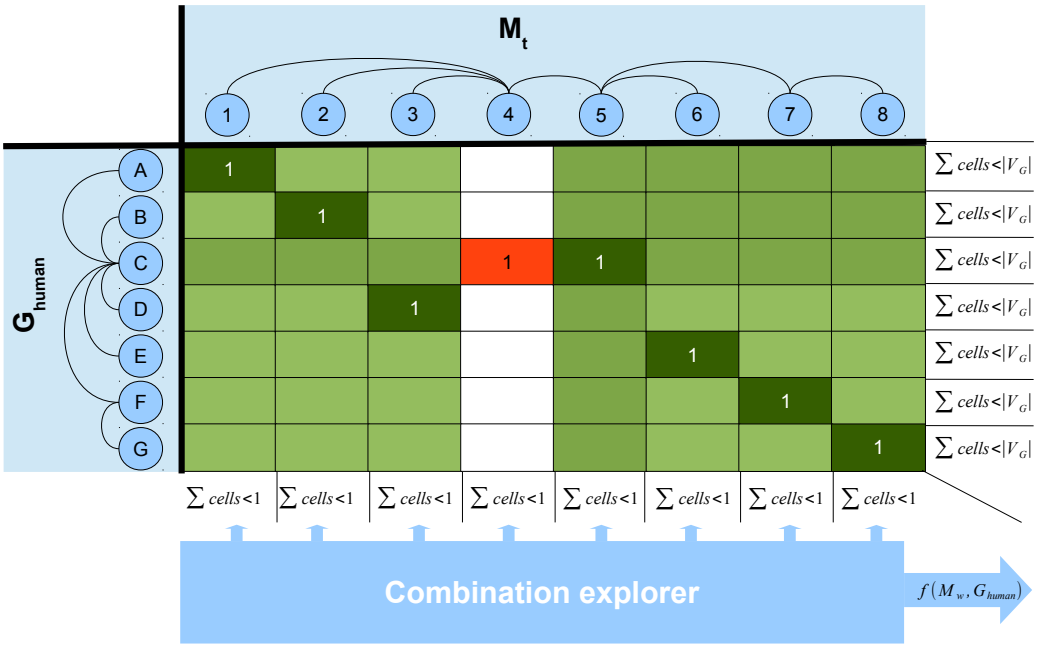

Figure 5.8: The matching matrix H_{match} of the TopoMerg library and the solution explorer. The red cell represents the constraint of the solution.

In order to validate the fitness function and evaluate the influence of different levels of detailed descriptions obtained from the human-robot interaction, the *TopoMerg* library was created to explore all possible combinations for the h function. For this, the *TopoMerg* library creates a matching matrix H_{match} that represents the solution for the homomorphism function h, as depicted in figure 5.8. This matrix has $|V_G|$ rows and $|V_M|$ columns; the rows are related to the vertex of

G_{human} and the columns are related to the vertex of M_t. If the cell of row i and column j has the value 1, this means that vertex i from G_{human} is associated with vertex j from M_t.

Considering that there is a problem of one-to-many matches, this means: each vertex from M_t (topological map) can only be linked to one vertex in G_{human}. However, the opposite is not true; each vertex in G_{human} can be associated to an empty set of vertices or to a set of vertices of M_t. In the matching matrix, this one-to-many matching rule can be represented by two constraints: each cell can take two values 0 or 1, and the sum of all cells in each column must be less than or equal to 1.

Taking into account that the robot can infer the correct matching of one vertex of M_t to another vertex in G_{human}, from the human interaction and as depicted in figure 5.7, this means that there is a constraint that will simplify the matching problem. Therefore, the number of possible combinations for this exploration is given by the following equation:

$$N = (|V_G| + k_s)^{(|V_M|-1)} \tag{5.5}$$

where, k_s takes the value 1 ,if a vertex of G_{human} can be matched to a dummy vertex $\{\varnothing\}$, or takes the value 0 if the vertex of G_{human} must match any vertex of M_t. For example, for $|V_G| = 7$, $|V_G| = 8$ and $k_s = 0$ there are 823,543 possible combinations, and if k_s takes the value of 1 there are 2,097,152 possible combinations.

In order to explore all these combinations, a numeral system is created in each exploration with $|V_G| + ks$ symbols and with $|V_M|$ positions, which starts at the lowest number and iterates to the highest. The symbol in each j position will reflect the number one position in the j column of the matching matrix. At each step, the symbol in the lower position is incremented; when this symbol reaches the higher value, it is reset to the lower value and it is carried the value one to the next position, similarly to typical numeral system. However, in this algorithm the symbol in the position related to the constraint or constraints will remain static during this exploration. In the algorithm this is reflected as a vector of $|V_M|$ integers, which can take any value from k_s to $|V_G|$, this vector will be referred to as H^{State}_{match}.

For each iteration, there will be a new matching matrix which will reflect a new M_w graph. This new M_w graph has exactly the same number of vertices as G_{human} however, each vertex of M_w is related to a set of vertices from M_t. This new M_w will have a new set of edges, which is a projection from M_t into M_w. Taking into account that M_t, M_w, and G_{human} are undirected graphs, it is possible to represent the edge connections of G_{human} and M_w using a square matrix as the one depicted in figure 5.9. This edge matching matrix has exactly $|V_G|$ columns per $|V_G|$ rows, and the upper triangle stores the edge connections of G_{human}, which will remain static during the exploration, and the lower triangle stores the edge connections of M_w, which will be updated during the exploration. Each cell of this matrix can take the value 1 or 0; 1 if there is a connection between vertices or 0 if a such connection does not exist. Without taking into account attributes of the edges and vertices, if the edge matching matrix is a symmetric matrix it is possible to say that the two graphs M_w and G_{human} are a perfect matching.

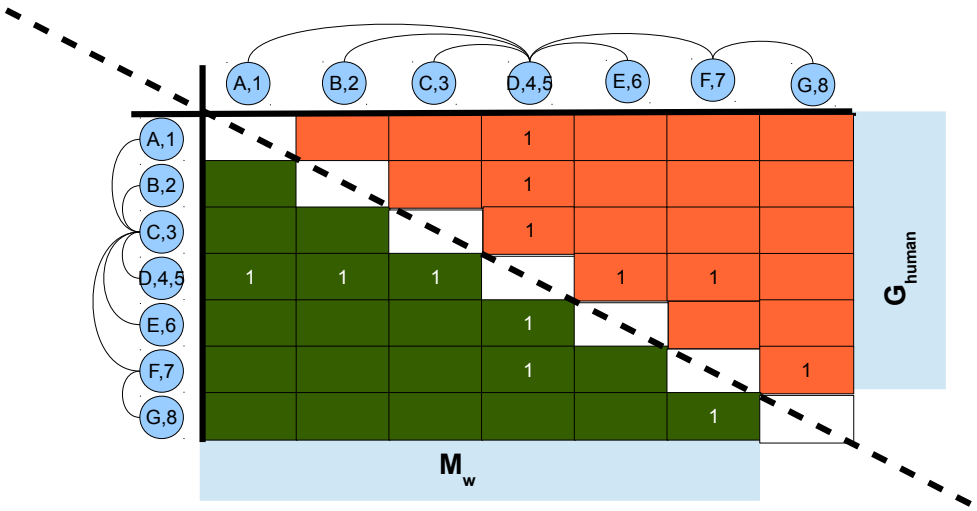

Figure 5.9: Edge matching matrix stores the edge connections of M_w and G_{human}.

This edge matching matrix E_{match} will simplify the function $Ch_n(i,j)$ that is used in the fitness function. Therefore, using E_{match} the function $Ch_n(i,j)$ is given by:

$$Ch_n(i,j) = \begin{cases} 1, & \text{if } E_{match}(i,j) = E_{match}(j,i) \\ 0, & \text{if } E_{match}(i,j) \neq E_{match}(j,i) \end{cases} \tag{5.6}$$

This definition of $Ch_n(i,j)$ penalizes following cases similarly: the nonexistence of a similar edge in M_w as in G_{human} and the existence of one edge in M_w that is not present in G_{human}. However, sometimes it may be important to have less penalties if there is one edge in M_w that does not exist in G_{human} because the human has forgotten to describe that connection. In that case, a third parameterizable value should be considered.

Besides this edge matching matrix E_{match}, a second matrix $E_{match}^{orientation}$ is considered the edge matching matrix for the edge orientation attribute. The edge orientation attribute can take five values: left, right, front, back and empty. These values are translated into a positive real value in radians and relate a secondary vertex to a main vertex. For example, if the main vertex is connected to four vertices, each one having different values, these edge attributes will be translated as follows, $front = 0$ $right = \frac{\pi}{2}$ $left = \frac{4*\pi}{3}$ $back = \pi$. However, if there are n edges with the same value, they will be spaced by $\frac{\pi}{(2*n)}$ radians, with the first edge described taking the value closest to π however, they are all focused on a central value, defined for each of the four values (left, right, front, back). Therefore, each cell of the upper triangle of $E_{match}^{orientation}$ matrix will take this translated value, or -1 if there is no edge connecting vertex i and j or if the edge attribute is empty. This upper triangle of $E_{match}^{orientation}$ will remain static during the exploration, while the lower triangle is updated in every

time the matching matrix is updated. This update will take into consideration the projection of M_t into M_w and the E_{match} matrix. All edges of M_w will have the orientation attribute updated by the following equation:

$$\psi = \arctan\left(v_k(y) - v_m(y), v_k(x) - v_m(x)\right) + \psi_{robot}^{human} \tag{5.7}$$

where, $v_m(y)$ and $v_m(x)$ are the coordinates of the central position of all vertices of M_t related to the vertex (v_i or v_j) of M_w which is related to a main vertex of G_{human}; $v_k(y)$ and $v_k(x)$ are the coordinates of the central position of all vertices of M_t related to the vertex (v_i or v_j) of M_w which is not related to a main vertex of G_{human}; ψ_{robot}^{human} is the orientation of the robot at the moment of the description. Therefore using E_{match} the function $Ne(i,j)$ is given by:

$$Ne(i,j) = \begin{cases} |E_{match}(i,j) - E_{match}(j,i)|, & \text{if } E_{match}(i,j) \geq 0 \\ 0, & \text{if } E_{match}(i,j) = -1 \end{cases} \tag{5.8}$$

With these three matrices matching matrix H_{match}, edge matching matrix E_{match} and the edge matching matrix for the edge orientation attribute $E_{match}^{orientation}$, it is possible to search in all possible combinations the best matching for M_t and G_{human} by using the simple procedure *TopoMergExplorer*, described by the algorithm 2.

Algorithm 2 HySeLAM - TopoMergExplorer

$H_{match}^{State} \leftarrow initBy\left(G_{human}, M_t\right)$
$E_{match} \leftarrow initBy\left(G_{human}\right)$
$E_{match}^{orientation} \leftarrow initBy\left(G_{human}\right)$
$\lambda_{best} \leftarrow 0$
repeat
  $H_{match}^{State} \leftarrow incrementBy\left(H_{match}^{State}, H_{match}^{Constraints}\right)$
  $H_{match} \leftarrow updateBy\left(H_{match}^{State}\right)$
  $M_w \leftarrow updateBy\left(H_{match}, M_t\right)$
  $E_{match} \leftarrow updateBy\left(M_w\right)$
  $E_{match}^{orientation} \leftarrow updateBy\left(M_w, M_t\right)$
  $\lambda\left(H_{match}^{State}\right) \leftarrow f\left(M_w, G_{human}\right)$
  if $\lambda_{best} < \lambda\left(H_{match}^{State}\right)$ **then**
    $\lambda_{best} \leftarrow \lambda\left(H_{match}^{State}\right)$
    $H_{match}^{best} \leftarrow H_{match}^{State}$
  end if
until $H_{match}^{State} = H_{match}^{StateMaximum}$

Using the *TopoMergExplorer*, it is possible to characterize the matching quality of all combination spaces. In order to test this matching function, two attributed graphs were used, one extracted from a topological map obtained in a virtual scenario, and another extracted from a human description of this virtual scenario, using techniques previously described.

The topological map obtained is depicted in figure 5.10. The human description was *"Robot, you are in the corridor. On your left you have room Room B, then Room A. You are in the corridor.*

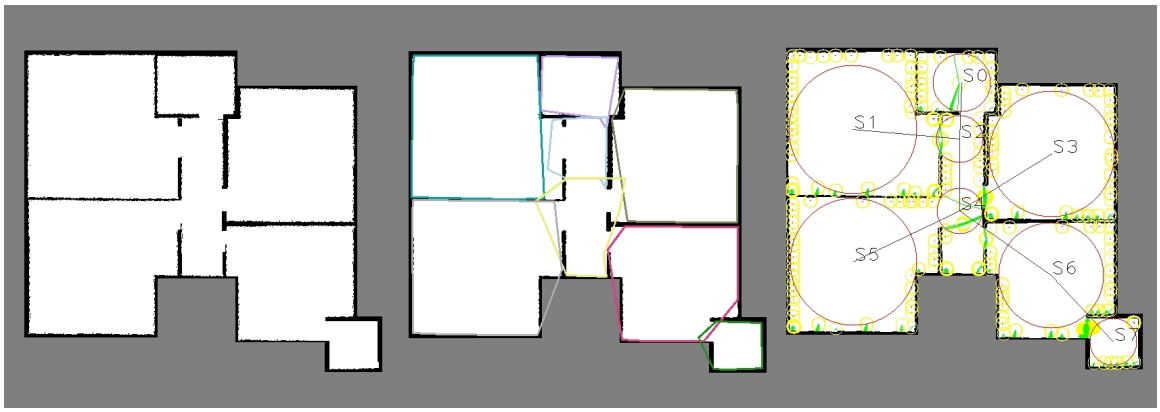

Figure 5.10: From left to right, the gridmap obtained from hector slam in a virtual scenario , the space segmentation obtained from the Gr2To, and the topological map extracted from the Gr2To.

On your right you have the kitchen and then Living Room. You are in the corridor. In front of you have the WC. In the kitchen you have the Laundry. Room A is 6 by 8 meters. Room B is 6 by 6 meters. The laundry is by 2 by 2 meters.".

Applying these two topological maps to the *TopoMergExplorer* algorithm, it is possible to obtain the H_{match}^{best} depicted in figure 5.11. The constraint applied was the vertex "corridor" associated with vertex 4 of the topological map. With the weights parameters of the fitness function taking the values $\gamma_v = 1/4$, $\gamma_e = 1/4$, $\gamma_a = 1/4$, and $\gamma_d = 1/4$.

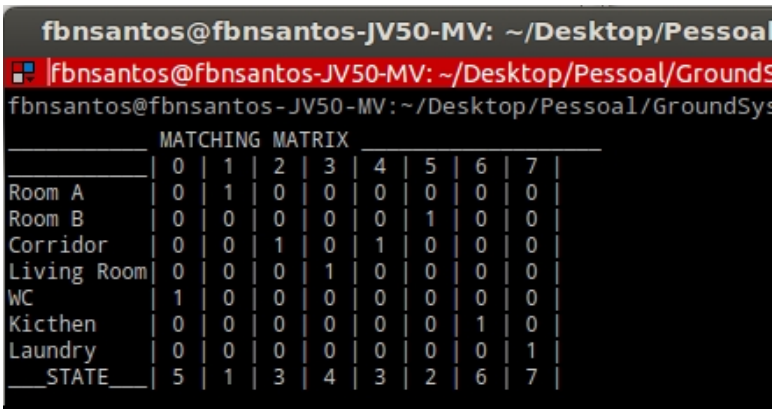

Figure 5.11: The best matching matrix H_{match}^{best} for the two topological map obtained from the human description and from the Gr2To.

This best matching matrix is the correct solution for these two graphs, the human description and the topological map in figure 5.10. Furthermore, another test was conducted in order to study the influence of the weight parameters on the fitness functions. For this experiment, a set of four tests was conducted with different values for these parameters, which are summarized in table 5.1.

For this test, $k_s = 0$ was used and a total of 823542 combinations have been explored in each of the four runs. From table 5.1, it is possible to conclude that the vertex and edge attributes

Table 5.1: Test of fitness function with a set of four values for the parameters.

Test	Parameters				Results			
	γ_e	γ_v	γ_d	γ_a	t(ms)	$f(M_w, G_{human})$	Solutions	Explored
First run	1.0	0.0	0.0	0.0	3400	1.0	48	823542
Second run	1.0	1.0	0.0	0.0	3600	1.0	48	823542
Third run	1.0	1.0	1.0	0.0	3900	0.948	1	823542
Fourth run	1.0	1.0	1.0	1.0	4010	0.953	1	823542

must be included in the fitness functions in order to help disambiguate the best solution. This happens because most places are characterized by centrally connecting to others, which results in two star shaped graphs, creating ambiguities for the best solution. For example, when only edges connectivity is considered ($\gamma_e = 1, \gamma_v = 0, \gamma_d = 0, \gamma_a = 0$) or only edges connectivity and number of associated vertices ($\gamma_e = 1, \gamma_v = 1, \gamma_d = 0, \gamma_a = 0$), the algorithm will detect 48 best solutions, which have a perfect match $f(M_w, G_{human}) = 1$, and it is not possible to disambiguate them. Although from this table it seems that there is no influence when the number of vertices associated is considered, that is not true. Looking in more detail, when all results obtained from the fitness function are discretized for all possible combinations into 1000 slices, from 0 to 1 and then the number of solutions for each one of these slices is plotted into a graph, figure 5.12, there is a slight change in both datasets (yellow and green) in the worst set of matching solutions. This can be an important clue for tree search approaches.

When the orientation attributes from the topological map obtained from the human description are considered, the number of best matching solutions decreases to 1 and the quantification of best matches also decreases to nearly 0.95. Two facts are worth highlighting. Firstly, as expected, these attributes help disambiguate best matches for the pure form of a graph (without attributes). Secondly, when a topological map obtained from a human description is used along with the attributes described in this matching, this will lead to imperfect perfect match for the best solution found, also as expected. Another interesting fact was that the best matching quantification of the fourth run was higher than the best matching quantification of the third run. This happens because the human description described the area more precisely for the places.

By analyzing figure 5.12 more closely, it is possible to conclude that introducing more details into the fitness function will result in less matches with the same value for matching quantification or ambiguities. This does not lead to less local maximums for the best matching search algorithm, but to a better differentiation between matching solutions. The *TopoMergExplorer* is a robust approach to find the best matching solution. However, it is not optimized to eliminate uninteresting matching solutions from the search step, which means that it takes longer to complete matching task. In this simulation performed with a laptop with an Intel Dual Core 2.4GHz processor and Ubuntu, the algorithm took between 3.4 seconds to 4 seconds to find the best matching for two graphs. This can be considered a very bad performance in a context of human-robot interaction. These times were found by running each test 10 times and then calculating the average for these execution times. The next subsection presents another algorithm based on the tree search approach.

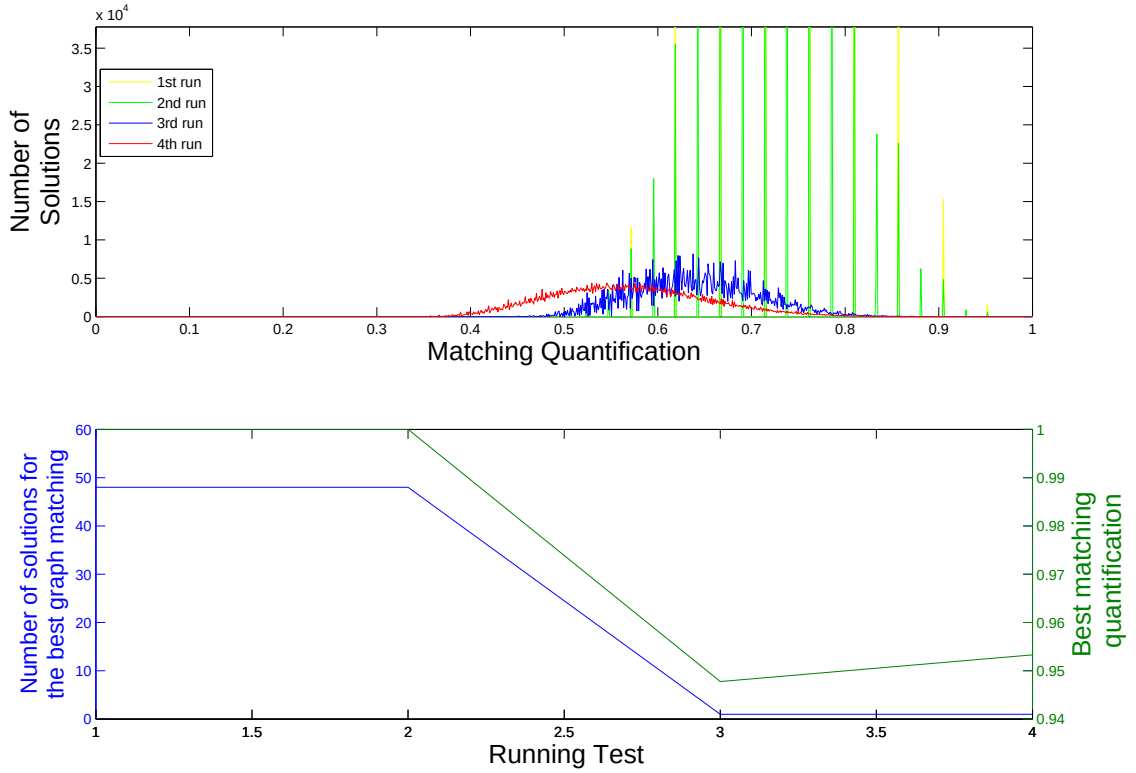

Figure 5.12: The results obtained from the fitness function, for all possible combinations, discretized into 1000 slices from 0 to 1

5.3.3 The TopoMerg algorithm

There are two ways of solving the problem, when a graph matching process is required; trying all matching combinations, as done by the *TopoMergExplorer*, or using optimized search techniques, described in section 5.2.3. The first option has the advantage of selecting the best or the set of best solutions (the global maximums). However, it is less efficient in terms of search time because it will look into all possible combinations for the value of the matching quantifier. If the same computer from the previous section is considered to estimate the time that *TopoMergExplorer* takes to process a more complex problem, it is found that the time required for this approach is given by the next equation:

$$t = (|V_G| + k_s)^{(|V_M|-1)} * \frac{4010}{823542} ms \tag{5.9}$$

Where $\frac{4010}{823542}$ represents the time taken by the algorithm to conduct the test in each combination possible, obtained from previous tests. By using small graph matching problem with two graphs with 9 vertices, each one will lead to a time processing of 209.603 milliseconds, almost

3.5 minutes. This is a considerable amount of time to process a graph matching solution within a human-robot interaction. Therefore, other strategies should be used to reduce the amount of time required to find the best solution. Considering the state of the art described in section 5.2.3, the tree search concept with backtracking was selected in order to develop an algorithm for the graph matching problem in the topological layer of the HySeLAM framework. The tree search was selected due to its efficiency and easy adaptation to attributed graphs.

TopoMerg is the name of the proposed algorithm. The *TopoMerg* algorithm is based on the tree search concept and searches for the best matching solution for the two graphs G_{human} and M_t, by considering the constraints, attributes of the edges and vertices of graphs. This algorithm uses the same matrices applied by the *TopoMergExplorer*, such as H_{match}, E_{match} and $E_{match}^{orientation}$ and the intermediate graph M_w.

The algorithm starts by constructing the solution with the constraints imposed earlier, which will have $|V_G|$ of states. This means that the algorithm will try matching each vertex of the human description G_{human} to a set of vertices of the topological map M_t which maximizes the fitness function $f(M_w, G_{human})$ and minimizes the error returned by the function $ErrorN_{edges}(i)$. The $ErrorN_{edges}(i)$ function returns the difference between the number of edges in vertex i of G_{human} and the number of edges in the set of vertices of M_t (only edges that connect to other vertices outside of the set).

The algorithm is described by the HySeLAM-FMAC First Maps matching Algorithm pseudo-code shown in algorithm 3. The algorithm has two core functions: $Graph_{AddVertice}(G_{human}, H_{match}, M_t, i)$ and $Graph_{explorer}(G_{human}, V_{Gvisited})$.

The function $Graph_{AddVertice}(G_{human}, H_{match}, M_t, i)$ adds a vertex from the topological M_t to the set of vertices w_M^i which is associated to the vertex i of G_{human}. The set of vertices w_M^i linked to the vertex i of G_{human} is reflected by the H_{match} matrix. The vertex is added to the set according to the following rules:

- if the set w_M^i is empty, the function will find the vertex from M_t which maximizes the output of the fitness function $f(M_w, G_{human})$ (the matching quantification). If there is more than one vertex with best matching quantification, it will choose the first option and the function will return H_{match} reflecting this option, and the other options will be added to the $H_{match}^{backtracking}$, for future revisiting step.

- if the set w_M^i is not empty, the function will find the vertices that are connected to the set of vertices w_M^i and choose from these vertices the vertex that obtains the lowest value from $ErrorN_{edges}(i)$ if there is more than one option the vertex that maximizes the output of the fitness function will be chosen. If there is more than one vertex with the best matching quantification, the first option will be chosen and the function will return H_{match} reflecting this option, the other options will be added to the $H_{match}^{backtracking}$, for future revisiting step.

The function $Graph_{explorer}$ returns an index of a vertex from G_{human} which has not been visited yet. The index is selected based on the following rule: vertex with more defined attributives and with at least one edge connecting to the visited vertices. If there is more than one vertex in these

Algorithm 3 HySeLAM-FMAC First Maps matching Algorithm with the Constraints

$H_{match}^{State} \leftarrow initBy(G_{human}, M_t)$
$E_{match} \leftarrow initBy(G_{human})$
$E_{match}^{orientation} \leftarrow initBy(G_{human})$
$\lambda \leftarrow 0$
$i \leftarrow updateBy(G_{human})$
$V_{Gvisited} \leftarrow \varnothing$
$H_{match}^{backtracking} \leftarrow \varnothing$
repeat
  $H_{match} \leftarrow updateBy\left(H_{match}^{State}\right)$
  repeat
    $V_{Gvisited} \leftarrow V_{Gvisited} \cap \{i\}$
    repeat
      $\left[H_{match}, H_{match}^{backtracking}\right] \leftarrow Graph_{AddVertice}(G_{human}, H_{match}, M_t, i)$
      $M_w \leftarrow updateBy(H_{match}, M_t)$
      $E_{match} \leftarrow updateBy(M_w)$
      $E_{match}^{orientation} \leftarrow updateBy(M_w, M_t)$
      $\lambda \leftarrow f(M_w, G_{human})$
    until $ErrorN_{edges}(i) \neq 1$ OR $Ah_n(i) < (1 - \varepsilon_A)$ OR H_{match}
    $i \leftarrow Graph_{explorer}(G_{human}, V_{Gvisited})$
  until $|V_{Gvisited}| = |V_G|$
  if $\lambda_{best} < \lambda$ **then**
    $\lambda_{best} \leftarrow \lambda(H_{match}^{State})$
    $H_{match}^{best} \leftarrow H_{match}^{State}$
  end if
  $H_{match}^{State} \leftarrow updateBy\left(H_{match}^{backtracking}\right)$
until $|H_{match}^{backtracking}| = 0$

conditions, the vertex with more edges connecting to the visited vertices and with more edge attributes filled will be chosen.

Algorithm 3 in the first round will visit all vertices of G_{human} and add the best option when there is a unique solution; when there is more than one option, the first option will be selected. After this first round, the algorithm will visit the other options and start creating a new solution from there. This algorithm will save the matching solution with the best matching quantification.

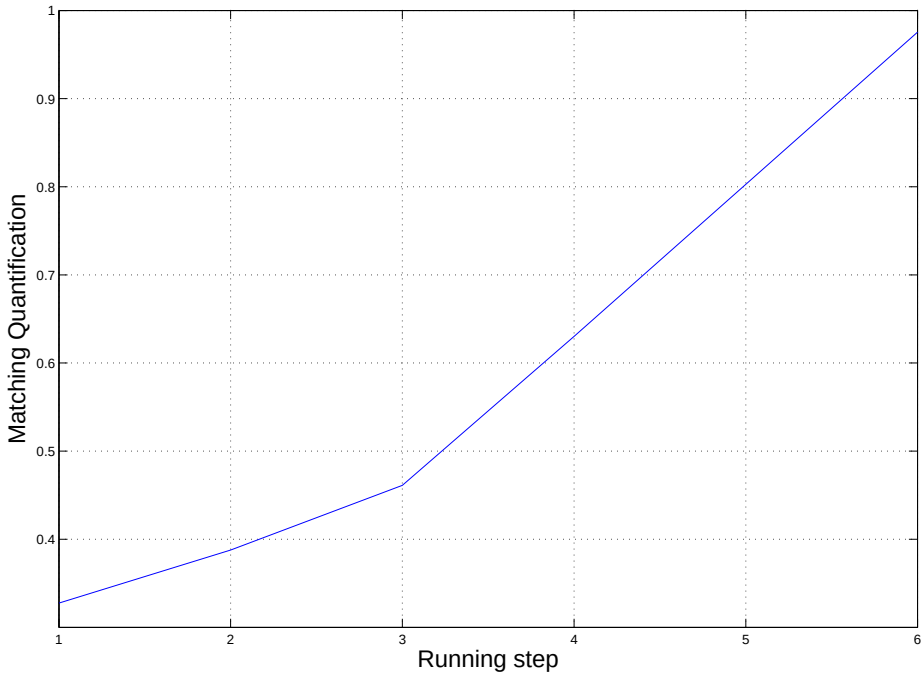

Figure 5.13: The evolution of matching quantification during construction of the solution, by the *TopoMerg* algorithm.

The matching problem described in subsection 5.3.1 was used in order to compare the performance of the *TopoMerg* algorithm to the *TopoMergExplorer* algorithm. For this problem and using the same computer the algorithm took 380 μs to find the same solution as the *TopoMergExplorer*, figure 5.11. The *TopoMerg* algorithm was 10000 times faster than the *TopoMergExplorer* which is an acceptable amount of time to find the solution. Figure 5.13 shows the evolution of the matching quantification while constructing the solution.

Besides finding the best graph matching solution, the *TopoMerg* will hold this solution until it is verified in a human-robot interaction. The verification process will be performed in each place where there is a human available to interact with the robot. In these conditions, the robot will ask the human whether the associated name (through graph matching) is the correct one for that place. If that name is confirmed by the human, it will be added to the vertex of the topological

map. If this is not the correct word, the robot will ask for the name of the place and rerun the *TopoMerg* with this new constraint. If the unverified name of the place is required to complete a task described by a human, the task planner module (from the HySeLAM framework) will receive the vertex associated with that human word and will be notified about the uncertainty related to that human word.

5.4 Conclusions and Future Directions

With the HySeLAM-TopoAskPlace procedure, described by algorithm HySeLAM-TopoAskPlace, it is possible for the robot to acquire the correct name of a place. However, from a human viewpoint, this procedure is not the most efficient because it requires the human to follow the robot through all segmented places. In order to simplify this semantic mapping, the question *how to simplify this procedure* emerged. One way of doing that is when a human is available to interact with the robot, and the robot asks for a description of the place. From here, two question have arisen:

- *How is it possible to translate the description of the place given by a human into an augmented topological map?*
- *How can these two augmented topological maps be merged to form a single map?* .

A sequence of two operations was developed in order to answer these questions and to make it possible for the robot to infer the names of places from a human description. The first operation to be explored answers the first question, which was the translation from a textual description into a topological map. An approach based on the Nooj tool was developed for this operation. This approach was efficient during all the tests where a well structured textual description was provided. Although this approach requires a well structured textual description, this requirement can be relaxed if other procedures closer to human interaction are used. These procedures closer to human interaction can translate informal texts into formal texts.

The second operation to be explored answers the second question, in which the human description is fused into the topological map. It was found that this problem can be described as a graph matching problem. For the graph matching problem, several approaches were studied but due to the performance and the easy-to-use attributed graphs, the tree search concept was chosen to develop our approach, which is called *TopoMerge*. Before developing this approach, a study was conducted on how to measure the quality of the solution found for the fusion. A fitness function was proposed and tested using the *TopoMergeExplorer*. The *TopoMergeExplorer* was another approach developed to find the best matching; however, this algorithm will calculate the matching quantification, using the fitness function in all possible combinations. These combinations follow the rule of graph matching in a one-to-many way. Although as expected the *TopoMergeExplorer* is not efficient in terms of processing time, it is an important tool for the process of validating the fitness function and also for studying the impact of including each attribute in the fitness function.

Also as expected, more attributes can help differentiate the best solution for the matching problem. In this work only two attributes were considered: the area of the place and relative orientation between the places. Sometimes the human does not describe the area of the place, which will lead to graph matching without the area attribute. This can be a problem for both methods, but especially for the *TopoMerge*, which will present more ambiguities. It is not a significant problem for the algorithm because it will store these ambiguities, and after one of the options is tested, the other hypotheses will be tested. However, in the end this could lead to multiple solutions with the same matching quantification. One way to solve this in the future is including another procedure in the *TopoMerge* while creating the possible solution. For example, when one ambiguity appears, the robot can ask the human for other attributes for the vertex of the topological map described. Only two attributes were considered in this work. Nevertheless, other attributes can be used, such as the dominant color of a place, the shape of the place or the height (all information that can be inferred by the robot). For example, when the robot stores places with dominant colors and there are ambiguities to solve the vertices (places), the robot can ask if the dominant color of the places described is the same. When a semantic map (objects) is completed or filled, the objects related to a place can also be used as attributes in order to help the *TopoMerge* to find the correct solution; for example, specific objects can be related to specific place for the places. Even with fewer attributes this approach was able to find the correct answer for the tested scenarios.

Another important contribution for future research is the inclusion of the uncertainty associated with human descriptions and answers. Sometimes, the human can supply incorrect or incomplete descriptions and answers, which will lead to wrong solutions. This should also be considered in the future.

Chapter 6

Towards semantic Localization and mapping based on visual signatures

Recognizing a place at a glance is a useful human capacity. It allows humans to know their location in the world and to plan their tasks/activities according to that knowledge. This is an important skill that should be added to mobile robot systems in order for them to gain the ability to locate themselves and perform context interpretation. However, in robotics systems, localization is usually based on a purely geometric model. In the robotics research community we believe that the use of vision and place recognition opens up a number of opportunities in terms of flexibility and the association of semantics to the model.

Moreover, attention should be paid to the fact that robots working and cooperating in public places or at our homes or offices will require redundancy for the most important modules for their use to be safer. One of that modules is the SLAM module, which can supply accurate robot localization on the acquired occupancy or features based map. However, situations such as robot kidnapping or malfunction in the SLAM module can occur in any robot, and these may force it to execute behaviors that are dangerous, for itself and for humans.

The HySeLaM framework formalizes an augmented topological map which includes visual signatures such as propriety of the place that can be used for visual place recognition and localization. This can be useful in situations in which the SLAM approach fails, kidnapping occurs, as well as for fast SLAM recovery or even use by a supervisor of the SLAM (to detect malfunctions, which can be done by the topological engine). The questions that drove the work presented in this section of the thesis were the following.

1. How must these visual signatures be constructed to allow for robust visual place recognition?

2. How can the augmented topological map be included in the semantic localization procedure for increased place recognition accuracy?

6.1 Global overview

Place recognition through visual images is a simple task for humans. However, the process through which the brain performs it is unknown. It is a complex task and a vastly researched topic in the robotics and computer vision communities. This task is complex because the information provided by the image sensor is not directly usable and usually requires at least two complex stages: information extraction and information classification. Despite the numerous approaches to these two stages, there are no systems that possess the accuracy of the human system working under the normal changes in illumination, perspective and scenario.

Information extraction or image digestion (in content-based image retrieval (CBIR)) is the stage which extracts primitive features from the image. This stage describes an image using low-level features such as color, shape, and texture, while removing unimportant details. Color histograms, color moments, dominant color, scalable color, shape contour, shape region, homogeneous texture, texture browsing, and edge histogram are some of the popular descriptors in use. An overview of the process of extracting from images some of these low-level features and descriptors is described in chapter 2.2, while the main techniques for this extraction can be classified as global or local, and can be itemized as follows.

- Local features and descriptors:
  - Scale Invariant Feature Transform (SIFT), by Lowe (2004);
  - PCA-SIFT by Ke and Sukthankar (2004);
  - Maximally stable extremal regions (MSER), by Matas et al. (2004);
  - Gradient Location and Orientation Histogram (GLOH), by Mikolajczyk and Schmid (2005);
  - Speeded-Up Robust Features (SURF), by Bay et al. (2006);
  - Local Energy based Shape Histogram (LESH), by Sarfraz and Hellwich (2008);
  - Binary robust invariant scalable keypoints (BRISK) descriptors, by Leutenegger et al. (2011);
  - Weighted Histograms of Gradient Orientation (WHGO), by Zhou et al. (2011);
  - ORB based on BRIEF, by Rublee and Rabaud (2011);
  - Fast retina keypoint (FREAK), descriptor only by Alahi et al. (2012);
- Global features and descriptors:
  - GIST, by Oliva and Torralba (2001);
  - Composed Receptive Field Histograms (CRFH), by Linde and Lindeberg (2004b);
  - PCA of Census Transform Histograms (PACT), by Wu and Rehg (2008);
  - CENsus TRansform hISTogram (CENTRIST), by Wu and Rehg (2010);

  - Pyramid of Histograms of Orientation Gradients (PHOG), by Bosch et al. (2007);
  - bag-of-visterms by Quelhas and Monay (2005), this technique is also named as bag-of-words or bag-of-features (for example, BOW-SIFT when this technique is applied with the SIFT descriptor);

Most of these earliest and most widely used techniques for extracting local/global features/descriptors work mainly with grayscale image. In order to take advantage of all the information that color cameras are able to provide, some researchers proposed extensions for the original algorithms. For example, Ancuti and Bekaert (2007) proposed an extension to the SIFT descriptor (SIFT-CCH) that combines the SIFT approach with the color co-occurrence histograms (CCH), proposed by Chang and Krumm (1999) in the normalized RGB color space. This approach performs the same detection step of SIFT, but introduces one dimension to the descriptor. Thus, features are described by a two element vector that combines the SIFT and the CCH descriptor vectors. The main problem of such an approach is the increase in the computational effort during the feature matching due to the extra 128 elements added to the descriptor vector. More efficient approaches can be found in literature:

- Hue-SIFT, proposed by Weijer et al. (2006), computes the SIFT descriptor in H channel of HSV color space and uses the Harris operator as the feature detector;
- CSIFT, proposed by Abdel-Hakim and Farag (2006), uses the H component from HSV space for feature detection and description;
- HSV-SIFT, proposed by Bosch et al. (2007), computes the SIFT descriptor over all three channels of the HSV color model;
- C-SIFT, proposed byBurghouts and Geusebroek (2009), computes the SIFT descriptor over the O_1 and O_2 channels from the normalized opponent color space.
- color-SURF, proposed by Fan et al. (2009), is similar to the SIFT-CCH but the descriptor is computed over all three channels of the YUV color space;
- OpponentSIFT, described by van de Sande et al. (2010), processes the SIFT descriptor over all the channels of the opponent color space;
- rgSIFT, described by van de Sande et al. (2010), computes the SIFT descriptor in the R and G components of normalized RGB color space;
- RGB-SIFT, described by van de Sande et al. (2010), computes the SIFT descriptor for every RGB channel independently;
- Trasformed color SIFT, described by van de Sande et al. (2010), is similar to RGB-SIFT but uses normalized RGB color space;

Most of these approaches were tested by van de Sande et al. (2010) under different datasets. It was concluded that their performance changes from dataset to dataset and, therefore, no approach is best under all datasets.

Correct image classification requires high repeatability of the information extracted from the image under different illuminations and perspectives. Most of these techniques for extracting local/global features/descriptors are stable under image rotation and scale but they are not able to cope with large photometric variations. Photometric variations imply differences in color measurements made by the camera, caused by: shading, shadows, specularities, interreflections, and variation in the intensity or color of the illumination. The ability to perceive color as constant under changing conditions of illumination is known as color constancy, and it is a natural ability of human observers.Zeki (1993) and Horn (1986) have found evidences that prove the existence of color constant cells inside the visual area V4 of the human extrastriate visual cortex. Indeed, Ebner (2007a) states that these cells seem to respond to the reflectance of an object irrespective of the wavelength composition of the light it reflects. Ebner (2012) says that although the mechanism used by the brain to achieve color constancy is not yet well understood, the brain somehow does arrive at a descriptor which is independent of the illuminant.

Figure 6.1: Chang and Pei (2013) suggest a new algorithm to achieve color constancy. From left to right: the original input images (with different illuminants); after the results obtained from Max-RGB (proposed by Land and McCann (1971)); GGW, based on the gray world hypothesis (proposed by Van De Weijer et al. (2007)); GE1, based on the gray edge hypothesis (also proposed by Van De Weijer et al. (2007)); the local space average color (LSAC) (proposed by Ebner (2009)); and the chromaticity neutralization process (proposed by Chang and Pei (2013)).

Before information extraction or image digestion, a process to achieve color constancy on the image can be an important step, as proven by Petry (2013) and Petry et al. (2013). Indeed, in scenes with large photometric variations, Petry et al. (2013) show that using a color constancy algorithm before the SURF algorithm for feature extracting can help improve the correct feature association, between two images, by 41.69%. A considerable number of color constancy algorithms are proposed in the literature, most of which were reviewed and tested by Ebner (2007b), Hordley (2006), Gijsenij et al. (2011) and Gijsenij et al. (2012). Figure 6.1 displays the results obtained from five algorithms that reduce the influence of illuminant colors. From the literature

review, the local space average color (LSAC) algorithm (by Ebner (2009)) seems to be the most simple and efficient algorithm in terms of computational cost and color constancy.

After color constancy processing, a more stable image description can be achieved using a global descriptor or multiple local descriptors (like SIFT or SURF, among others). However, image classification based on local descriptors is a complex task. In order to simplify the use of local descriptors, the concept of the bag-of-words model, used in natural language processing and information retrieval (IR), has been adapted for use in image classification.

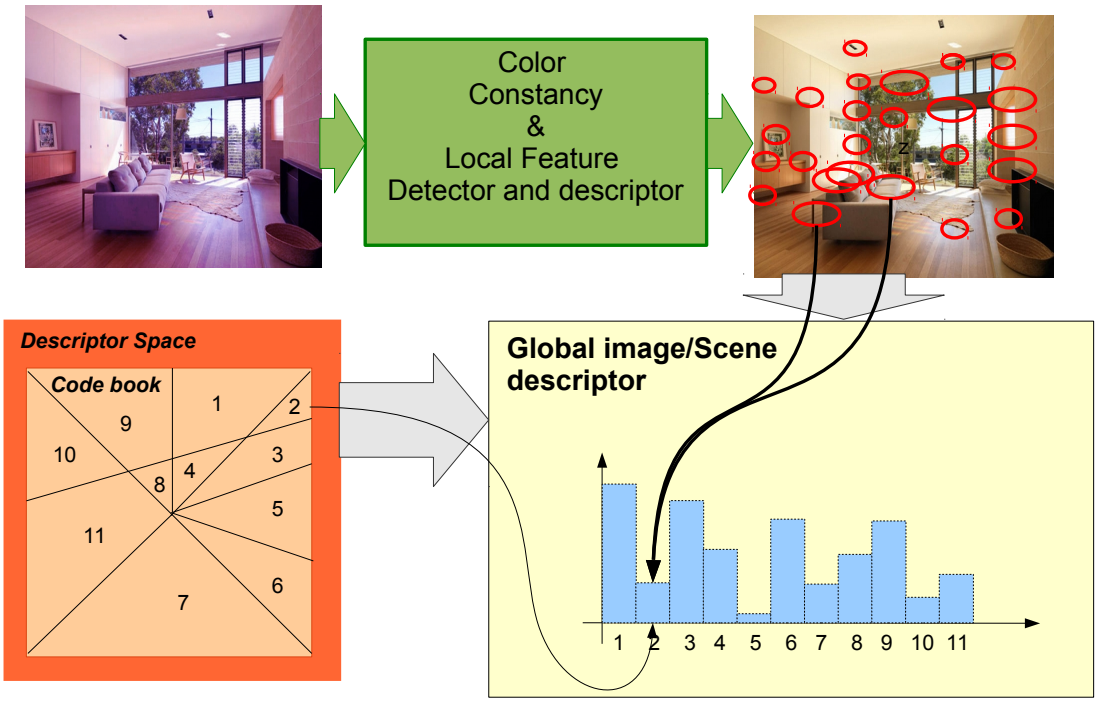

Figure 6.2: An example of the global image/scene descriptor extractor using the concepts of “bag-of-features” and “codebook“. The global descriptor can be defined as a histogram of occurrences of the discretized descriptor space.

In the bag-of-words model, a text (like a sentence or a document) is represented as an unordered collection of words, disregarding grammar and even word order. Usually a statistical process is used, like a histogram with the number of occurrences of each word in the text. On image processing, this concept is commonly known as the ”bag-of-features“ or ”bag-of-visterms“. To make the descriptor usable in this ”bag-of-features“ model, the descriptor space is usually discretized into n clusters, and these n clusters form a ”codebook“, as depicted in figure 6.2. The influence of the size of these “codebooks” and techniques for clustering is explored by Nowak et al. (2006) and Jurie and Triggs (2005). The k-means, K-medoids and clustering expectation–maximization (EM) are most used techniques.

This analogy between visual-words and text-words was explored by Sivic and Zisserman (2003). Their work has explored both concepts used in text retrieval and concepts used by Google

search engine in image retrieval from databases and videos. Two techniques were used for the feature extraction: shape-adapted regions (SA), proposed by Mikolajczyk and Schmid (2002), and maximally stable extremal regions (MSER), proposed by Matas et al. (2004). These two types of feature extraction were employed because they detect different image areas and thus provide complementary representations of a frame/image. The SA regions tend to be centered on corner like features, and the MSER regions correspond to blobs of high contrast with respect to their surroundings, like a dark window on a gray wall. Both types of regions are represented by ellipses. These are computed at twice the originally detected region size in order for the image appearance to be more discriminating. After the feature detection, the feature descriptor is obtained using the SIFT descriptor technique. Two ”codebooks“ are used, one for the SA features and another for the MSER features. These ”codebooks“ are constructed using the K-means clustering technique. The image description is given by weighed histogram which contains the number of occurrences of each descriptor cluster in the scene/image. In order to optimize the final result, this weighed histogram has different weight for each descriptor cluster; the cluster with more occurrences in all images has less weight because it is less descriptive.

Describing an image/scene simply through the number of occurrences of each descriptor clustered, the simple feature histogram (frequency histogram), does not include information about the order of occurrence. Although the simple feature histogram works well for image/scene classification when the group of classes is small and has a very distinctive set of features, it does not perform well when there are multiple classes with similar sets of features. In these cases, a description encompassing the spatial organization of the features can increase the performance of the classifier. Lazebnik et al. (2006) present a method for recognizing scene categories based on approximate global geometric correspondence. This technique works by partitioning the image into increasingly fine sub-regions and computing histograms of local features found inside each sub-region, as depicted in figure 6.3. Their work shows that the resulting ”spatial pyramid“ is a simple and computationally efficient extension of an orderless bag-of-features image representation, and it shows that this approach can significantly improve the performance of scene categorization tasks.

The spatial pyramid approach has been successfully applied by other researchers, among which Battiato et al. (2009) and Bosch et al. (2008), in scene classification. Zhou et al. (2012) claim to have achieved an improvement of this approach by recurring to multiple image resolutions and to horizontal and vertical image partitions. Their scene classification approach is based on the incorporation of a multi-resolution representation into a bag-of-features model. In the proposed approach, multiple resolution images are constructed and features extracted from all the resolution images with dense regions. These extracted features are quantized into a visual ”codebook“ using the k-means clustering method. To incorporate spatial information, two modalities of horizontal and vertical partitions are adopted to partition all resolution images into sub-regions with different scales. Each sub-region is then represented as a histogram of codeword occurrences by mapping the local features to the ”codebook“. The proposed approach was evaluated over five commonly used data sets including indoor scenes, outdoor scenes, and sports events. From the experimental results it is possible to conclude that this approach outperforms the previous meth-

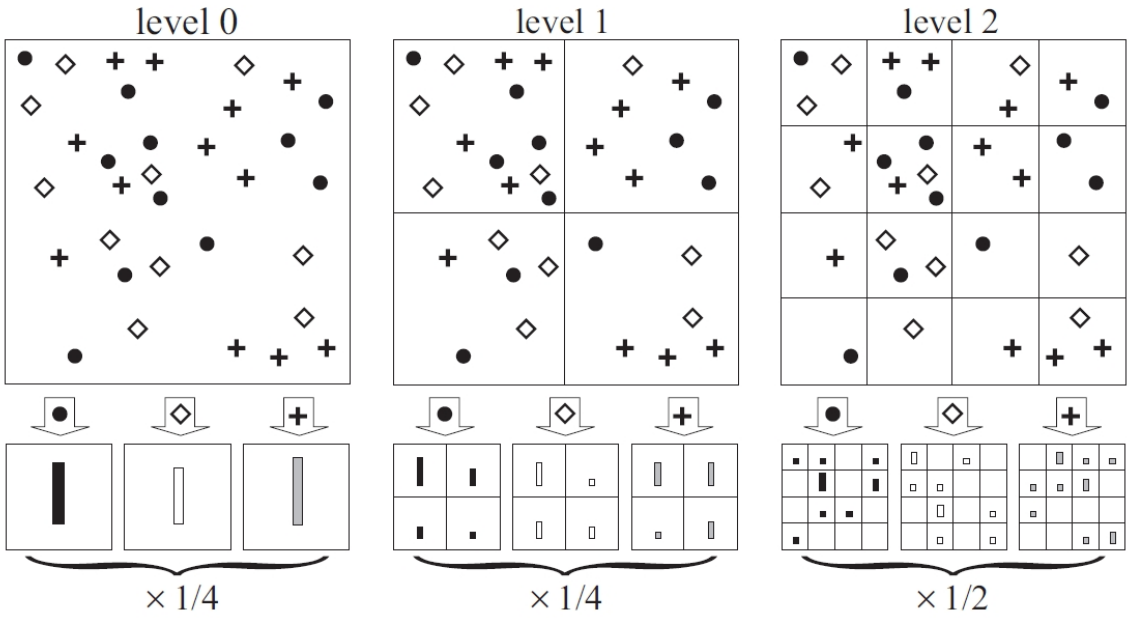

Figure 6.3: Lazebnik et al. (2006) give an example of constructing a three-level pyramid. The image has three feature types, indicated by circles, diamonds, and crosses. At the top, Lazebnik et al. (2006) subdivide the image at three different levels of resolution. Next, for each level of resolution and each channel, they count the features that fall into each spatial bin.

ods, and it is shown that changing from the SIFT to the WHGO descriptor has no influence on the performance.

Biederman (1988) has shown that humans can recognize scenes by considering them in a "holistic" manner, without recognizing individual objects. This perspective has helped other researchers to construct other alternatives to describe an image globally without using local features/descriptors. Therefore, drawing inspiration from the perceptual literature, Oliva and Torralba (2001) have proposed the gist Descriptor, a low dimensional representation of scenes, based on several global properties such as "naturalness", "openness", "roughness", "expansion", and "ruggedness". Torralba and Murphy (2003), Oliva and Torralba (2006) and Murphy et al. (2006) have shown the power of this descriptor on the context-based vision system for place and object recognition. Although a system using only the gist descriptor works quite well on outdoor images, it does not get the same performance in indoor scenarios, due to the weak invariance to image rotation of gist descriptor.

Hu and Guo (2012) and Meng and Wang (2010) propose alternatives to describe a scene that do not use local feature detectors. Hu and Guo (2012) state that although the Local Binary Patterns (LBP) is an effective and efficient texture descriptor, the conventional LBP histogram ignores spatial information of the descriptor present on the image. Thus, a new Spatial Local Binary Patterns (SLBP) approach to describe an image was proposed. This SLBP approach is based on LBP and spatial linear and circular histogram. These spatial histograms are proposed by Cao et al. (2010) and they are depicted in figure 6.4. Hu and Guo (2012) compare the performance of

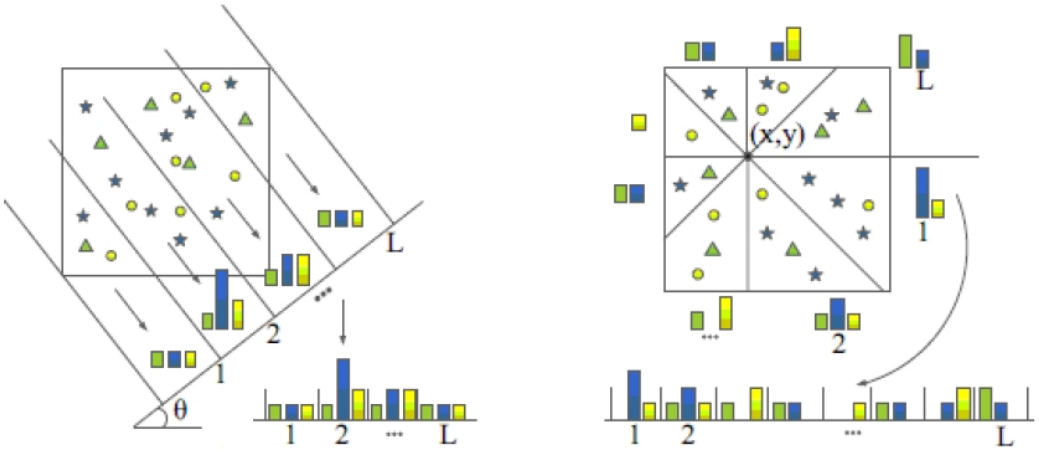

Figure 6.4: Cao et al. (2010) propose two spatial bag-of-features approaches. This image gives an example for the construction of linear and circular ordered bag-of-features. Stars, triangles, and circles represent three kinds of features. (a) Linear projection: all features are projected onto a line with angle θ and resolution $L = 4$, and then the features within each spatial bin are counted. (b) Circular projection: we locate the center at $(x; y)$ and evenly divide the space into $L = 8$ sectors, and then count features within each sector.

SLBP using the linear and circular spatial histogram approach. The obtained results show a big improvement of the SLBP performance against the conventional LBP histogram performance, and an equal performance when SLBP uses the linear or the circular spatial histogram approach. This SLBP approach has the advantage of being computationally efficient due its simplicity and more descriptive because it includes some spatial arrangement of the descriptors.

Meng and Wang (2010) propose a similar approach that tries to eliminate noise sensitivity from local binary pattern descriptors. The spatial pyramid splitting approach, proposed by Lazebnik et al. (2006), is used rather than the linear or circular spatial histogram approach, proposed by Cao et al. (2010). In order to remove noise sensitivity from the local binary pattern, a threshold is given to the central pixel, for the comparison of its value to the neighboring pixels value. A tristate value is obtained for this comparison instead of a binary value, where 0 means similar pixel values, -1 lower pixel value and 1 higher pixel value. The result is a vector of these tristate values, which is then converted into two binary codes: the upper pattern and the lower pattern. These two binary codes are then used to construct the spatial pyramid of the local binary patterns. The conclusion was that, in terms of average accuracy rate, the proposed method outperforms Spatial Envelope (by Torralba and Murphy (2003)) and spatial PACT (by Wu and Rehg (2008)) by 4.0% and 1.9% respectively.

With the information extraction stage, the vector image/scene description, also known as visual signature, is obtained. This visual signature is required for the image/scene classification stage. A classifier must be chosen for this stage that is capable of learning from a set of labeled visual signatures and also of classifying other visual signatures into the corresponding classes according to the learned model. For this stage, a machine learning technique is required. In literature several

works can be found that use at least one of these techniques:

- Neural Networks (NN);
- Support Vector Machine (SVM);
- Adaboost;
- Fuzzy systems;
- Genetic Algorithms;
- Bayes classifiers;

The Neural Networks and Support Vector Machine based approaches are the most commonly used techniques, due to their demonstrated performance. However, the use of these two classifiers requires some empirical experience to extract the best performance from the system. Over-fitting and nonlinear separation between classes are the two most common problems to influence the parameter choice of NN and SVM. For NN, these parameters include the choice of the number of hidden layers and the number of neurons. For SVM, the kernel type, such as Fisher kernel, Graph kernel, Polynomial kernel, RBF kernel and String kernels, must be selected. The parameter selection depends largely on the input information and the number of classes, and there is not a formal method for its configuration.

The local feature detectors, local and global descriptors, color constancy algorithms and classifiers are the basic techniques required for place/scenario classification. Looking into more detail into the state of the art related to scene classification, Rasiwasia and Vasconcelos (2008) create an intermediate space between information extraction (image description) and classification so as to improve the classification performance. This intermediate space is based on a low dimensional semantic ”theme“ image representation. Each ”theme“ induces a probability density on the space of low-level features, and images are represented as vectors of posterior ”theme“ probabilities. This enables an image to be associated with multiple themes, even when there are no multiple associations in the training labels. This vector of theme associations is denoted as a semantic multinomial (SMN) distribution and it is used as the input of the final classifier.

Rasiwasia and Vasconcelos (2012) revisit this approach and improve it by including one more layer of semantic representation to reduce negative effect from co-occurrences between concepts (scenarios), which can induce the system into a wrong classification. The basic idea behind this new layer is that, while images from the same concept (scenarios) are expected to exhibit similar contextual co-occurrences, this is not likely for ambiguity co-occurrences. For example, although one ”street scene“ could contain some patches that could also be attributed to the ”bedroom” concept, it is unlikely that this will hold for most images of street scenes. By definition, ambiguity co-occurrences are accidental; otherwise they would reflect common semantics of the two concepts and would be contextual co-occurrences. Thus, while impossible to detect from a single image, stable contextual co-occurrences should be detectable by joint inspection of all SMN derived from

the images of a concept. These ideas were the basis for the development of a new approach using a contextual model. This approach was compared against other approaches and it was shown that it outperforms the previous ones and all other tested approaches, such as those based on probabilistic latent semantic analysis (pLSA), Bag-Of-Features and bag-of-concepts using the SVM as the classifier. Although this new approach outperforms all the others, the approach proposed by Lazebnik et al. (2006), which is based on pyramidal spatial Bag-Of-Features and SVM, has a very close performance.

In the context of mobile robots for outdoor scenarios and SLAM improvement, Cummins and Newman (2010) propose an approach that allows a robot to identify when it is revisiting a previously seen location, using only the imagery captured by its camera as input. On the basis of their work is the FAB-MAP, which was previously proposed by Cummins and Newman (2008). The FAB-MAP uses the bag-of-words (using the SURF feature detector and descriptor) in a similar way as Sivic and Zisserman (2003). This FAB-MAP approach also uses the standard frequency–inverse document frequency (tf-idf) relevance measure, which was borrowed from text retrieval. Cummins and Newman (2010) argue that tf-idf ranking is not particularly suitable for image data, so they have outlined a complete probabilistic framework for the place recognition task, which is applicable even in visually repetitive environments where many locations may appear identical. They demonstrate that place recognition performance can be improved by learning an approximation to the joint distribution over visual elements. Their improved FAB-MAP approach was successfully used with a metric SLAM approach so as to solve the loop closure detection and multi-session mapping problems, in the outdoor scenario.

In the more complex context of mobile robots for a mixed indoor/outdoor scenarios, Quelhas and Monay (2005) suggest an approach to model visual scenes using local invariant features and probabilistic latent space models. Their work also explores three open questions:

1. whether the invariant local features are suitable for scene (rather than object) classification;
2. whether unsupervised latent space models can be used for feature extraction in the classification task; and
3. whether the latent space formulation can discover visual co-occurrence patterns, motivating novel approaches for image organization and segmentation.

From their extensive experiments on binary and multi-class scene classification tasks, they have conclude that a Bag-Of-Features representation, derived from local invariant descriptors, consistently outperforms previous state-of-the-art approaches. Also, they have concluded that probabilistic latent semantic analysis (pLSA) generates a compact scene representation, discriminative for accurate classification, and significantly more robust when less training data are available.

pLSA is another concept extracted from text retrieval and it is a probabilistic extension of latent semantic analysis (LSA). LSA has basically the following goal: given a corpus of K documents, comprising a dictionary of M words, find the "relations" between words and documents (usually cluster the documents). This corpus of K documents and the occurrence of these M words in

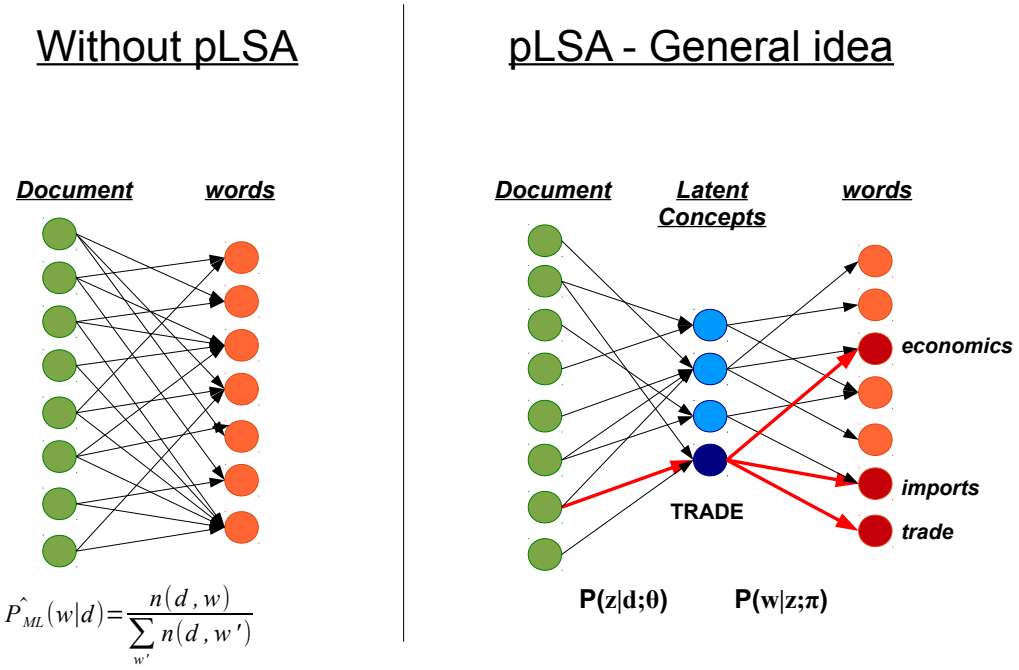

Figure 6.5: In BoW, each document is a bag of words, as it assumes that word order has no significance (the term "car drive" has the same probability as "drive car"). On the other hand, in pLSA, a larger corpus is considered instead of a single document, to infer more meaningful conclusions. In pLSA, the latent concepts (or topics), denoted as Z, act as a bottleneck variable.

each document is stored into a Co-occurrence Matrix, where the matrix element at (i, j) is the word count (or, frequency) of the *ith* word in the *jth* document. The multiplication between two lines ($t_i^T t_p$) gives the correlation between two terms over all documents and multiplication between two columns ($d_i^T d_p$) gives the correlation between all the terms in two documents. After several operations, these properties enable the extraction of an intermediate space, the latent concepts, which is a set of "topics" placed between the document space and the word space. So now it is possible to relate the documents to "topics" and the "topics" to words, as depicted in figure 6.5. On scene classification, the pLSA model can be applied in three steps: creating the visual words set (denoted w), similarly to BoF; learning the topic specific distribution $P(w|z)$ from the training set by fitting the training set into the pLSA model; representing each training image *im* by a K vector of $P(Z|im)$, where $|Z| = K$ is the amount of topics. One disadvantage of the pLSA model is the inability to add new documents to the model without probabilities being recalculated.

Unlike previous works, Wu and Rehg (2008), Wu et al. (2009), Wu and Rehg (2010) and Paris and Gloti (2010) have explored the use of census transform histogram based approaches (also known as local binary pattern histograms) in the context of indoor robotics. They have shown that scene recognition approaches based on Local binary patterns can outperform SIFT and SURF descriptor based approaches. Local binary patterns seems to be the most simple computational

approach to process local and global descriptors, as seen before on this section of the thesis.

Another important issue in scenario classification is the detection of context breakpoints from successive visual images. For example, if a robot is navigating inside of a building it is important for it to know when the context changes. With this in mind, Ranganathan (2012) proposed the Place Labeling through Image Sequence Segmentation (PLISS) approach for detecting and labeling places online. PLISS uses a change-point detection technique to temporally segment image sequences which are subsequently labeled. Change-point detection and labeling are performed inside a systematic probabilistic framework. Unknown place labels are detected by using a probabilistic classifier and keeping track of its label uncertainty. The PILSS was tested by means of a spatial pyramidal Bag-Of-Features. This Bag-Of-Features was tested with three descriptors SIFT, CENTRIST (3x3) and CENTRIST (5x5). The "codebook" was constructed with the k-means technique. The authors concluded that PLISS with the SIFT descriptor outperforms PLISS with CENTRIST.

The key idea of context breakpoints is also used by Angeli et al. (2008) whose proposed approach builds the topological map from scratch, incrementally and with a single monocular wide-angle camera. Their approach relies on the visual bag-of-features paradigm to represent the images and on a discrete Bayes filter to compute the probability of loop-closure. The visual bag of features is constructed by a discretization of the SIFT descriptor space. The system starts building the topological map by adding new nodes when two consecutive images have less than 90% similarity (breakpoints) and if prospective new nodes are not similar to existing ones. Local image similarity is defined between a node N_i and the current image I_t as the percentage of visual features extracted in I_t that are visual words characterizing N_i.

Figure 6.6: Pham et al. (2010) show an example of a visual graph extracted from an image. Concepts are represented by *nodes* and spatial relations are expressed by *directed arcs*. Nodes and links are weighted by the number of times they appear in the image.

For the task of encoding the spatial organization of image features, Pham et al. (2010) suggest

an alternative to the spatial pyramid histogram approach. Their proposed approach exploits an extension of the language modeling (LM) approach, from information retrieval to the problem of graph-based image retrieval and categorization which does not require the use of the SVM or NN approaches. They claim that a graph model is needed to represent the multiple points of views of images. A language model is defined on such graphs to achieve fast graph matching. Briefly put, an image is modeled as a set of visual concepts derived from different visual features. The principles of the bag-of-words model are used for concept extraction. The procedure for the concept extraction follows the next four main steps.

1. Identifying regions within the image that will form the basic blocks for concept identification.

2. Indexing each region with a predefined set of visual features (HSV histogram, Edge histogram, SIFT descriptor).

3. Clustering all the regions found in the collection into K classes, each class representing one concept.

4. Extracting relations between concepts.

Then it is possible to describe each image by a visual graph using these clustered concepts, as shown in figure 6.6. Pham et al. (2010) experimental results, using images obtained from an indoor robot, show that the use of visual graph model improves the accuracies of the results of the standard language model (LM) and outperforms the SVM methods.

Liu et al. (2009) describe a scene recognition method that uses an adaptive descriptor based on color features and geometric information for omnidirectional vision. They claim that their approach enables the robot to automatically add nodes to a topological map and solve the localization problem in real-time. They propose a new descriptor, the Fast Adaptive Color Tags (FACT), which is extracted from the YUV color space. The dimension of the FACT descriptor is adaptive depending on the segmentation result of the panoramic image. The authors claim that their descriptor is invariant to rotation and slight changes of illumination, due to the use of the YUV color space. They have compared the time performance between FACT and SIFT, and it is shown that FACT takes much less time to process and it also has a stable time to process. Albeit the good performance, the method is optimized for use with omnidirectional cameras with a 360 degree perspective, which are not frequently available on robots.

When the concept of the bag-of-words is used to classify an image or a document, several correlated "words" are obtained on the coodebook and this reduces the performance of the classifier. In order to minimize the negative effect of the correlated "words", the principal component analysis (PCA) technique is usually applied to the observations to eliminate these correlated words/variables from the observation vector. PCA is a mathematical procedure that uses orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. However, the majority of

the learning methods for visual scene recognition that compute a space of eigenvectors by PCA traditionally require a batch computation step, in which the only way to update the subspace is to rebuild it from scratch when it comes to new samples. The possibility of using PCA in the online mode can improve the performance of a system that is at the learning stage all its "life", as is the case with robots. Qu et al. (2011) propose an approach to scene recognition based on an online PCA algorithm with adaptive subspace, which allows for complete incremental learning. A different subspace updating strategy was advanced for new samples dependent on the degree of difference between new and learned samples. This can improve adaptability in different situations and also reduce the calculation time and storage space. From their experimental results, it is possible to see that their approach has less accuracy when compared to other approaches using an offline PCA algorithm. However, on their approach, the same mode can be used for the training and learning stages and this helps recognizing unknown scenes and performing online scene accumulation and updating.

In the context of robotic research, there are other key ideas that can be used for place/scene recognition, as, for example: the Bayesian learning approach, proposed by Ramos et al. (2012); the inclusion of feature distance extracted from stereo vision, presented by Cadena et al. (2010); the object centric approach instead of a low-level feature descriptor, such as those proposed by Viswanathan et al. (2011) and by Espinace et al. (2013).

Pronobis et al. (2006) explore and suggest a visual place recognition algorithm as an alternative to the traditional localization systems which perform localization based on a purely geometric model. Besides that, they have explored algorithms that work under realistic conditions for robotics applications, i.e., with varying illumination conditions and over time. Their approach is based on a large margin classifier in combination with a rich global image descriptor. The global image descriptor used is based on the Composed Receptive Field Histograms (CRFH), originally proposed by Linde and Lindeberg (2004b), and the classifier used is the SVM with chi-squared kernel. The learning stage was supervised and in each location several images with different orientations were taken. The approach was evaluated with several different cameras, changes in time-of-day and weather conditions. The results show accuracy between 47% and 97%, depending on the camera and the illumination conditions (sunny, cloudy and night).

Pronobis et al. (2008) revisits the previous work and proposes a new method for integrating multiple cues. This new method improves the Generalized Discriminative Accumulation Scheme (G-DAS), proposed by Pronobis and Caputo (2007), and is called SVM-DAS. Basically, for each cue a large margin classifier is trained which outputs a set of scores indicating the confidence of the decision. These scores are then used as input to a Support Vector Machine, which learns how to weight each cue, for each class, optimally during training. Like in the previous work, the CRFH descriptor is still used as cue input. However, two more cue input channels are included: the SIFT features (as local features) and the laser measurements features. The method employed 4 different single-cue models: CRFH with SVM, SIFT with SVM, and laser range features with both SVM (L-SVM) and AdaBoost (L-AB). From these 4 models, three are chosen for the input on the final classifier, which is based on G-DAS, SVM-DAS (Linear and RBF kernel), as depicted in figure

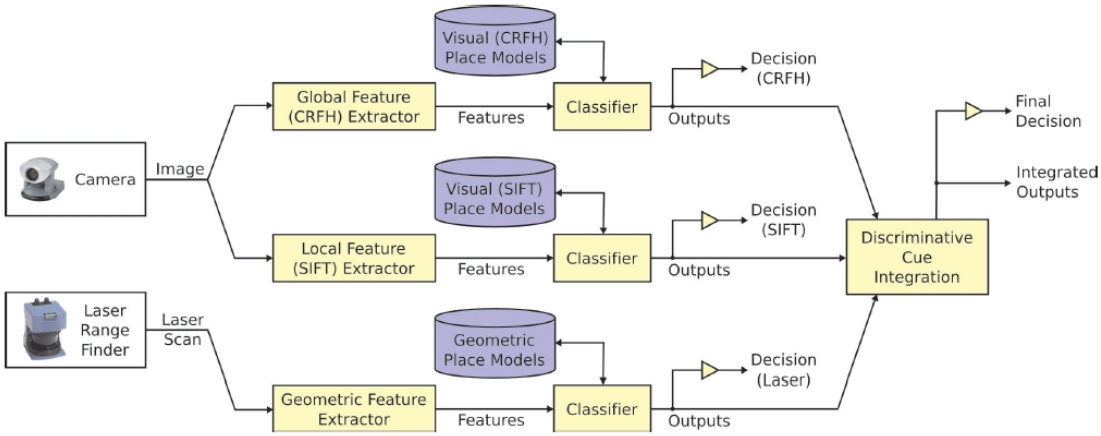

Figure 6.7: Architecture of the multi-modal place classification system, by Pronobis et al. (2009).

6.7. From the experiments, it is possible to conclude that L-SVM outperforms L-AB, and the best option for the multiple cue integration is SVM-DAS, against G-DAS. As for the SVM kernel, the Radial Basis Function (RBF) outperforms the linear. Pronobis et al. (2009) conducted the same test, but this time the SVM kernels tested were histogram intersection (HI), RBF and linear. With the latest experiments, they have concluded that the HI and RBF SVM kernel have a similar performance. A careful analysis of these experiments and results shows that the classifier that uses integration of the 3 information channels (local and global visual features and laser features) significantly outperforms the individual usage of each of the information channels.

6.1.1 Remarks and conclusions

From this global state of the art overview, it is possible to conclude that text retrieval is an important source of concepts for visual place recognition and classification, such as bag-of-words, pLSA, descriptor clusterization among others. The process of recognizing/classifying an image can be divided into two stages: information extraction and classification. The information extraction, also known as image digestion, is the stage which extracts primitive features from the image. This stage describes an image using low level features, such as color shape and texture while removing unimportant details. Color histograms, color moments, dominant color, scalable color, shape contour, shape region, homogeneous texture, texture browsing and edge histogram are some of the popular concepts that are used to build the descriptors. These descriptors can be divided into two classes: Local and Global descriptors.

Local descriptors describe local properties of the image, which are usually attached to a particular feature, like corners, blobs or lines. These local descriptors are usually associated with a feature detector, which detects features where the local descriptors are extracted. SIFT by Lowe (2004), SURF by Bay et al. (2006), ORB by Rublee and Rabaud (2011), BRIEF by Calonder et al.

(2010) and FREAK by Alahi et al. (2012) are some popular approaches for extracting local descriptors. In order to make these descriptors usable by the classifier, the concept of bag-of-words is applied to all descriptors extracted from a single image, as described by Quelhas and Monay (2005). When these descriptors are applied in scenarios where lighting is highly variable, a preprocessing step is required so as to increase the illumination invariance, as performed by Petry et al. (2013).

On the side of global descriptors, the GIST operator, which is proposed by Oliva and Torralba (2001), was developed based on several experiments with humans and based on the concept that *Humans can recognize the gist of a novel image in a single glance, independent of its complexity*. In Oliva and Torralba (2006), it is possible to verify the good performance of this global descriptor in outdoor scenarios. However, other global descriptors can be found in the literature, such as: Composed Receptive Field Histograms (CRFH) Linde and Lindeberg (2004b), PCA of Census Transform Histograms (PACT) Wu and Rehg (2008), CENsus TRansform hISTogram (CENTRIST) Wu and Rehg (2010), Pyramid of Histograms of Orientation Gradients (PHOG) Bosch et al. (2007).

It should also be retained that a global descriptor is useful to describe a general scene and works quite well for outdoor scenes, as shown by Torralba and Murphy (2003), Meng and Wang (2010), and Rasiwasia and Vasconcelos (2012). However, the distinctiveness of indoor places is present on some place/scene details and not in the overall scene description. Therefore, the scene description should not be exclusively provided by a global descriptor but should also include local features and descriptors, like SIFT, SURF, pyramidal based consensus transform, among others, in order to get more details about the scene/place. Object centric scene place recognition and classification should in theory be more robust when compared to other techniques that do not use an higher concept for scene description. However, the purpose of the topological layer of HySeLAM is to recognize a place from a lower-scene description and increase its robustness by using the previous knowledge acquired from SLAM and abstracted by the HySeLAM framework, the augmented topological map.

Pronobis' work shows that the inclusion of observations collected by other sensors, like features extracted from laser measurements, can greatly improve place classification/recognition. This happens because these observations describe the scene by including the size of the area and the geometric shape of the place. These features seem to be important for indoor place classification. For instance, a corridor is usually described as an elongated rectangular area, and the rooms are described as square areas. In place classification/recognition systems relying only on sensor cameras, these features can be obtained by stereo vision systems.

Another important conclusion obtained from this global overview is that the construction of intermediate spaces between output classes and description can also help to increase the performance of the classifier. This is naturally done in the pLSA model as shown by Quelhas and Monay (2005). Pronobis' work shows a similar intermediate space and its advantage, but using the bag-of-features model.

For the classification stage, several approaches and techniques can be used. However, SVM-based approaches seem, from the literature review, the most consensual approach with the best performances and it is the one that requires the less tunning parameters when compared to neuronal base approaches. For these reasons, SVM was the classifier approach selected for this thesis work. A brief explanation about SVM theory is given in section 6.2.

In order to evaluate the best visual signature for place recognition and to answer the first question of this thesis chapter, which is *how must this visual signatures be constructed to allow for robust visual place recognition and classification*, a new descriptor for global description is presented in section 6.3, and a performance comparison between this descriptor and other local and global based descriptors is presented.

The HySeLAM framework formalizes a knowledge structure that encodes the place connections and organization, and it can be used to help improving place recognition/classification by filtering impossible place transitions. This subject and the answer to the question *How can the augmented topological map be included in the semantic localization procedure for increased place recognition accuracy?* are presented in section 6.4.

6.2 Support Vector Machines

SVMs belong to the class of large margin classifiers, and the theory behind SVM is extensively detailed by Cristianini and Shawe-Taylor (2000) and Vapnik (1998).

Consider the problem of separating the set of training data $(x_1,y_1),(x_2,y_2),...(x_m,y_m)$ into two classes, where $x_i \in \Re$ is a feature vector and $y_i \in \{-1,+1\}$ its class label. If it is assumed that the two classes can be separated by a hyperplane $w \cdot x + b = 0$ in some space $\mathscr{H}$, and that there is no prior knowledge about the data distribution, then the optimal hyperplane is the one that maximizes the margin, as stated by Vapnik (1998), or simply the one which has maximum distance to the closest points in the training set and is placed between these points, as illustrated by figure 6.8.

The optimal values for w and b can be found by solving the following constrained minimization problem:

$$minimize \frac{1}{2}\|w\|^2, \text{ subject to } y_i(w \cdot x_i + b) \geq 1, \forall i = 1,...m \tag{6.1}$$

Solving it using Lagrange multipliers $\alpha_i (i = 1,...m)$ results in classification function:

$$f(x) = sgn\left(\sum_{i=1}^{m} \alpha_i y_i x_i \cdot x + b\right) \tag{6.2}$$

where α_i and b are found by using a support vector clustering (*SVC*) learning algorithm, as it is detailed by Cristianini and Shawe-Taylor (2000), Vapnik (1998).

Most of the α_i's take the value of zero; x_i with nonzero α_i are the support vectors. In cases where the two classes are non-separable, the solution is identical to when they are separable except

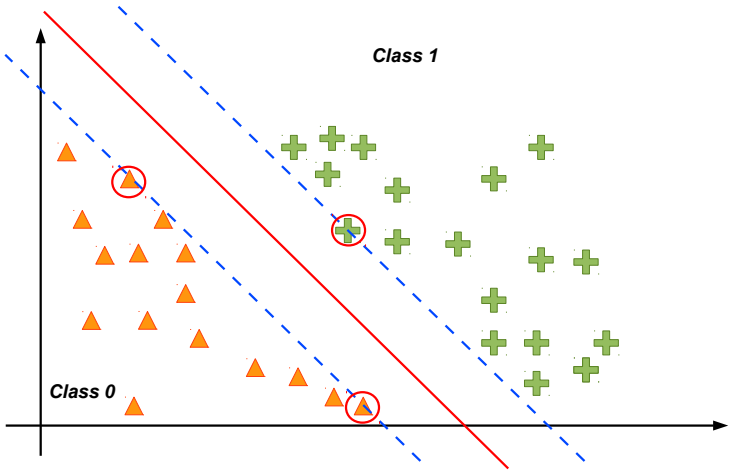

Figure 6.8: This image illustrates the maximum-margin hyperplane (red line) and margins (blue dash lines) for an SVM trained with samples from the two classes. Samples on the margin (red circles) are called "support vectors".

for a modification of the Lagrange multipliers into $0 \leq \alpha_i \leq C, i = 1, ...m$, where C determines the trade-off between margin maximization and error minimization.

To obtain a nonlinear classifier, the data is mapped from the input space $\Re^N$ to a high dimensional feature space $\mathscr{H}$ by $x \rightarrow \Phi(x) \in \mathscr{H}$, such that the mapped data points of the two classes are linearly separable in the feature space. Assuming there is a kernel function K such that $K(x,y) = \Phi(x) \cdot \Phi(y)$, then a nonlinear SVM can be constructed by replacing the inner product $x \cdot y$ in the linear SVM by the kernel function $K(x,y)$. This corresponds to constructing an optimal separating hyperplane in the feature space. Kernels commonly used are:

- polynomials $K(x,y) = (x \cdot y)^d$, which can be shown to map into a feature space spanned by all order d products of input features; and,

- Gaussian radial basis function (RBF): $K(x,y) = exp\{-\gamma \|x-y\|^2\}$, which sometimes is parametrized using $\gamma = 1/2\sigma^2$. However, in some cases, the best combination of C and γ is often selected by a grid search with exponentially growing sequences of C and γ, for example, $C \in \{2^{-5}, 2^{-3}, \ldots, 2^{13}, 2^{15}\}$; $\gamma \in \{2^{-15}, 2^{-13}, \ldots, 2^{1}, 2^{3}\}$. Typically, each combination of parameter choices is checked using cross-validation, and the parameters with best cross-validation accuracy are selected.

There are classification problems that have more than two classes, where a multiclass SVM approach is required. The dominant approach used in multiclass SVMs consists in reducing the single multiclass problem into multiple binary classification problems, by using concepts such as one-versus-all or one-versus-one. More details are given by Cristianini and Shawe-Taylor (2000), Vapnik (1998), and Kai-Bo and Keerthi; S. Sathiya (2005).

In this work, an open source implementation of SVM was used, which is described by Chang and Lin (2011) and the source code is available at http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

6.3 Visual signature for place recognition in indoor scenarios

Recognizing a place at a glance is the first capacity that humans use to understand where they are. Making it available to robots will allow an increase on the redundancy of the localization systems available and allow the implementation of semantic localization systems. However, this ability requires the setting up of a robust visual signature which can be used by a classifier for indoor scenarios.

The work presented in this section tries to show the kind of visual signatures that can be used in this augmented topological map, formalized in HySeLAM, to enable visual recognition of the place. The choice of the visual signature is constrained by the absence of higher landmarks/references, such as objects or particular references available in the places, as proposed by Lin and Wang. (2010). Nevertheless, it should be possible to mimic the human capacity to recognize a particular place at a glance, as shown by Oliva and Torralba (2001). Taking these constrains, the LBPbyHSV approach is proposed. LBPbyHSV extracts a global descriptor from an image that can be used as the visual signature for indoor scenarios. This global descriptor was tested using videos acquired from three robots in three different indoor scenarios. This descriptor have shown a good accuracy and computationally performance when compared to other local and global descriptors.

Section 6.3.1 presents the theory behind the LBPbyHSV global descriptor. Section 6.3.2 presents the performance comparison between this descriptor and other descriptors using real images from three robots in different places. Finally, section 6.3.3 presents the conclusions.

6.3.1 The LBPbyHSV global descriptor

The LBPbyHSV approach is the approach proposed here for the global descriptor extractor that can be used as a visual signature for place recognition. The LBPbyHSV is based on the Local Binary Pattern (LBP) operator and on the uniform patterns.

6.3.1.1 Local Binary Pattern Histogram

The theory behind the LBP operator and other LBP based approaches are described in detail by Pietikäinen et al. (2011). The success of the Local binary pattern methods is shown in several computer vision applications, due to the flexibility of the LBP, which makes it easily modifiable and makes this representation more adaptable for certain real world problems. The advantage of using LBP as a descriptor to describe features/images is the fact that it is computationally simple, robust in terms of gray scale variations, and efficient against illumination changes.

The original version of the local binary pattern operator, introduced by Ojala et al. (1996), works in a 3×3 pixel block of an image. The pixels in this block are thresholded by its center

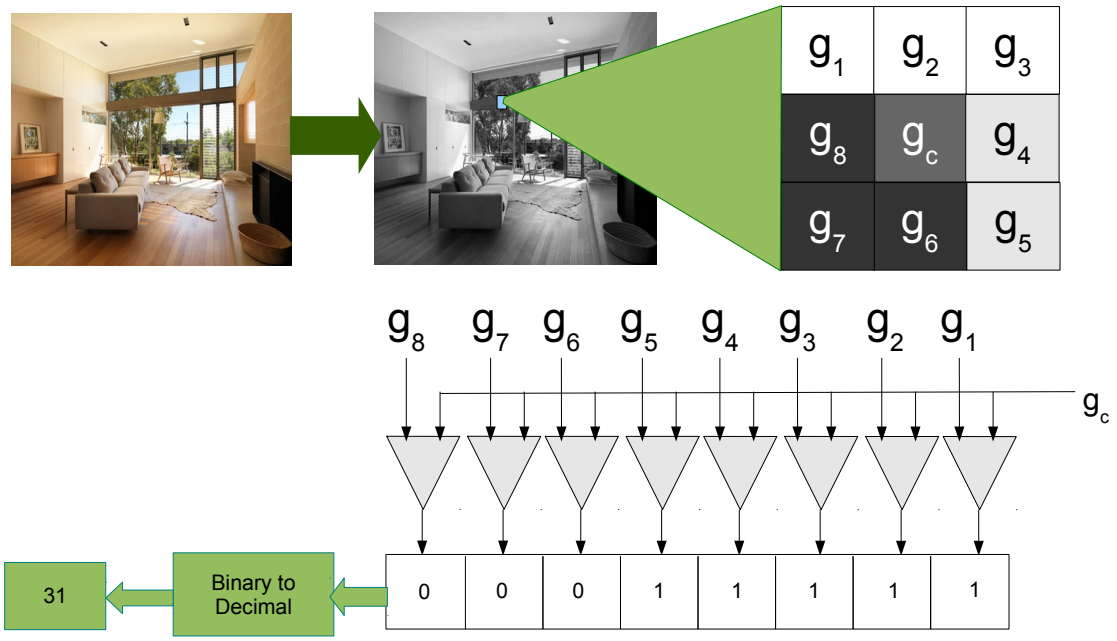

Figure 6.9: This image illustrates the $LBP_{8,1}$. Using a code, the LBP operator describes the pattern around a central pixel. To obtain this code, the image is converted to a gray scale image and the central pixel is compared to the neighbor pixels. This comparison results in a binary value of "0" if the central pixel has a higher intensity value, or "1" if the central pixel has a lower intensity value.

pixel value, multiplied by powers of two and then summed to obtain a label for the center pixel. As the neighborhood consists of 8 pixels, a total of $2^8 = 256$ different labels can be obtained depending on the relative gray values of the center and the pixels in the neighborhood, as shown in figure 6.9. The label returned by the LBP operator is defined as a natural number which is given by equation 6.3.

$$LBP_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_c)2^p, \text{with } s(x) = \begin{cases} 1, x \geq 0 \\ 0, x < 0 \end{cases} \tag{6.3}$$

Where $s(x)$ represents the signal function, g_c is the gray value of the central pixel, g_p is the value of its neighbor pixels and P represents the number of sampling points in the neighborhood.

The sampling process is performed commonly in the clockwise direction, around the central pixel according to a particular radius value R, which determines the spatial resolution of the distributed sampling points. The sampling process is given by:

$$g_p = I\{x_p, y_p\}, p = \{0, 1 \ldots (P-1)\} \tag{6.4}$$

$$x_p = x + Rcos(2\pi p/P) \tag{6.5}$$

$$y_p = y - Rsin(2\pi p/P) \tag{6.6}$$

After the LBP pattern of each pixel in the input image (gray) I_{gray} of size $M \times N$ is identified, it is possible to build a histogram of LBP patterns as follows:

$$H(k) = \sum_{m=1}^{M} \sum_{n=1}^{N} f(LBP_{P,R}(m,n),k), \text{ with } k \in [0,K], \tag{6.7}$$

$$f(x,y) = \begin{cases} 1, x = y \\ 0, otherwise \end{cases} \tag{6.8}$$

Where, K is the maximal LBP pattern values.

Note that local binary pattern code $LBP_{8,1}$ is also known as Census Transform (there is a difference in bit ordering). A histogram of CT values for an image or image patch is also known as CENTRIST (CENsus TRansform hISTogram).

6.3.1.2 Uniform patterns

In order to increase the robustness of the LBP operator, usually this operator is extended to the approach called *uniform patterns* ($LBP^{u2}_{P,R}$) by Ojala et al. (2002), the superscript "$u2$" refers to the "$uniform$" patterns with ($U \leq 2$), which represents the number of spatial transitions allowed in a circular binary form. When measuring uniformity in the U pattern (see equation 6.10), the uniformity designation merges the bitwise transition from 0 to 1 and vice versa. The LBP pattern is considered uniform if its uniformity measurement is less or equal to 2.

$$LBP^{u2}_{P,R} = \begin{cases} \sum_{p=0}^{P} s(g_p - g_c)2^p & \text{, if } U(LBP_{P,R}) \leq 2 \\ P(P-1)+3 & \text{, Otherwise} \end{cases} \tag{6.9}$$

where:

$$U(LBP_{P,R}) = \mid s(g_{p-1} - g_c) - s(g_0 - g_c) \mid + \sum_{p=1}^{P-1} \mid s(g_p - g_c) - s(g_{p-1} - g_c) \mid \tag{6.10}$$

In uniform LBP mapping there is a separate output label for each uniform pattern and all non-uniform patterns are assigned to a single label. Thus, the number of different output labels for mapping patterns of P bits is $P(P-1)+3$. For instance, the uniform mapping produces 59 output labels for neighborhoods of 8 sampling points, and 243 labels for neighborhoods of 16 sampling points.

6.3.1.3 LBPbyHSV

In order to include the color information into the image descriptor, the LBPbyHSV approach is proposed. Instead of applying the LBP operator over different color channels, such as in OCLBP by Mäenpää and Pietikäinen (2004), HSV-LBP and RGB-LBP by Banerji et al. (2011), the LBPbyHSV generates a LBP histogram for each basic color.

The LBPbyHSV approach segments the image into n regions, with the same pixel color information. This segmentation is based on the definition of n basic colors. These basic colors are obtained from a HSV color space segmentation. The HSV color space was selected, because it is more robust to light variations than the RGB color space. A LBP histogram is extracted in each one of these n image regions, and then these histograms are concatenated into a single histogram, as shown in figure 6.10. To form the LBPbyHSV global descriptor, another LBP global descriptor is concatenated to this histogram.

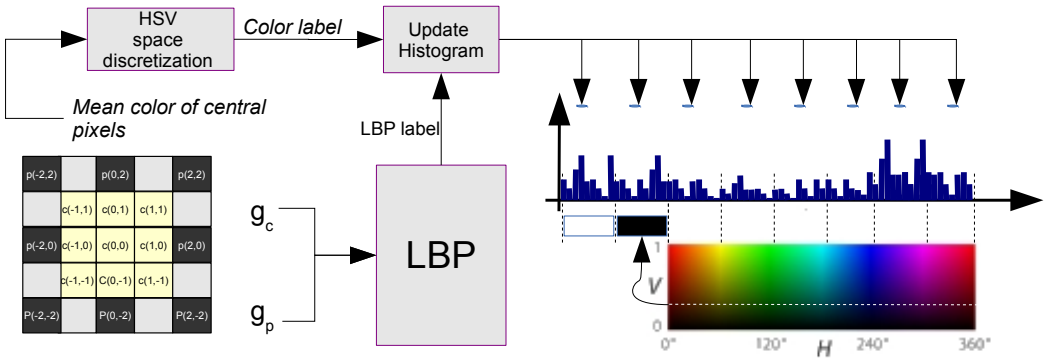

Figure 6.10: The LBPbyHSV approach clusters the LBP histogram into n basic colors. Then the LBP label and the basic color of the central pixel are extracted from each image pixel, which then increments the bin related to the LBP label in the color set bin related to the central pixel.

The output descriptor is labeled as LBPbyHSV-XnY, where X refers to the LBP approach used in the first stage (S if the standard LBP operator is used, U if the uniform LBP operator is used), n refers to the number of segmented colors in the HSV space, and Y refers to the LBP approach used in the second stage (S if the standard LBP operator is used, U if the uniform LBP operator is used, N if no operator is used). Five variants were tested: LBPbyHSV-U8N, LBPbyHSV-U8U, LBPbyHSV-U8S, LBPbyHSV- U16N, and LBPbyHSV-U16S.

In the LBPbyHSV approach, the LBP pattern of each pixel in the input image I of size $M \times N$ is identified and the pixel color is associated with the one with the basic colors; thereafter, the histogram for each basic color in the image is built as follows:

$$H_c(k) = \sum_{m=1}^{M} \sum_{n=1}^{N} f(LBP_{P,R}(m,n), k, hsvc(m,n), c), \text{ with } k \in [0,K], \tag{6.11}$$

Where K is the maximal LBP pattern value (which depends on the LBP operator used, uniform or standard, and on the number of sample points P), with:

$$f(x,y,x_c,y_c) = \begin{cases} 1, x = y \wedge x_c = y_c \\ 0, otherwise \end{cases} \tag{6.12}$$

and,

$$hsvc(m,n) = \begin{cases} 0 & , \text{ if } p_V(m,n) < \delta_v \\ 1 & , \text{ if } p_S(m,n) < \delta_s \wedge p_V(m,n) \geq \delta_v \\ int(p_H(m,n)/\kappa) + 2 & , \text{otherwise} \end{cases} \tag{6.13}$$

Where $p_V(m,n)$ $p_S(m,n)$ and $p_H(m,n)$ are functions that return the value of HSV components from the pixel $p_{m,n}$ of the original I_{hsv} image, δ_v defines the limit value for the black color definition, δ_s defines the value of saturation for the white colors, and κ is given by $\frac{360}{(n-2)}$, where n is the number of segmented colors (black and white are always considered) and 360 is the maximum value that the Hue component can obtain (using the opencv library this value should be set as 180).

These histograms are then concatenated into a single one, $H = [H_0, H_1, ..., H_n, H_{LBP}]$, where H_{LBP} is the second LBP histogram applied (standard, uniform or none depending on the LBPbyHSV variant) over the $I_{LBPbyHSV}$ image. This intermediate image $I_{LBPbyHSV}$ is a gray scaled image, and the pixel values are obtained as follows:

$$g(x,y) = LBP_{P,R}(m,n) + hsvc(m,n)K \tag{6.14}$$

In the LBPbyHSV approach, the $LBP_{P,R}$ operator used is also slight modified when $R > 1$, in these cases, the central value g_c is not only obtained from the central pixel value, but also obtained as mean value from the pixels inside radius R, as shown in figure 6.10.

6.3.2 Comparison of descriptor performances

Two videos were created with images collected from three different robots and places for the purpose of comparing the performance of the LBPbyHSV approach with other descriptors for indoor place recognition. One of the videos was used for classifier training and the another video for accuracy testing.

The SVM classifier with three kernels (Linear, Polynomial and Radial Basis Function) was used for the classification stage. However, other techniques can be applied, such as: Neural Networks (NN), Adaboost, Bayes based classifiers. The theory behind SVM is briefly described in section 6.2 and described in detail by Cristianini and Shawe-Taylor (2000) and by Vapnik (1998). This work has used an SVM open source implementation, the libsvm by Chang and Lin (2011), which can be found at `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`.

Table 6.1: Performance comparison of global and local descriptors. The time column presents the average time required by the descriptor approach to obtain the descriptor vector; a variance of this average is presented in brackets. The SVM Accuracy presents the accuracy obtained in the testing video using the descriptor and the SVM kernel, after a training stage with the training video.

Approach	Descriptor size	Time ms	SVM Accuracy		
			Linear	Poly	RBF
LBPbyHSV-U8N	472	13.505(0.001)	71.32	71.91	67.24
LBPbyHSV-U8U	530	27.634 (0.004)	72.49	71.45	71.39
LBPbyHSV-U8S	737	28.376 (0.007)	72.59	72.01	70.27
LBPbyHSV-U16N	944	14.249 (0.002)	72.58	72.91	71.24
LBPbyHSV-U16S	1200	26.645 (0.004)	74.53	74.31	73.24
HSV-LBP	768	29.319 (0.003)	**75.27**	68.77	66.42
HSV-LBP-u	174	29.670 (0.003)	73.65	74.01	67.15
SLBP(1+2+4) Hu and Guo (2012)	1218	19.125 (0.004)	75.12	75.22	71.37
HSV-GIST (32x32)	960	8.827 (0.003)	66.08	69.95	19.81
HSV-GIST (64x64)	960	45.187 (0.009)	69.19	70.17	28.79
HSV-GIST (128x128)	960	188.959 (0.101)	69.70	70.05	28.05
BOF-SIFT (200 words)	200	91.849 (0.297)	53.90	58.52	61.45
BOF-SIFT (500 words)	500	96.859 (0.347)	55.80	59.03	64.70
BOF-SIFT (1000 words)	1000	98.492 (0.391)	57.50	56.51	65.56
BOF-SIFT (2000 words)	2000	95.866 (0.334)	61.36	54.57	65.43
BOF-SURF (500 words)	500	140.995(3.697)	57.99	56.20	65.03
BOF-SURF (2000 words)	2000	145.210(3.761)	65.01	58.82	60.01
BOF-ORB (200 words)	200	23.837(0.031)	41.64	45.30	46.16
BOF-ORB (500 words)	500	30.693(0.082)	45.36	39.40	44.81
BOF-SURF+BRIEF (500 words)	500	90.646 (0.513)	49.35	46.43	51.27
BOF-SURF+BRIEF (2000 words)	2000	108.547(1.401)	48.77	34.82	47.62
BOF-SURF+FREAK (500 words)	500	87.055 (0.454)	32.99	35.88	42.19
BOF-SURF+FREAK (2000 words)	2000	108.628(1.503)	31.22	30.99	37.27

Table 6.2: Performance comparison of global and local descriptors, which encodes in a single descriptor all, half top and half bottom image (these descriptors are identified by (D) extension). And, performance comparison of LBPbyHSV descriptor concatenated with different local descriptors.

Approach	descriptor size	Time ms	SVM Accuracy Linear	Poly	RBF
LBPbyHSV-U8N-D	1416	14.620 (0.002)	74.93	76.30	**76.69**
			(+3.61)	(+4.39)	(+9.45)
LBPbyHSV-U8N-IRIS	2360	18.113 (0.001)	72.09	72.40	72.22
			(+0.77)	(+0.49)	(+4.98)
BOF-ORB-500-D	1500	28.269 (0.121)	44.34	45.64	50.96
			(-1.02)	(+6.24)	(+6.15)
BOF-SIFT-500-D	1500	91.809 (0.388)	55.62	58.27	60.40
			(-0.20)	(-1.03)	(-4.30)
BOF-SURF-500-D	1500	116.750 (2.058)	60.25	62.35	60.69
			(+2.01)	(+3.32)	(-4.01)
LBPbyHSV-U8N-D + BOF-ORB500	1916	43.465 (0.112)	72.69	74.42	75.46
LBPbyHSV-U8N-D + BOF-SIFT500	1916	105.571 (0.380)	73.84	**76.42**	75.07
LBPbyHSV-U8N-D + BOF-SURF500	1916	150.000 (3.384)	75.48	75.21	74.78

The descriptors test were conducted by a ROS node application, running in a 2,16 GHz Intel Pentium Dual Core T4300 processor, with 2GB of memory, and with a robotic operating system (ROS) (fuerte version) over the Ubuntu OS (12.04). The HySeLAM framework implementation, available in `hyselam.com`, was also used to manage the groundtruth of the two videos by publishing a semantic localization topic.

The localization topic is limited to a set of human words. Each human word tags a specific place, in this work the set of human words is: *Freiburg Corridor(ss0), Freiburg Printer office(ss1), Freiburg Room A(ss2), Freiburg Stairs(ss3), Freiburg WC(ss4), Ljubljana Corridor(ss5), Ljubljana Stairs(ss6), Ljubljana Room A(ss7), Ljubljana WC (ss8), FEUP Corridor(ss10), FEUP I-110(ss9), FEUP room I-108(ss11), FEUP room I-109(ss12), and FEUP room I-111(ss13)* . The names inside brackets represents the short identification used in the graphs.

The two videos used are characterized as follows:

- Testing video: it contains 7320 frames with a 640x480 resolution. These frames were collected from three robots at three different places: Faculdade de Engenharia da Universidade do Porto (FEUP), University of Freiburg in Germany, and University of Ljubljana in Slovenia. The FEUP video was acquired in a laboratory and it is 337 seconds long at 10fps with a resolution of 640x480. The other two videos are available in the COLD database, by Pronobis and Caputo (2009) in `http://www.cas.kth.se/COLD/index.php`. The Freiburg video is 182 seconds long and the Ljubljana video is 213 seconds long, and both were acquired at 10fps with a resolution of 640x480 on a cloudy day.

- Training video: it contains 6254 frames with the same resolution and frame rate. These

frames were collected from the same robots, but with slightly different path, time and illumination conditions. Here the FEUP video is 435 seconds long, the Freiburg video is 171 seconds long, and the Ljubljana video is 194 seconds long.

It is important to highlight that both videos were obtained at different moments, in different paths and weather conditions (cloudy and sunny), and with people present in the scenes. The LBP operator used in LBP based approaches has 8 samples and a radius of 3 ($LBP_{8,3}$). The SVM Poly kernel used is a polynomial function of second order.

The time taken by SVM in the classification stage was not analyzed in these tests because they are negligible, when they are compared against the time taken by the descriptor extraction. In average the classification stage has take between 8 to 12 *us*. In contrast, the training time taken by SVM is significantly high. Depending on the descriptor used to train the SVM, the time taken was between 2 to 25 minutes.

6.3.3 Discussion of the results

The proposed approach is more accurate than the local based descriptors and more accurate than the GIST based approach, as shown by table 6.1. However, the LBPbyHSV variant approaches have a slightly worse accuracy than the HSV-LBP and SLBP, but the processing time is faster (more than 2 times) and the descriptor is more compact. This makes the LBPbyHSV approach interesting for real time applications running in embedded systems, because it encodes the color image information on the descriptor faster than the HSV-LBP.

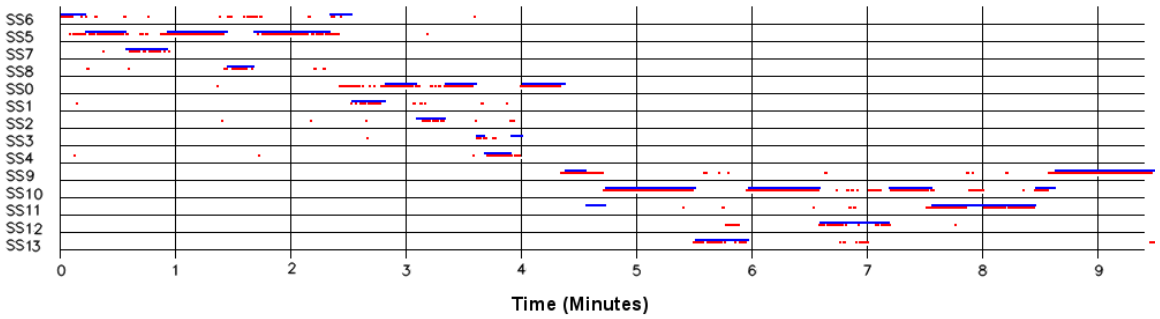

Figure 6.11: Semantic localization provided by the LBPbyHSV approach. The semantic localization provided by the HySeLAM (groundtruth) is in blue, the semantic localization provided by the LBPbyHSV-U16S with SVM (Linear kernel) is presented in red.

A visual descriptor which is only based on BOF concept and local descriptors, such as SIFT,SURF, ORB, BRIEF, FREAK, is not the best approach for place classification. This conclusion is obtained from table 6.1 and it is based on fact that: these descriptors take more time to process, are less accurate and require an additional step to cluster the descriptor space into the main representative descriptors that forms the features dictionary.

A further analysis of the local descriptors shows that they describe local proprieties of interesting key-points in the images. This is an useful approach to finding image correspondences, but it is not the best approach to construct visual signatures that are used in image/scene classification problems. In contrast, the key-points detector step can be useful to describe the richness of the image, which can be used as a feature for the SVM input vector. This input feature can be useful to detect non-descriptive images, such as those shown in figure 6.12. These non-descriptive images exist in all classes and should be filtered from the training and classification step, because they reduce the accuracy of the place classification.

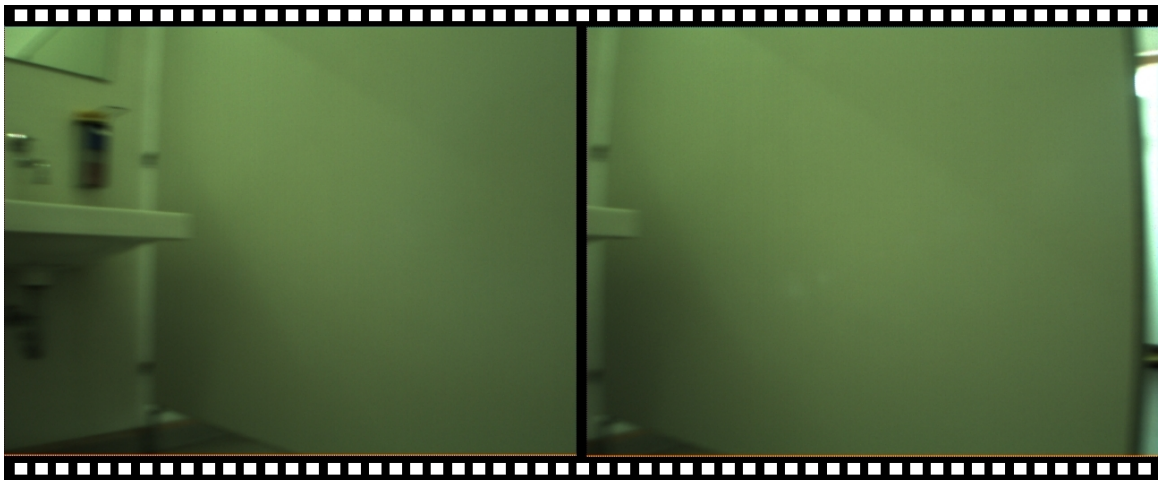

Figure 6.12: This image shows two frames from the Freiburg university path. These images belong to the set of non-descriptive images that reduce the accuracy of the classifier. As the reader can see, these images do not include enough features for the place they belong to to be detected.

In general, table 6.1 shows that the RBF based kernel has improved the accuracy of the approaches based on local descriptors. On the other hand, the RBF based kernel has slightly reduced the accuracy of approaches based on global descriptors. Indeed, on the tests using the HSV-GIST descriptor, the over-fitting problem has appeared, which has resulted in very low accuracy in the testing video.

The LBPbyHSV approach was tested using an image slicing technique. In the LBPbyHSV-U8N-D, the image is sliced in two parts (bottom and top) and then the global descriptor is extracted for each half and after that both descriptors are concatenated with their sum. This is a simpler version of the pyramidal concept described by Hu and Guo (2012). Table 6.2 allows us to conclude that the performance has slight improvement when the linear and the polynomial kernels are used, and it has significant improvement when the RBF kernel is used. In contrast, when ORB and SIFT based approaches (BOF-ORB-500-D, BOF-SIFT-500-D) are used, accuracy is slightly reduced; if a polynomial or a RBF based kernel is used on the ORB based approach, accuracy is slightly improved; if a linear or a polynomial based kernel is used on the SURF based approach, accuracy is slightly improved .

Figure 6.13: This image shows four frames from the Ljubljana path. These images are from the fuzzy/gray zone, where it is hard to tell if they are acquired from the Ljubljana WC or the Ljubljana corridor.

Taking into account that the pyramidal concept can encode on the descriptor the spacial organization of the features and taking into account that indoor images contains wall, floor and ceiling, it was tested a different image slicing. In the LBPbyHSV-U8N-IRIS, the image is sliced into four parts with the shape of triangles, by using the circular projection concept proposed by Cao et al. (2010). When the correct perspective is taken, on the upper triangle is extracted the descriptor for the ceiling, on the left and right triangle is extracted the descriptor for the walls and on the bottom triangle is extracted the descriptor for the floor. These descriptors are concatenated with the sum of all descriptors. From table 6.2, this approach shows a slightly better accuracy than LBPbyHSV-U8N but a worse accuracy when compared to LBPbyHSV-U8N-D. The reason for this worse accuracy may reside on the fact of the common triangle vertex be static and parametrized on the image center instead of being placed on the vanishing point of the scene/image. As a future work, it should be explored the association of this common triangle vertex to the vanishing point.

Transitions between places is a challenge for the classifier, as it is for the humans when they are

watching the videos, as illustrates the figure 6.13, it is hard to tell if we are seeing the scene from Ljubljana WC or from Ljubljana corridor. In these cases, the descriptor changes smoothly between place transitions, which puts the descriptor in the frontier of two classes where the classifier has less accuracy.

From figure 6.11, it is possible to find impossible state transitions, most of them are related to a classification of a non descriptive image (for example an image of a white wall, such as figure 6.12) or related to a classification of a similar place. The inclusion of the augmented topological map in classification stage will allow to remove these outliers. Therefore, in order to improve the classification accuracy, it is proposed on section 6.4 the use of the augmented topological map, defined in the HySeLAM framework, in order to create a semantic localization based on graph transitions, where transitions are managed by the classification provided by the visual signature and the by classifier. This approach will make it possible to detect these transitions, which will help improve the accuracy of the classification. Here, emerges the question: *How can the augmented topological map be included in the semantic localization procedure for increased place recognition accuracy*.

6.4 SeLoViS - Semantic Localization based on Visual Signatures

It is important to highlight that this section formalizes the semantic localization procedure, which is a part of the *TopoVisualSignatures* component, formalized in the HySeLAM framework, section 3.2. In this work, the semantic localization procedure localizes the robot in a space defined by human words instead of a geometric space, as in a conventional SLAM. As this procedure is a part of the *TopoVisualSignatures* component, it is constrained to use only visual signatures and the augmented topological map.

To answer the question *How can be included the augmented topological map into the semantic localization procedure in order to increase the accuracy of the visual place recognition skill?* and to build this semantic localization procedure, it is important to summarize the conclusions obtained in section 6.3. The main conclusion was: LBPbyHSV is a fast global descriptor extractor which encodes the color information on the descriptor and that can be used as a visual signature for place recognition. However, three main problems were detected when the visual place recognition procedure is based only in visual signatures and on a SVM classifier. These are:

- The place recognition allows impossible place state transitions. For example, as depicted in figure 6.11, the place recognition detects in one moment that the robot is in room I-111 at FEUP, and the next moment it detects that it is in room I-110 at FEUP, without crossing the corridor that is the only connection between the two rooms.

- There are images which are poor in terms of scenario description, such those on figure 6.12. These images are noise for the place recognition approach, and may result in classification outliers.

- The transition from place to place is also a challenge for place recognition, because the visual signatures of the images are in the frontier of the neighboring classes/places. Is is even a challenge for humans, as figure 6.13 shows. These place transitions are usually related to crossing a door.

These problems appear because the previous visual place recognition procedure:

- allows the probability to flow in a random and unstructured way between successive SVM classifications;
- does not include the definition of place transition, where the robot can be in the frontier of two places; and,
- does not ignore non-descriptive images;

If the visual semantic localization procedure includes knowledge about place connections, the probability can be constrained to flow according to the structure defined in the topological map. This knowledge also makes it possible to detect the moments where the robot is placed on the border between two places. The Markov chains formalism was selected to constrain the probability to flow according to the topological structure. This procedure inherits the augmented topological map, as in section 6.4.2. In order to remove sporadic recognition outliers from non-descriptive images, a simple filtering process from signal processing theory was tested, as in section 6.4.1. The performance of both approaches is compared in table 6.5.

6.4.1 A direct filter over the classification

In order to reduce the negative effect on the visual place recognition procedure from single outliers, such as those obtained from images that are non-descriptive, a simpler filter applied over each probability class $p_k(x)$ was tested. $p_k(x)$ is provided by the multi-class SVM at each instant k. This filter is based on a simple low-pass filter theory, from signal processing theory. The result of the multi-channel filters is normalized in each iteration, so as to satisfy the probabilities constrain $\sum_{\text{all x}} p_k(x) = 1.0$

$$p_k(x) = \frac{\hat{p}_k(x) + \alpha_k p_{k-1}(x)}{\sum_{\text{all x}} \hat{p}_k(x) + \alpha_k p_{k-1}(x)} \tag{6.15}$$

Where $p_k(x)$ is the probability, provided by the filter, for the robot to be in place x; $\hat{p}_k(x)$ is the probability obtained from SVM for the robot to be in be in place x at the instant k; α_k is the filter parameter.

In order to test the performance of the procedure using this filtering approach, the same training and testing data described in section 6.3.2 were used. Exactly the same input was provided to this new approach and the parameter filter α_k was set with value 2.

After these tests, accuracy was improved in all descriptors tested in section 6.3.2, as can be verified by comparing the results in tables 6.3 and 6.4 against tables 6.1 and 6.2.

Table 6.3: Performance comparison of global and local descriptors, in the visual place recognition procedure using a simpler filter. The time column presents the average time required by the descriptor approach to obtain the descriptor vector; a variance of this average is presented in brackets. The SVM Accuracy presents the accuracy obtained in the testing video using the descriptor and the SVM kernel, after a training stage with the training video.

Approach	Descriptor size	Time ms	SVM Accuracy Linear	Poly	RBF
LBPbyHSV-U8N	472	13.505 (0.001)	74.77	76.81	72.65
LBPbyHSV-U8U	530	27.634 (0.004)	75.38	75.44	75.17
LBPbyHSV-U8S	737	28.376 (0.007)	75.80	75.81	75.38
LBPbyHSV-U16N	944	14.249 (0.002)	77.01	76.91	76.12
LBPbyHSV-U16S	1200	26.645 (0.004)	79.36	79.02	78.24
HSV-LBP	768	29.319 (0.003)	**81.25**	74.75	72.15
HSV-LBP-u	174	29.670 (0.003)	78.31	80.00	72.05
SLBP(1+2+4) Hu and Guo (2012)	1218	19.125 (0.004)	77.66	78.45	75.47
HSV-GIST (32x32)	960	8.827 (0.003)	73.21	75.81	~~19.71~~
HSV-GIST (64x64)	960	45.187 (0.009)	75.58	77.73	~~27.05~~
HSV-GIST (128x128)	960	188.959 (0.101)	76.29	77.00	~~25.86~~
BOF-SIFT (200 words)	200	91.849 (0.297)	63.10	67.42	68.78
BOF-SIFT (500 words)	500	96.859 (0.347)	64.26	68.45	74.86
BOF-SIFT (1000 words)	1000	98.492 (0.391)	63.64	60.43	71.83
BOF-SIFT (2000 words)	2000	95.866 (0.334)	68.71	57.52	72.31
BOF-SURF (500 words)	500	140.995(3.697)	63.46	60.37	71.53
BOF-SURF (2000 words)	2000	145.210(3.761)	70.20	64.70	66.38
BOF-ORB (200 words)	200	23.837(0.031)	51.60	53.05	53.95
BOF-ORB (500 words)	500	30.693(0.082)	54.86	45.80	52.81
BOF-SURF+BRIEF (500 words)	500	90.646 (0.513)	58.16	54.82	57.64
BOF-SURF+BRIEF (2000 words)	2000	108.547(1.401)	57.42	40.26	54.06
BOF-SURF+FREAK (500 words)	500	87.055 (0.454)	40.03	41.34	50.00
BOF-SURF+FREAK (2000 words)	2000	108.628(1.503)	39.01	34.05	44.26

Table 6.4: Results from the visual place recognition procedure using a simpler filter. Performance comparison of global and local descriptors, which encodes in a single descriptor all, half top and half bottom image (these descriptors are identified by (D) extension). And, performance comparison of LBPbyHSV descriptor concatenated with different local descriptors.

Approach	descriptor size	Time ms	SVM Accuracy Linear	Poly	RBF
LBPbyHSV-U8N-D	1416	14.620 (0.002)	78.88 (+4.11)	**79.73** (+2.92)	79.72 (+7.07)
LBPbyHSV-U8N-IRIS	2360	18.113 (0.001)	76.83 (+2.06)	78.79 (+1.98)	75.32 (+2.67)
BOF-ORB-500-D	1500	28.269 (0.121)	55.30 (+0.44)	55.16 (+9.36)	60.91 (+8.10)
BOF-SIFT-500-D	1500	91.809 (0.388)	64.04 (-0.22)	66.31 (-2.14)	68.79 (-6.07)
BOF-SURF-500-D	1500	116.750 (2.058)	67.68 (+4.22)	67.13 (+6.76)	67.37 (-4.16)
LBPbyHSV-U8N-D + BOF-ORB500	1916	43.465 (0.112)	76.69	77.98	80.28
LBPbyHSV-U8N-D + BOF-SIFT500	1916	105.571 (0.380)	77.55	79.09	78.84
LBPsbHSV-U8N-D + BOF-SURF500	1916	150.000 (3.384)	**80.90**	77.81	79.91

The results obtained allow to conclude that the use of this simple filter in each probability class is a method that can mitigate the negative effects of single outliers, which are produced by non-descriptive images or by a similar perspective of two different places. However, when the ground-truth graph (similar to figure 6.11) from these results is compared against those obtained in previous tests (section 6.3.2), it was observed that the filter can not handle multiple successive outliers and accuracy remains low in transition places.

6.4.2 Markov Chain Formalism

In this subsection the Markov chains formalism done by Cassandras, Christos G., Lafortune (2008) is summarized. Markov chains provide a rich framework for studying many discrete event systems of practical interest, ranging from gambling and the stock market to the design of "high-tech" computer systems and communication networks.

In the Markov chains processes, discrete-time Markov chain events (and hence state transitions) are constrained to occur at time instants $0,1,2,\ldots,k,\ldots$. Thus, it is formed a stochastic sequence $\{X1,X2,\ldots\}$ which is characterized by the Markov (memoryless) property:

$$P[X_{k+1}=x_{k+1}|X_k=x_k,X_{k-1}=x_{k-1},\ldots,X_0=x_0]=P[X_{k+1}=x_{k+1}|X_k=x_k] \tag{6.16}$$

Given the current state x_k, the value of the next state depends only on x_k and not on any past state history (no state memory). Moreover, the amount of time spent in the current state is irrelevant in determining the next state (no age memory).

The first step is to model the stochastic discrete event systems by means of the stochastic timed automaton formalism based on the six-tuple:

$$(\varepsilon, \chi, \Gamma, p, p_0, G) \tag{6.17}$$

where:

- ε is a countable event set, which (in this thesis work) is inherited from the attributed graph the place connections, equation 3.1.
- χ is a countable state space, which (in this thesis work) is inherited from the attributed graph the places set, equation 3.1.
- $\Gamma(x)$ is a set of feasible or enabled events, defined for all $x \in \chi$ with $\Gamma(x) \subseteq \varepsilon$
- $p(x';x,e')$ is a state transition probability, defined for all $x,x' \in \chi, e' \in \varepsilon$ and such that $p(x';x,e') = 0$ for all $e' \notin \Gamma(x)$
- $p_0(x)$ is the pmf $P[X_0 = x], x \in \chi$ of the initial state X_0
- $G = \{G_i : i \in \varepsilon\}$ is a stochastic clock structure.

State transitions are driven by events belonging to the set ε. Thus, the transition probabilities are expressed as $p(x';x,e')$ where $e' \in \Gamma(x)$ is the triggering event, and $\Gamma(x)$ is the feasible event set at state x. In Markov chains, however, we will only be concerned with the total probability $p(x',x)$ of making a transition from x to x', regardless of which event actually causes the transition, with:

$$p(x',x) = \sum_{i \in \Gamma(x)} p(x';x,i)p(i,x) \tag{6.18}$$

where $p(i,x)$ is the probability that event i occurs at state x. This transition probability is an aggregate over all events $i \in \Gamma(x)$ which may cause the transition from x to x'. In general, it is allowed to the transition probabilities to be dependent on the time instant at which the transition occurs, so:

$$p_k(x',x) = P[X_{k+1} = x' | X_k = x] \tag{6.19}$$

Since the clock structure G is implicitly defined by a Markov property, by Cassandras, Christos G., Lafortune (2008), to specify a Markov chain model is only required to identify:

1. A state space X.
2. An initial state probability $p_0(x) = P[X_0 = x]$, *for all* $x \in X$.
3. Transition probabilities $p(x',x)$ where x is the current state and x' is the next state.

It should be highlighted that the transitions probabilities are constrained by:

$$\sum_{\text{all } x'} p_k(x',x) = 1 \tag{6.20}$$

Whenever the transition probability $p_k(x',x)$ is independent of k for all $x',x \in X$, it is obtained a homogeneous Markov chain. In this thesis work, the transition probability $p_k(x',x)$ is not independent of k. Therefore at each k time, it will obtained a different transition probability $p_k(x',x)$.

6.4.3 Visual semantic localization based on Markov chain formalism

The motion process of a robot, in the topological map space, can be described by the Markov chains framework. The place were the robot will be in the next $k+1$ time is only dependent on:

- the place where it was at k, the memoryless property; and
- the place connections where it was at k, the activated transition edges and the associated labeled probabilities.

Here, a semantic localization procedure based on Markov chains formalism is proposed, where the transition probabilities $p_k(x',x)$ depend on k. They are governed by the classification given by the classifier based on the visual signature acquired at instant k. The Markov chains are constructed over the augmented topological map and inherit the vertices and edges; the edges are split into directional edges, which are labeled with transition probabilities. A self transition edge and a state probability $p_k(x)$ are added to each vertex, as illustrated in figure 6.14.

In order to update the transition probability $p_k(x',x)$ in each k step, the definition of distance from the directed graph theory is recalled, where the distance $d(u,v)$ between two vertices/states u and v is defined as the length of the shortest path from u to v, consisting of edges, provided at least one such path exists. Taking as example the topological map illustrated in figure 6.14, the distance between the vertices *Marta office* and *Garden* is 4.

This distance function $d(u,v)$ is used by the proposed approach to build the matrix of observation influence M_{ObIn}, where:

$$m_{ObIn}(i,j) = d(i,j) \tag{6.21}$$

Another matrix $M_{\eth}$ is also built based on the M_{ObIn}, with:

$$m_{\eth}(i,j) = \eth(i,j) \tag{6.22}$$

where $\eth(i,j)$ is the function that returns the first vertex/state connected to j in the shortest path found by $d(i,j)$. From these two matrices, the transition probability for each directional edge can be defined:

$$p_k(\eth(i,j),j) = \left(\frac{\rho_1}{d(i,j)}\right)^{\rho_2} * \Upsilon(i) * max(\Upsilon(0),...,\Upsilon(t)),\ with\ t = |\chi| \wedge i \neq j \tag{6.23}$$

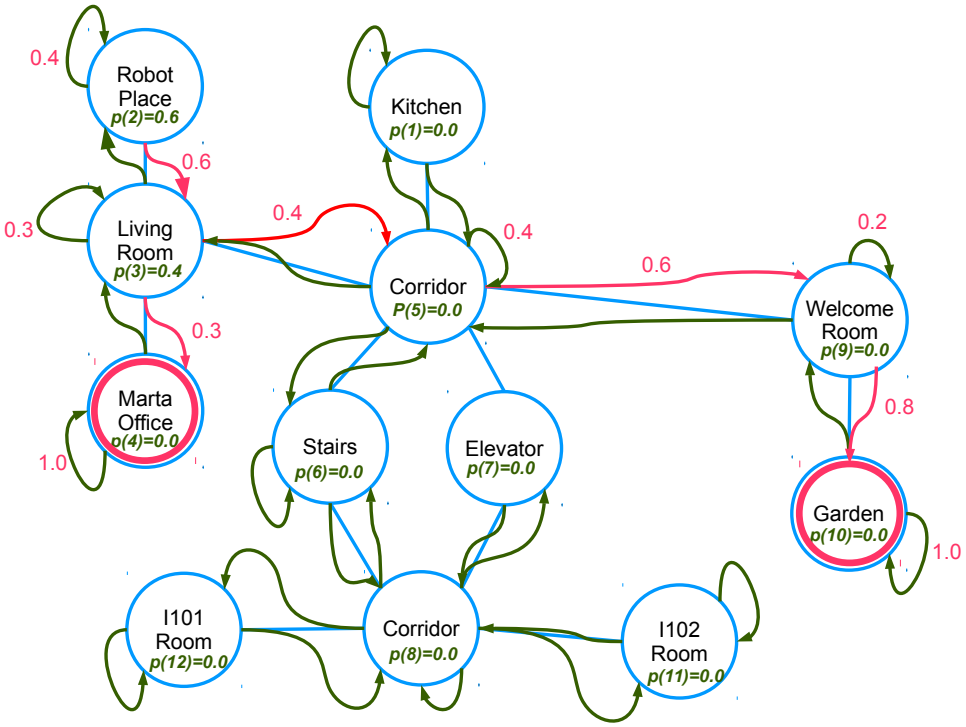

Figure 6.14: This figure illustrates a Markov chains representation (for the motion process) over an augmented topological map. The green edges represent the activated state transition edges that are built from the original place connections (in blue). The state probability and a self transition edge are included in each vertex. Each edge is labeled with the state transition probability value.

where, ρ_1 and ρ_2 are parameters that rule the influence of the state probability $\Upsilon(i)$, which was provided by the classifier at instant k, in this edge $p_k(\circlearrowright(i,j),j)$; $\Upsilon(i)$ is the rating classification given by the classifier for the class/state/vertex i based on the visual signature gathered at the k step. In order to satisfy the constraint equation 6.20 and allow permanence in the current state, the self transition probability $p_k(j,j)$ is defined as:

$$p_k(j,j) = 1 - \sum_{\text{all } x'} p_k(x',j), \; with \; x' \neq j \tag{6.24}$$

The state probability, at instant k, is obtained by:

$$P[X_k = x'] = \sum_{\text{all } x} p_k(x',x) p_{k-1}(x) \tag{6.25}$$

6.4.4 Results and discussion

The same videos that were in the tests of section 6.3.2 were also used here to compare place recognition accuracy for different procedures, using different visual signatures: with and without a filter based on the information about the place topology, which can be called Markov chain based filter (section 6.4.3); and with and without the simple filter (section 6.4.1).

Table 6.5: Accuracy comparison for different procedures using different visual signatures, with and without Markov chain based filter, and with and without the simple filter. Outside of the square brackets is presented the accuracy using a linear kernel and inside of the square brackets is presented the accuracy using a polynomial (second order) and a RBF kernel respectively. Inside round brackets is presented the value of the accuracy improvement when the number features detected by SIFT is used to filter non-descriptive images, those labeled with [SIFT] tag. The hLBP is the short name for LBPbyHSV-U8N.

Approach	SVM standalone	SVM + Filter	SVM + Markov
hLBP-D+SURF500	75.48 [75.22,74.78]	80.90 [77.81,79.91]	83.38 [83.76,84.72]
hLBP-IRIS	72.09 [72.40,72.22]	76.83 [78.79,75.32]	80.68 [**85.17**,80.12]
hLBP-D+HSV-GIST	72.46 [72.44,-]	77.91 [78.46,-]	83.00 [84.75,-]
HSV-GIST	65.19 [68.61, -]	73.10 [75.21, -]	81.46 [81.38, -]
hLBP-D+SURF500[SIFT]	75.06 [77.12,75.45] (-0.32)[(+1.90),(+0.33)]	79.91 [79.46,79.61] (-0.09)[(+1.65),(-0.30)]	85.10 [82.14,83.45] (+1.72)[(-1.62),(-1.27)]
hLBP-IRIS[SIFT]	77.61 [75.51,74.32] (+5.52)[(+3.07),(+2.1)]	83.23 [79.67,77.69] **(+6.40)**[(+1.21),(+2.37)]	**86.39** [85.96,83.13] (+5.71)[(+1.21),(+3.01)]
hLBP-D+HSV-GIST[SIFT]	72.54 [73.25,-] (+.08)[(+0.81),-]	78.06 [79.41,-] (+0.15)[(+0.95),-]	83.07 [85.37,-] (+.07)[(+0.62),-]

Place transition was a concept introduced to the procedure that uses a Markov chains based filter. A place transition is considered to exist when the procedure detects that the two most likely states have both a probability higher than 25%, each one, and both likely states are connected directly. When a place transition is considered to exist, the output of the procedure is the two places with the highest probability.

In these tests, another filter for non-descriptive images is introduced. This filter is based on the concept that the number of features detected in one image is related to the richness of the image description. When an image has a low number of detected features, it means that it does not supply enough information for place recognition. In this work, the SIFT feature definition procedure was considered for the filter. Therefore, when an image has less than an amount min_{SIFT} of SIFT features, it means that the visual signature obtained for this image should not be used as input for the classifier. This filter returns a binary value, which defines whether the visual signature is valid or not. The visual signatures filtered by this filter are tagged with the [SIFT] tag.

To simplify the review of the obtained results, new short identification were defined for each place. These id's are identified inside the round brackets, in the following list of places where the robots have traveled: (s-0) Ljubljana Stairs , (s-1) Ljubljana Corridor, (s-2) Ljubljana Room A, (s-3) Ljubljana WC, (s-4) Freiburg Corridor, (s-5) Freiburg Printer office, (s-6) Freiburg Room A, (s-7) Freiburg Stairs, (s-8) Freiburg WC, (s-9) FEUP I-109, (s-10) FEUP corridor, (s-11) FEUP I-108, (s-12) FEUP I-110, (s-13) FEUP i-11

Four visual signatures were tested: LBPbyHSV-U8N-D+SURF500, LBPbyHSV-U8N-IRIS, LBPbyHSV-U8N-D+HSV-GIST, HSV-GIST.

LBPbyHSV-U8N-D+SURF500 and LBPbyHSV-U8N-D+HSV-GIST visual signatures are obtained by the LBPbyHSV-U8N approach applied over a segmented image, bottom and top; then, this descriptor is concatenated with another descriptor obtained from:

- the BOF based procedure, which uses 500 feature clusters from SURF descriptor space; or
- the GIST operator applied in each channel of HSV color space (using an image of 32x32);

Visual signatures LBPbyHSV-U8N-IRIS and HSV-GIST are described in section 6.3.

Two set of tests were conducted. In the both set of tests, these visual descriptors were tested using an SVM classifier, with a linear, polynomial of second order, and RBF kernel. In both sets, the output of the SVM classifier was filtered by a simple filter and by topology information stored in the topological map. As these videos were obtained in three different universities a virtual connection between some places is required for the Markov filter, so a virtual link between the places that ends in each university video was defined in the topological map.

In the first set of tests, the [SIFT] filter was not used. In contrast, in the second set of tests, the [SIFT] filter was used and the parameter min_{SIFT} was set equal to 10. Also, in the second set of tests, only three from the first four visual signatures were used: LBPbyHSV-U8N-D+SURF500[SIFT], LBPbyHSV-U8N-IRIS[SIFT], LBPbyHSV-U8N-D+HSV-GIST[SIFT].

The results obtained in these sets of tests are summarized in table 6.5. The place recognition procedures used topology information to filter the classification provided by the classifier achieved higher accuracy in all pairs of visual signatures and SVM kernels. Figure 6.15 shows a probability distribution that is less spread through the matrix for the confusion matrix of the procedure using a Markov chain based filter. This means that the Markov filter constrains the probability to flow according to the topology of the place. These two facts prove that the inclusion of topology information into the place recognition procedure improves the accuracy of the system.

The use of a [SIFT] filter has only clearly improved the accuracy of one procedure, which has used the LBPbyHSV-U8N-IRIS approach, as is shown in table 6.5. On the LBPbyHSV-U8N-D+SURF500[SIFT] an accuracy decrease was verified. This may can be due to the fact that this visual signature encodes naturally the richness of the image with the BOF, which uses the SURF procedure with 500 clustered descriptors. Another reason for this accuracy decrease can be related to the fact that some images, with less than 10 features, are ignored which are helpful to identify a specific perspective of some place. However, it was clear that the [SIFT] filter improves significantly the procedure that uses the LBPbyHSV-U8N-IRIS approach and it should be employed.

In the context of indoor place recognition, the approach of constraining the state probability to flow according to place connections using Markov chains is novel. This approach has proved to be efficient in increasing the accuracy of the semantic localization procedure, as can be seen in table 6.5.

If the place recognition approach that is based on LBPbyHSV-U8N-IRIS[SIFT], SVM and Markov chain is considered to compare against others approaches described in state of the art, such those presented by Torralba and Murphy (2003), Wu et al. (2009), Xing and Pronobis (2010), Ranganathan (2010), and Pronobis and Jensfelt (2012). It is found that the accuracy of these

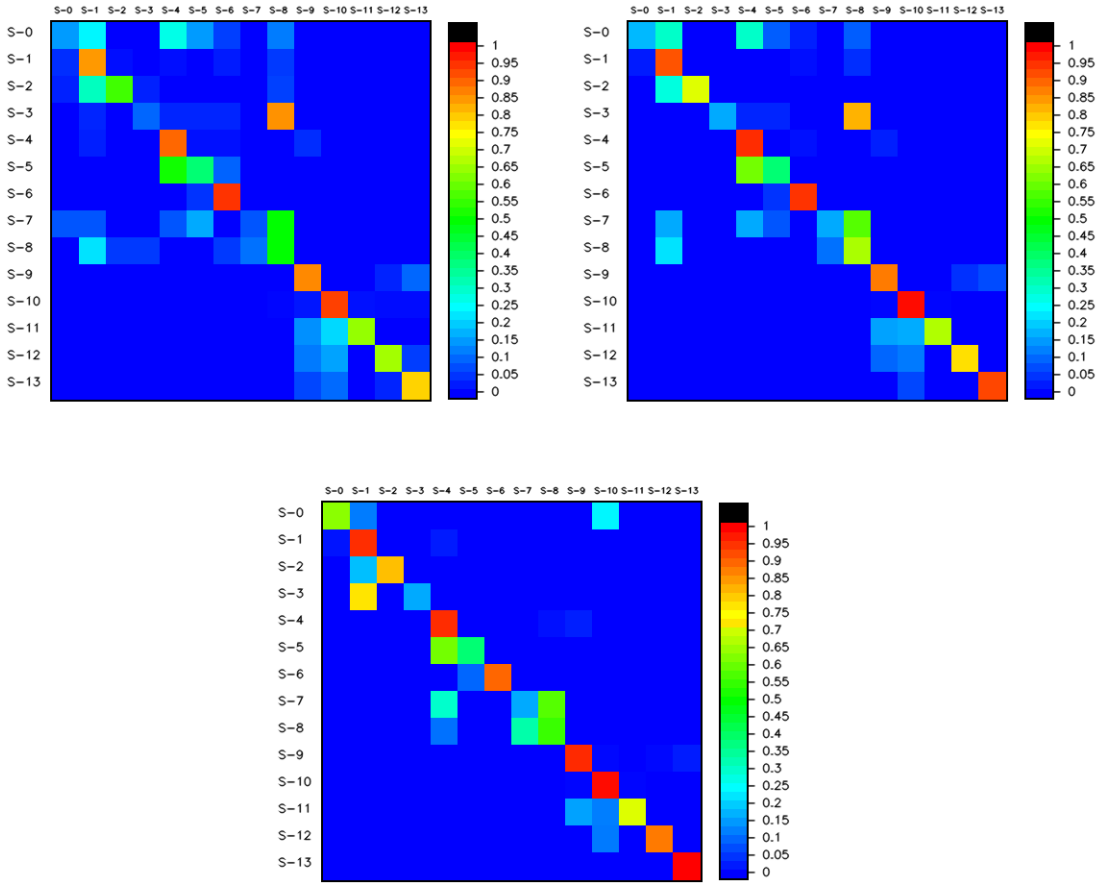

Figure 6.15: Confusion matrices obtained for the pair LBPbyHSV-U8N-IRIS and polynomial kernel. Top left: confusion matrix for the place recognition procedure without any filter. Top right: the confusion matrix for the place recognition procedure with a simpler filter. Bottom: the confusion matrix for the place recognition procedure with a Markov chain based filter.

procedures, described in the state of the art, varies between 54% to 89%. Therefore, the proposed approach is a good alternative, to those presented on the state of the art, because it has an accuracy of 86.39%, it uses a faster visual descriptor extractor which encodes the color into the visual descriptor (as was shown in tables 6.1 and 6.2), and it uses the LBP operator which is by nature robust under illumination variation.

Indeed, the proposed approach has a large margin for improvement; for example:

- the LBPbyHSV-U8N-IRIS subdivides the image using the central image point. However, if this image is divided using the vanish point concept, the descriptor will be more descriptive about ceiling, walls and floor, as illustrated in figure 6.16;

- the visual descriptor does not contain information about pixel depth. If a stereo image system or RGB-D image is available, pixel depth can be included in the visual descriptor

Figure 6.16: On the top, two image frames from the COLD dataset, with LBPbyHSV-U8N-IRIS, using a static point in the middle of the image for image splitting. On the bottom, the illustration of the same frames with the central point placed on the vanish point.

for it to be more descriptive. This information details the shape and the area of the scene observed in the image, and this important is information to helps distinguish the scenes. Pronobis and Jensfelt (2012) uses a laser range finder to obtain this information, and it shows that the accuracy of an approach using a descriptor with area or shape is higher than another that uses a global or local image descriptor, as can be seen in table 6.6. The use of a laser range finder does not satisfy the constraints imposed to this work, but the use of a stereo vision system does.

- the visual descriptor does not contain information about frame orientation. Compass is becoming a common a low-cost sensor in almost every robot; the orientation of the image taken can be an important information to enrich the visual descriptor.

Table 6.6: Classification rate obtained for each property and cue, obtained by Pronobis and Jensfelt (2012), in a dataset with 15 place classes.

Property	Cues	Classification Rate
Shape	Geometric Features	84.9
Size	Geometric Features	84.5
Appearance	CRFH	80.5
Appearance	BOF-SURF	79.9
Appearance	CRFH+BOF-SURF	84.9

6.5 Conclusions

In this section, a visual place recognition approach is presented which mimics the humans capacity to recognize a place at a glance without requiring scene interpretation. This is described as a semantic localization procedure, which is used inside of the *TopoVisualSignatures* component, formalized in the HySeLAM framework, in section 3.2. In this work, the semantic localization procedure locates the robot in a space defined by human words instead of a geometric space, as in conventional SLAM. The output of this procedure is redundant to the conventional localization provided by SLAM. This redundancy will help detect malfunctions, "kidnapping" situations, and reduce the starting time of a particle filter based SLAM.

This procedure satisfies the imposed constraints and only uses visual signatures and the augmented topological map. A descriptor based on the LBP operator was developed for this project which it uses color information to enrich the description. LBPbyHSV has proven to be the fastest global descriptor extractor, with high accuracy in the tested data.

This section presents another contribution, the use of Markov chains formalism to constrain the place probability to flow according to the place connections stored in the augmented topological map. The obtained results confirm that this approach increases system accuracy.

The datasets for system training and testing were build from videos acquired during the work for this thesis, which were then concatenated with other videos available in the COLD database, by Pronobis and Caputo (2009) in `http://www.cas.kth.se/COLD/index.php`. The videos compiled for training and testing were obtained at different moments, in different paths and weather conditions (cloudy and sunny), and with people present in the scenes. The results obtained by the approach based on LBPbyHSV-U8N-IRIS[SIFT], SVM, and Markov chains is at the same level of others shown in the literature. For example, Torralba and Murphy (2003), Wu et al. (2009), Xing and Pronobis (2010), Ranganathan (2010), and Pronobis and Jensfelt (2012) present approaches with accuracy between 54% and 88% against the 86.39% obtained by the proposed approach. Also, the proposed approach uses a faster visual descriptor extractor which encodes the color into the visual descriptor (as was shown in tables 6.1 and 6.2), and it uses the LBP operator, which is by nature more robust under illumination variation.

As future work, and considering the constraints to the visual place recognition procedure, there are several ways to increase system accuracy, as shown in the following list.

- Use of a multiclass SVM which adapts to the place probability distribution, provided by the Markov chain model. At this moment, the SVM classifier for each visual signature outputs an array of probabilities for all known classes (places); this output is used to update the transition probabilities of the Markov chain model, which is used to constrain the probability to flow according to the place connections. It is proposed that SVM, rather than testing all classes for each visual signature, should test only those classes (places) where a probability exists for the robot to be there on the moment of the visual signature acquisition.

- Considering the inclusion of pixel depth into the visual signature. Stereo image systems or RGB-D images are becoming popular in robots; these systems provide information about pixel depth, which can be used to infer the shape and area of scene. These are both powerful features for place recognition, as shown by Pronobis and Jensfelt (2012). However, Pronobis and Jensfelt (2012) proposes the use of laser range finders which conflicts with the constraint of using only cameras.

- Considering the use of Neuronal Networks as a classifier. Neuronal Networks based approaches are more complex and have more parameters to tune than SVM, but are more simple to manage new knowledge acquisition.

- Considering the use of a Graphics Processing Unit (GPU) or Field-programmable gate array (FPGA) for the LBPbyHSV-U8N extraction. This approach will boost even the descriptor extraction.

- Considering a dynamic subdivision of the image, in the LBPbyHSV-U8N-IRIS approach, by using the vanish point concept, as illustrated in figure 6.16. The descriptor will be more descriptive about ceiling, walls and floor.

- Considering an approach to detect doorway places and doors, as referred in section 4.4, this will help to detect important places where state transitions occurs.

Chapter 7

Experiments and Results (in a real test case)

The HySeLaM Framework was developed to be an extension to a conventional SLAM approach and to be easily integrable with other modules and components of the robot. Chapter 4, 5, and 6 have shown several isolated results obtained from the three *Gr2To*, *TopoMerge*, and *TopoVisualSignature* components. This chapter, on the other hand, gives an account of the overall performance of the HySeLaM framework with/when these components integrated with other components.

In section 7.1 the conceptual architecture used in real robot during the real experiments is shown. In section 7.2 the software architecture of the robot used during the experiments is described and several developed components which were required by these experiments are detailed. In section 7.3 the experiments and scenarios are described and the obtained results are shown. All source code was developed over/with ROS (Robotic operating system). The evolution and source codes of this work can be found in dos Santos (2012) (`www.hyselam.com`).

7.1 Conceptual robot architecture

Nowadays, robots are applied to services that require from them the capacity to accept high-level abstract language and the versatility to execute a large set of complex tasks. Most of the time, this set of tasks has a large amount of tasks which can be decomposed into smaller tasks. For the service robot to be robust and efficient, its architecture must be decomposed into several components which are attached to each smallest task to be performed, such as localization, mapping, object detection, route planing, mission supervision.

The list of requirements that such a complex system has to fulfill so as to function efficiently, robustly, steadily, and for long periods of time without interruptions are: modularity, robustness, efficiency, and scalability. Therefore, a distributed architecture will be privileged over a monolithic one, since this is a complex problem that must be decomposed so as to make the problem tractable.

Lets look at a generic example, which can be applied to a factory or an assistant robot. Imagine a robot that must accept the input sentence: “Robot, go to room A and pick up the box that is placed over the brown table”. This is a challenge that can be decomposed into smaller tasks, such as:

- Locating and Mapping;
- Path planning;
- Motion of the robot to a labeled place;
- Collision avoidance (dynamic and static obstacles);
- Objects/features identification/classification and tracking;
- Grasp planning and execution; and,
- Acquisition of knowledge from the human-robot interaction;

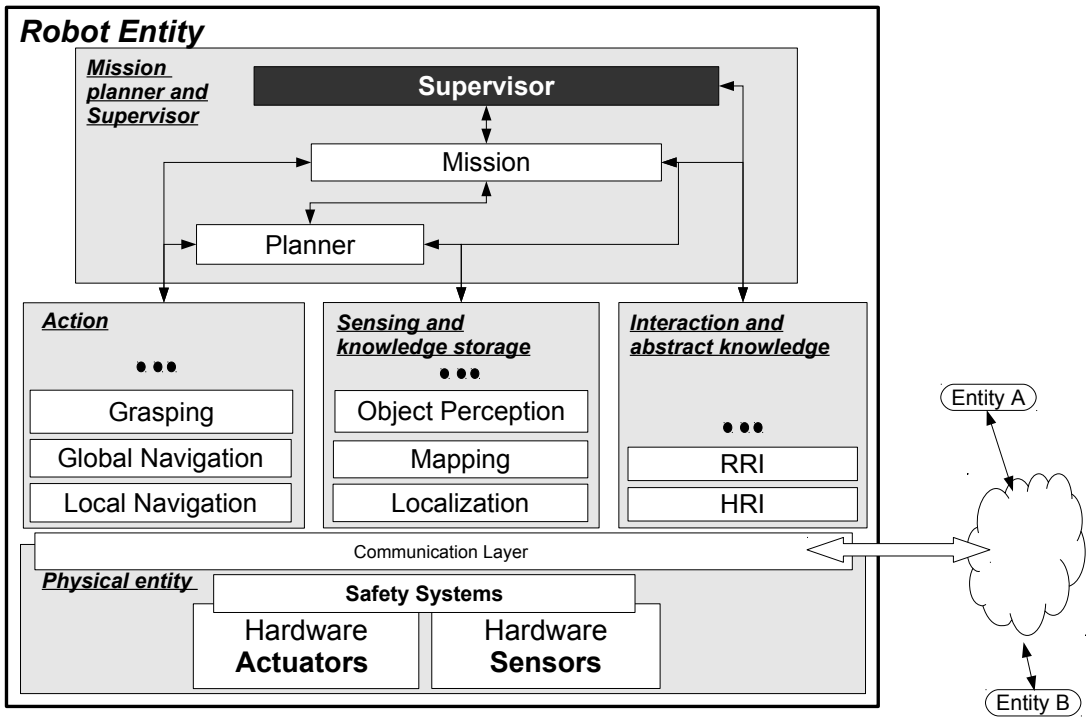

Figure 7.1: The robot entity is decomposed into five blocks: Mission, Planner and supervisor; Action; Sensing and knowledge storage; Interaction and abstract knowledge; and Physical entity.

Decomposing the problem into simple parts is a clear approach to solve it successfully. For this reason, a modular and hierarchical architecture should be adopted. This architecture can be divided into five main blocks, as depicted in figure 7.1. The physical entity block comprises all sensors, actuators and physical structure as well as the hardware abstraction layer. The interaction and abstract knowledge block comprises all tasks related to the interaction of the robot with the

external entities present in the world, such as humans and other robots. This block stores all abstract knowledge acquired by the robot, manages the relation of this abstract knowledge with the other knowledge stored in the robot, and translates the information gathered from the human interaction into the internal representation of the robot. The action block comprises all tasks that are related to the robot act into the physical world, such as moving the robot and interaction with objects. The sensing and knowledge storage block comprises all tasks that are related to:

- estimating the state of the robot in the world (Localization and orientation);
- gathering information from the sensors and creating an internal representation of the world;
- segmenting the acquired information into a set world features/objects/places.

The Mission, planner and supervisor block comprises all tasks that are related to the supervision of the mission's correct execution. This block includes all tasks that translate a task, given by the human to the robot, into a set of sequential parameterizable tasks available in the action block. This block has tasks whose purpose is to verify if the missions and tasks are reachable. For example, in this block:

- the planner: can decompose a big task, such as *Robot, go to the kitchen then go to Room B*, into simpler ones, as "GoTo x,y,z" or "GoPlace A", and optimizes the execution by organizing when is possible the order of this simpler tasks;
- the global Navigation: estimates the best trajectory taking into account the occupancy grid map, the robot state and the waypoint destination; and
- the local Navigation: executes the trajectory of Global Navigation but avoiding local obstacles sensed, using for that a local planning.

The Hybrid localization and mapping (HySeLAM) is inserted into the sensing and knowledge storage block.

7.2 Robot platform and developed components

The Produtech Robot, in figure 7.2, was one of the robots used in our experiments. This platform is an industrial AGV (Autonomous Ground Vehicle) with one traction directional wheel and two freewheels. The two SICK laser range finders (2D) are the main sensors. The main computer is a notebook PC with a quad-core Intel Atom processor. The robot software architecture follows the conceptual robot architecture, depicted in figure 7.1. This computer runs ROS fuerte over Ubuntu 12.04, as well as the SLAM algorithms and the mission controller.

The Hector SLAM package is used for SLAM and the lattice planner (also a ROS package) is used for the global planner. The HySeLAM extension and human interaction components run in another computer. This computer and ROS packages can be connected to the main computer via wireless or an Ethernet cable.

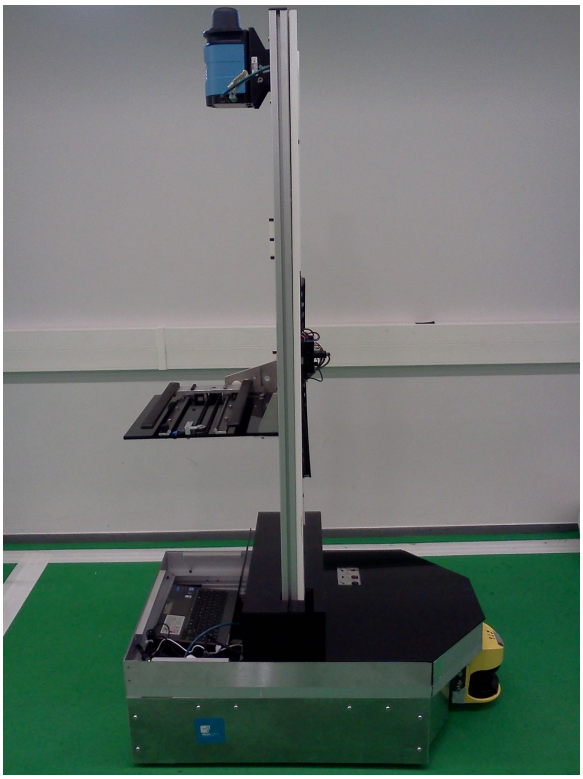

Figure 7.2: The Produtech Robot is an autonomous mobile platform. This platform was built with two freewheels and one traction directional wheel. The main sensors are two SICK laser range finders (2D). The main computer is installed with Ubuntu (Linux SO) and ROS (Robotic operating system).

The integration of the HySeLAM extension with other modules of the robot is depicted in figure 7.3. The HySeLAM subscribes an occupancy grid-map published by map server. This occupancy grid-map is managed by the Hector SLAM package, which is based on the description given by Kohlbrecher et al. (2011a). Although Hector SLAM has a good localization estimation, another component for robot localization is used. This component is based on a perfect match technique, as described by Pinto et al. (2013a). The advantage of this localization component over the Hector SLAM package is lower time consumption and CPU usage.

The HySeLAM extension provides a service over ROS which converts a human word into coordinates related to the occupancy grid-map. This service is mainly used by the action block (Global Navigation and Local Navigation tasks).

To make the connection between HySeLAM and the human possible, a human natural language interaction package was created, for the human interaction block, with minimal functionalities. This connection is required to test the HySeLAM extension, *Gr2To* and *Topomerge* in a real environment. Four components were developed: *HyFaceDetector*, *HyGtalk*, *eSpeak* and *speachRecognizer*. For the HySeLAM extension in the topological engine other components were also developed for the *TopoFeatures* and *TopoParser*, so as to solve some challenges.

The *eSpeak* is an application which converts a text file in to speach. This application is open

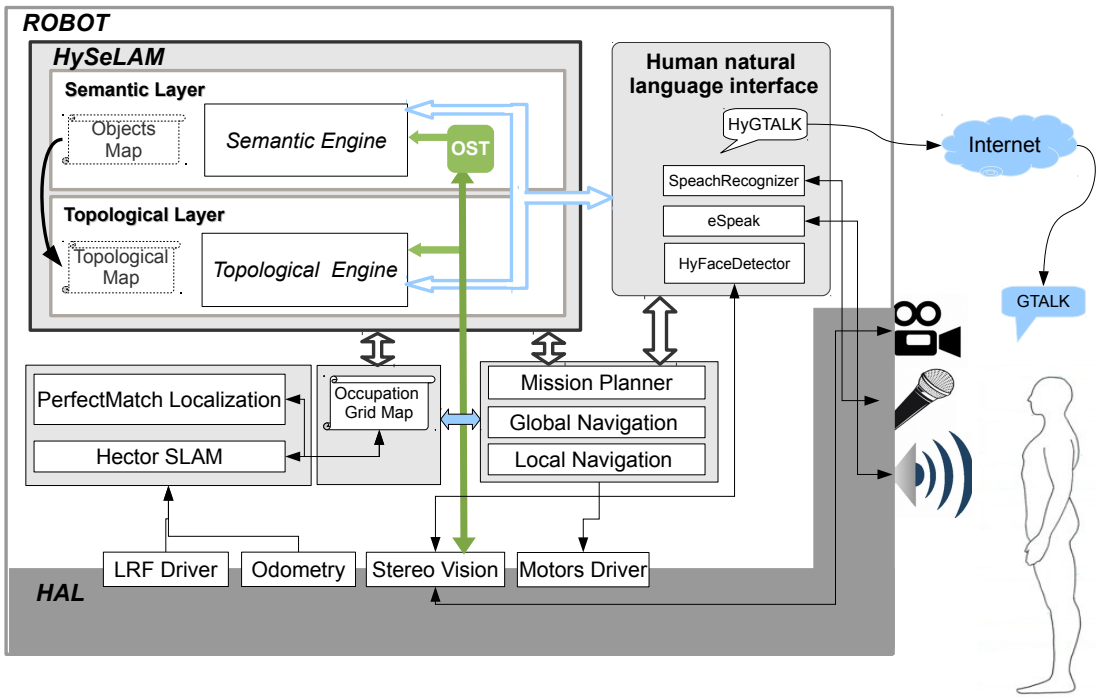

Figure 7.3: The HySeLAM extension integrated with Hector SLAM.

source and a python script is available in the ROS community, with that same name, which makes this application available as a service to all applications in the system. This python script was adapted to filter different message levels, since it can be configured to convert all received texts or to convert only the texts related to a Human-Robot interaction. Moreover, it was adapted to work (convert text into speak) when the *HyFaceDetector* component detects a human face in front of the robot.

The *speachRecognizer* is a module which converts voice into text and is based on pocketsphinx from the CMU Sphinx toolkit. To make the connection between pocketsphinx and the other modules (HySeLAM and Mission planner) a component was developed to make a bridge and to filter the output result from pocketsphinx. The pocketsphinx requires a dictionary file and language model files to work. These two files are created from a sentence corpus file which is constructed from the acceptable sentences for HySeLAM and Mission planner. However, as described earlier, in chapter 5.1, the translation of voice sentences into textual sentences is a complex task which is not completely solved. The noise present in the environment, the wrong calibration of the hardware/software, and the accent of each person makes this task a huge challenge.

Despite the acceptable accuracy of *speachRecognizer* in environments without multiple persons talking at the same time and when a good calibration of hardware/software exists, this approach has the disadvantage of requiring a dictionary of words that are acceptable by the system. When a human is describing a place, he or she is likely to use words which are not in the dictionary file. In order to make it possible to use words which are not in the dictionary file a tool based on

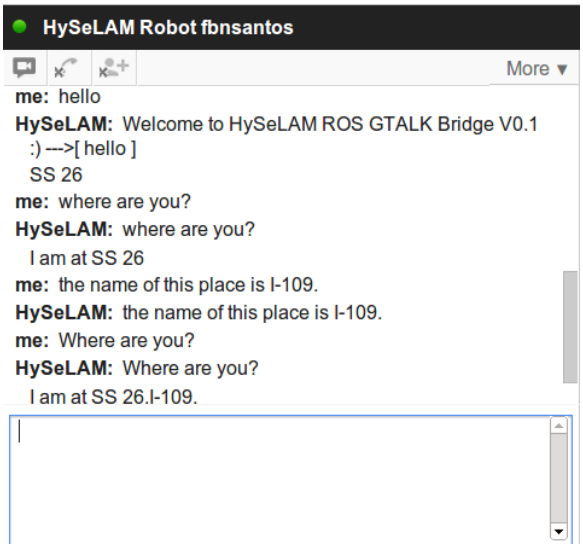

Figure 7.4: Interacting with the robot using the Gtalk messenger service from google. This version of the HyGtalk replies always with the received sentence and answer of the robot.

the messenger system was developed. This tool, *HyGtalk*, connects the robot to the Gtalk messenger service, which is provided by the Google company. When someone wants to interact with the robot, the process is as simple as adding a new user with the user name of HyGtalk component, to your gtalk account, and then use the messenger to interact with the robot, as depicted in figure 7.4. As the current implementation of the human interaction block is very simple, elaborated talks with the robot cannot be processed. Nevertheless, this tool has simplified the communication human/human-robot. When a new human word is added to the HySeLAM maps, it is also added to the sentence corpus file of*speachRecognizer*, with all possible combinations, and from this new corpus file is generated the dictionary and language model.

During the experiments with the HySeLAM extension in a real scenario, several walls made of glass were found. These walls were not detected by the Laser Range Finder, nor by the cameras, and therefore the resulting occupancy grid-map did not include them. To solve this, a wall feature addition through the HySeLAM extension was implemented. In the *TopoParser* component a new procedure was developed for the addition of virtual walls which was based on human interaction input, the current state of the robot and the occupancy grid map. This procedure is started when the *TopoParser* component receives a sentence in the form of *Robot, at your W there is a glass wall.*, where W can be one of these words $(left, right, front, back)$. From here, the *TopoParser* component detects all convex corners that lie inside the described robot side, then the convex corner that is closer to the robot is selected. From this closest corner, the previously detected nearest corner is found. However, the nearest convex corner must satisfy two conditions: it must be k meters apart from the first corner and no occupied cells can exist between the two of them. It is considered the data from occupancy grid-map instead of real laser measurements because the data from the occupancy grid-map is more stable and there is noise attenuation.

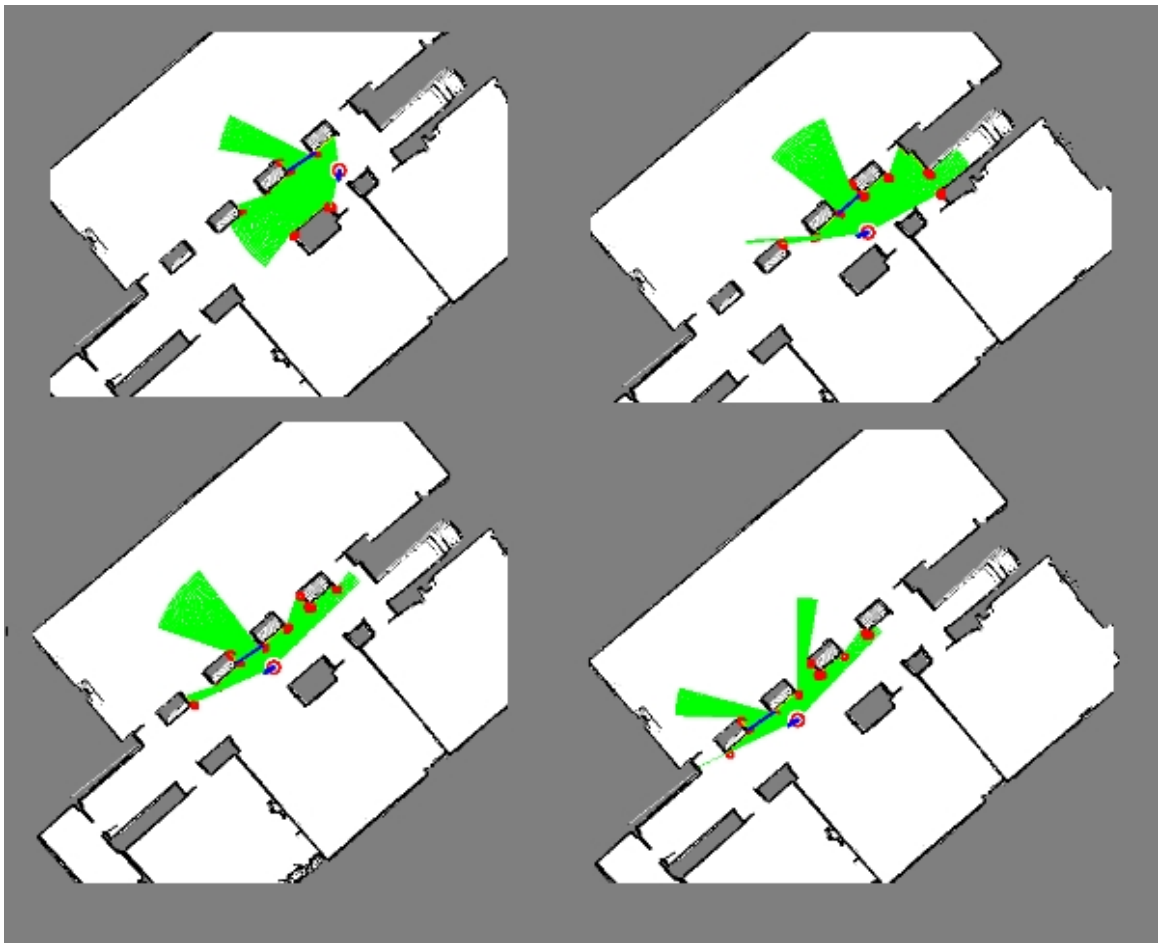

Figure 7.5: Glass wall detector based on human description. Four tests are shown. The robot is represented by a red circle with the blue line representing the robot orientation. The searched area is marked with green color. The detected corners are represented by red squares and the estimation of the glass wall position is represented by a blue line.

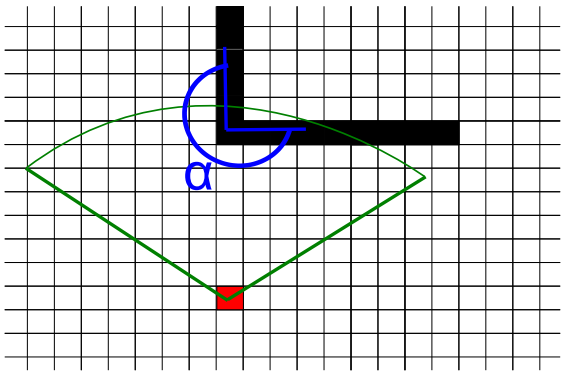

Figure 7.6: Convex corner detector.

Since the corner detector should detect only convex corners (from the robot's perspective), the

Harris corner detector, by Harris and Stephens (1988), or other approaches based on line intersection, as the one suggested by Pinto et al. (2013b), were deprecated in favor of another corner detector. This corner detector simulates a laser scan with a predefined aperture angle α_{laser} and maximum beam size $ł_{laser}$, centered in the estimated robot position and with the central beam aligned with the described orientation $(left, right, front, back)$. For each beam the nearest occupied cell is found. In this nearest cell, another beam is created with a predefined size $ł_{corner}$ which will turn 360 degrees which will be used to find the maximum angular distance α between two occupied cells, as depicted in figure 7.6. This occupied cell is considered to be a corner when the maximum angular distance satisfies the condition $\alpha > \alpha_{corner}$. During these experiments, the parameters of the algorithm have taken the following values, $\alpha_{laser} = \pi - \frac{\pi}{6} rad, l_{laser} = 6meters, l_{corner=4} cels, \alpha_{corner} = \frac{8}{12} rad$. This algorithm has an edge over the Harris corner and the line-based corner detectors because no extra procedure is necessary to detect the convexity of the corner, and no other detector is required for the lines features.

Figure 7.5 shows the results obtained at four different points in the tested scenario and when a human tells the robot *Robot, at your left there is a glass wall.*. These tests proved the efficiency of this algorithm in estimating the most likely place for the existence of a glass wall. A more complete description of the results obtained is shown in a video available by dos Santos (2012), in the results section.

After the estimation of the most likely place for the existence of a glass wall, the *TopoParser* will interact with the human as follows: *This glass wall is at* $d_{distance}$ *meters to my W and is* d_{length} *meters long. Is this information correct?* If the human says *yes* the *TopoParser* will draw this glass wall into the temporary occupancy grid-map. In contrast, if the human says *no* the *TopoParser* will interact with the human with the question:*What is the size of this glass wall?* With this new answer the *TopoParser* will search for the pair in the set of detected corners that has the distance between them equal/closer to the distance described by the human and it will repeat the first question again.

The *HyFaceDetector* has two stages: face detection and face recognition. The face detection stage is based on AdaBoost classifier cascades which use Haar-like features for facial feature detection, as suggested by Wilson and Fernandez (2006). The face recognition stage uses the ficherfaces approach. This face detection and recognition process was implemented for the robot to know if there a human is available to interact with him and if this is the right person to ask questions to.

7.3 Test scenario and Results

The test scenario was the B block of the Faculty of Engineering of the University of Porto, figure 7.7.

Figure 7.7: At the left the map of the first floor of the building at right the occupancy grid-map obtained by the robot through the Hector SLAM.

7.3.1 Learning and extracting the topological map

The robot started the mission without an occupancy grid-map. First, it was guided through the building with HectorSLAM activated. During this period, HectorSLAM updated the map by using observations obtained from the robot sensors. This map is depicted in figure 7.7.

Afterwards, the robot was guided by the human to the beginning of the corridor (the red spot in figure 7.7). Then, HectorSLAM was deactivated and the localization process based on the perfect match was activated.

At this moment the HySeLAM extension and the human-interaction block were activated. The test was performed in an area of the building where some walls are made of glass. This robot, however, does not have any sensor capable of detecting this kind of wall/material. Therefore, a human-robot interaction was started so as to guide the robot through all the places where glass walls existed. In each place, here a glass wall exists, the human described the place orientation for the existence of such glass wall.

Using the same human-robot interaction procedure, the robot was guided back to the beginning of the corridor (the red spot in figure 7.7) and the human has provided a natural textual description of the place.

This human-robot interaction is described in more detail by the following recorded conversation (between the robot and the human):

Human Robot, follow me.

The robot has followed the human while keeping a certain distance. This was done by a procedure which tracks a human face. This procedure controls the robot in order for it to

keep the human face within a certain size and position in the camera image. This procedure is used between these two sentences: "Robot, follow me" and "Robot, Stop.".

Human Stop. Robot, on your right there is a glass wall.

Robot This glass wall is at 1.43 meters to my right and it has 1.35 meters long. Is this information correct?

Human Yes.

Human Robot, follow me.

Human Robot, at your right there is a glass wall.

Robot Robot This glass wall is 1.25 meters to my right and it is 1.35 meters long. Is this information correct?

Human Yes.

Human Robot, follow me.

Human Robot, at your left there is a glass wall.

Robot This glass wall is 2.60 meters to my left and it has 3.58 meters long. Is this information correct?

Human Yes.

Human Robot, at your right there is a glass wall.

Robot Robot This glass wall is 1.30 meters to my right and it is 1.47 meters long. Is this information correct?

Human Yes.

During this last interaction the debug console was activated. In figure 7.8 the representation of the glass wall in the occupancy grid-map is shown.

Human Robot, where are you?

Robot I am at SS04.

At this moment the human uses the gtalk messenger service to talk with the robot. The robot is located at the red spot in figure 7.7.

Human Robot, you are in the corridor. On your left you have room i-109, then i-110 and i-111. You are in the corridor. On your right you have room i-108. You are in the corridor. Behind you are the stairs. You are in the corridor. In front of you is the Hall. Room i-108 is also known as robosoccer room. Room i-110 is also known as industrial robotics room. Room i-109 is also known as mobile robotics room. Finally, room i-108 has 15 by 6 meters.

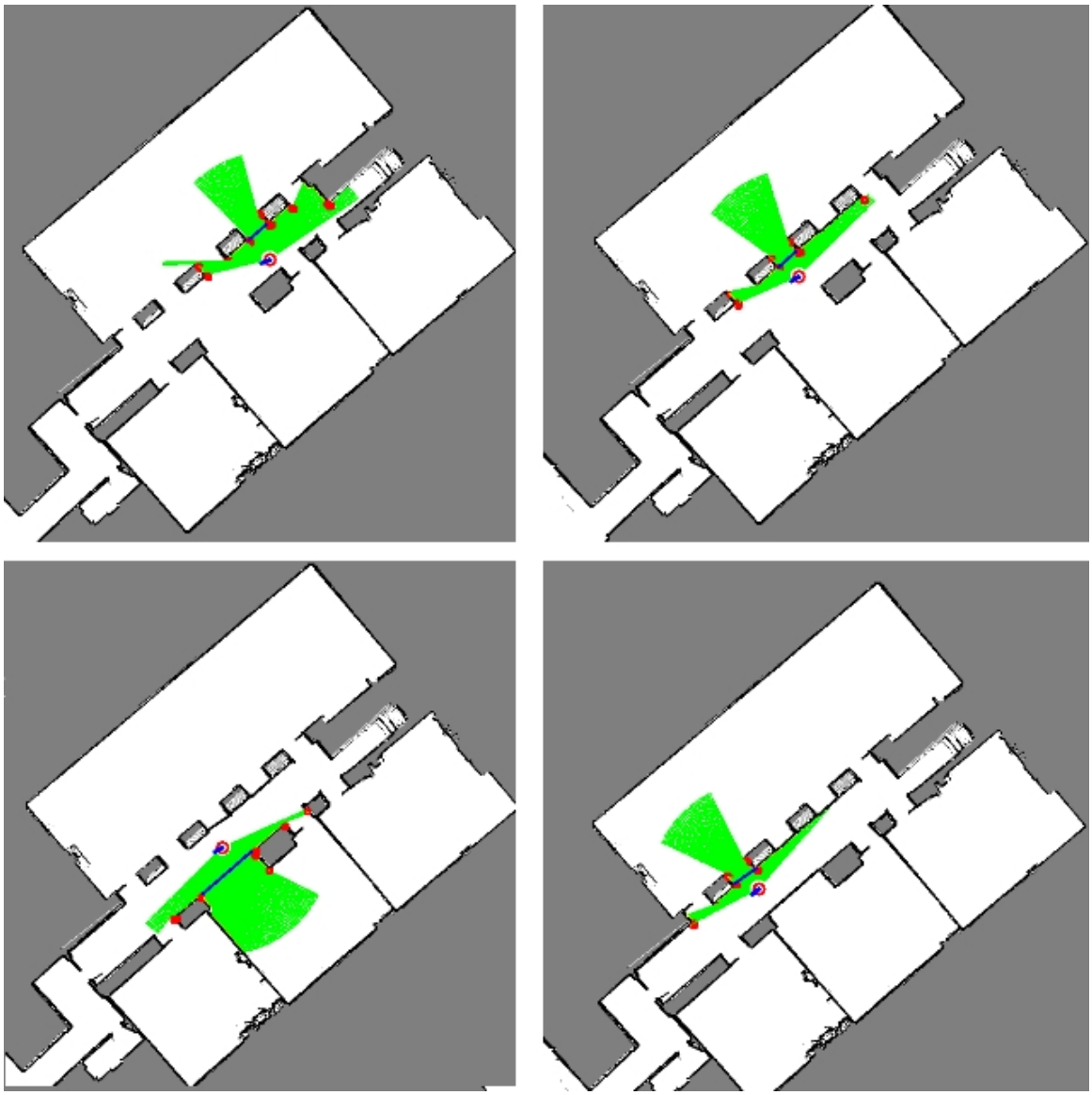

Figure 7.8: The four moments when the human tells the robot about the existence of a glass wall. The red spots are the corners detected, the area searched is in green and the blue line is the estimated location of the glass wall.

Robot Wait please, I am processing that information. Ok, it is done.

At this moment the Gr2To has constructed the augmented topological map,figure 7.9, from the occupancy grid-map. The Gr2To has taken eight seconds to process the topological map. After the topological map was extracted, HySeLAM called textitTopoParser with the NPL component and they translated the textual human description into another topological map. Using the TopoMerge component, the HySeLAM extension found a solution for the best matching. The matching quantifier obtained from the fitness function was 0.917064 (with 1 being a perfect match) and the solution was found in 2 milliseconds. The final merging

result is shown in figure 7.10.

Human Robot, where are you?

Robot I am at CC04, corridor.

Human Robot, follow me.

At this moment the human guides the robot to room I-109. (Using the human face tacking procedure.)

Human Robot, Stop. Robot, where are you?

Robot I am at SS04. Can you please confirm the name I-109 and room i-109, as the name of this place?

Human Yes.

Human Robot, go to I-110 and then go to the corridor.

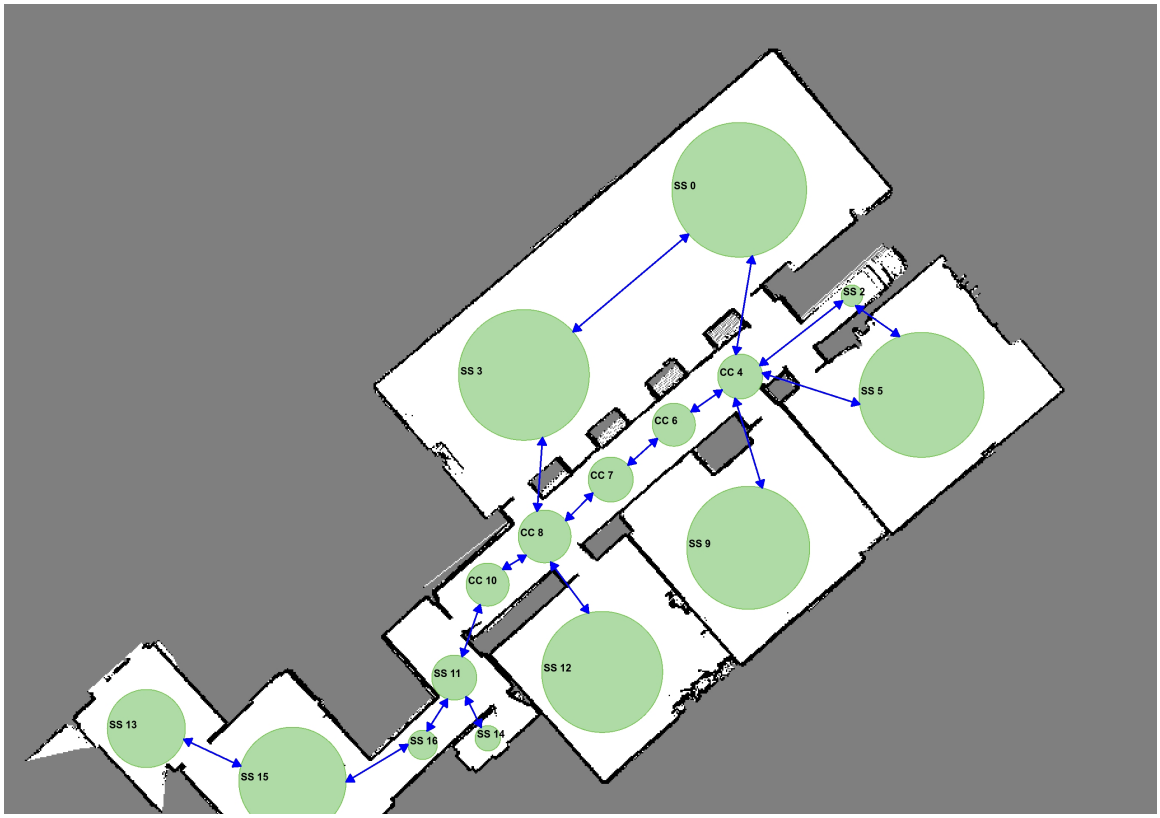

Figure 7.9: The augmented topological map obtained from the *Gr2To* algorithm, shown by the HySeLAM Graphical interface. Each place is identified by a unique code word. The group of places that can represent a single place (and a corridor) are tagged with a code word starting with CC; the other places are tagged with a code word starting with SS.

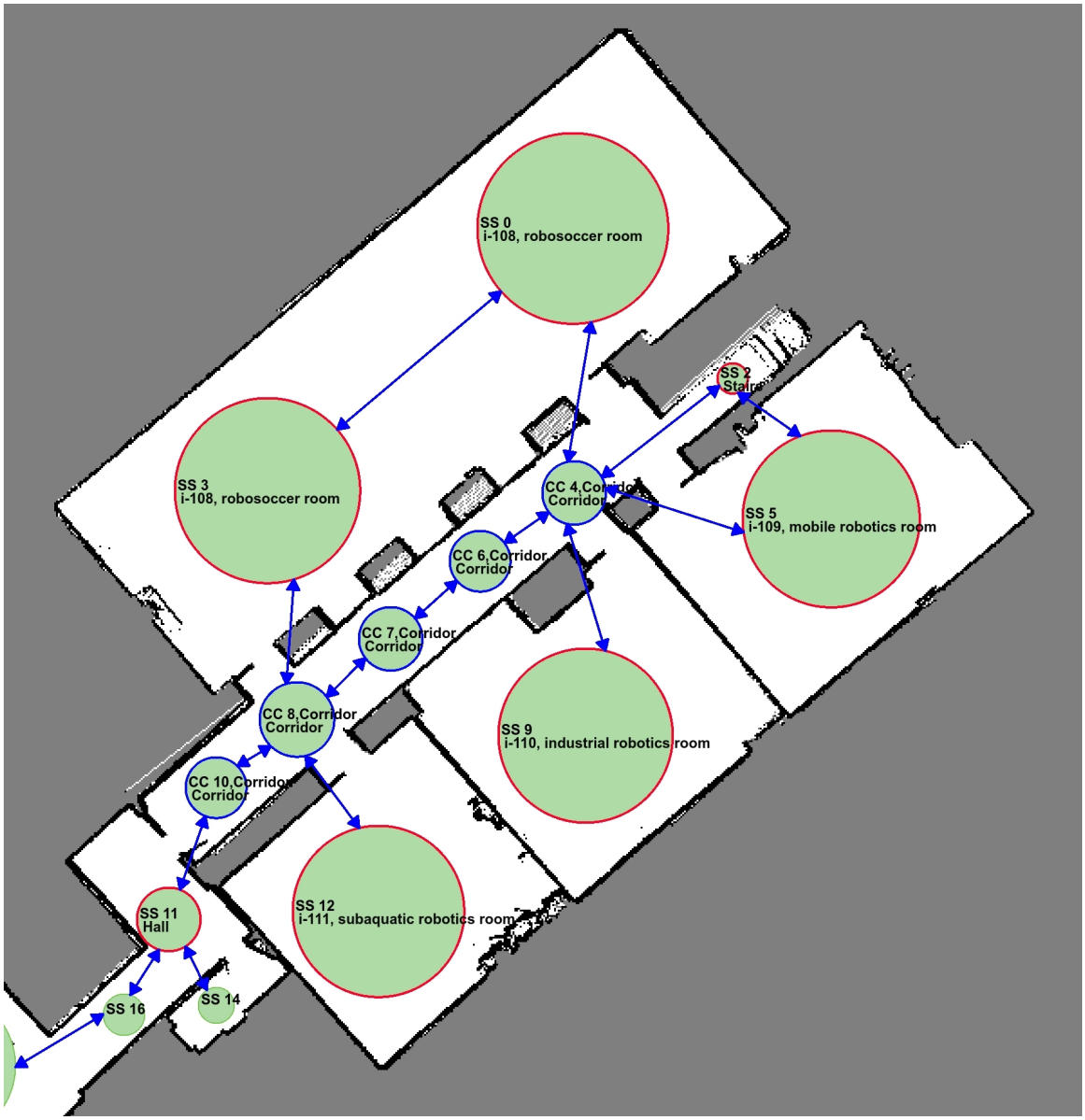

Figure 7.10: The augmented topological map merged with the description of the place provided by the human. The vertex with green borders represents places without associated human words; the vertices with blue borders are places with associated human words that were confirmed; and the vertex with red borders are places with associated human words which were not confirmed.

The last transcription of a human-robot interaction was the second test of the HySeLAM extension. In the first test, the last sentence of the place description provided by the human was omitted (*Finally, room i-108 has 15 meters by 6 meters*). In that test the *TopoMerge* considered room i-108 as the composition of one single vertex from the augmented topological map (the "SS 0") instead of considering the composition of two vertices ("SS 0" and "SS 3") as obtained in the transcript test and depicted in figure 7.10.

The correct solution is the composition of two vertices. Nevertheless, in the first test, *TopoMerge*

does not have enough information to guess that. Even, when both solutions are tested in the fitness function, the same matching quantifier value (0.701984) is obtained. However, the algorithm chooses all the time the solution which includes the lesser number of vertices.

As future work, when two solutions are found with the same matching quantifier value, the human-robot interaction block should be notified by HySeLAM so that more information from the human is requested.

TopoMerge merged the human description into the topological map but the human words inferred for each place must be verified. The unverified words are stored in each vertex as unverified words. Next, the robot in each place with unverified words and where a human-robot interaction appeared was able to ask if these were the correct name for the place.

7.3.2 Learning visual signatures

After the inference of human words that tag each place, the robot is ready to acquire and learn to correctly associate visual signatures to each place, using the knowledge stored in the augmented topological map and the place recognition procedure described in section 6.

Inspired by speculation that low-frequency magnetic fields are able to influence living organism, by Kirschvink et al. (1992), another sensor information was considered – the magnetic compass. Indeed, Wiltschko et al. (2002) and Hein et al. (2011) report the existence of a magnetic compass in one or both birds' eyes, embedded in the visual system, which is used during bird migration. There is no conclusive study on wether humans are able to sense magnetic fields. However, in these thesis experiments, the magnetic field information was considered to augment the visual signature.

As the robotic test platform had two laser range finders (LRF), several experiments that included the information provided by these sensors were conducted. These experiments were not meant to replicate those made by Pronobis et al. (2009) and Martínez Mozos et al. (2007), where the complete LRF field of view was used (360 and 180 degrees, respectively) to extract the geometric properties of the place, then used to model each place category. In contrast, these experiments were conducted to explore the advantage of using visual signatures that include depth information, provided by a stereo vision system (as the human vision). Therefore, only the LRF observations that remained in the camera's field of view were used.

Therefore, two sensors were added to the Produtech robot:

- Logitech QuickCam Ultra Vision, a 1.3Mpixel camera (V-UBH44), working with a resolution of 960x720 at 10 fps and placed in the robot at 1.60 meters from the floor;

- 9DOF Razor IMU, which incorporates three sensors - an ITG-3200 (MEMS triple-axis gyro), ADXL345 (triple-axis accelerometer), and HMC5883L (triple-axis magnetometer) - to give you nine degrees of inertial measurement. The outputs of all sensors are processed by an on-board ATmega328 and output over a serial. This IMU was attached to camera, as shown by figure 7.11.

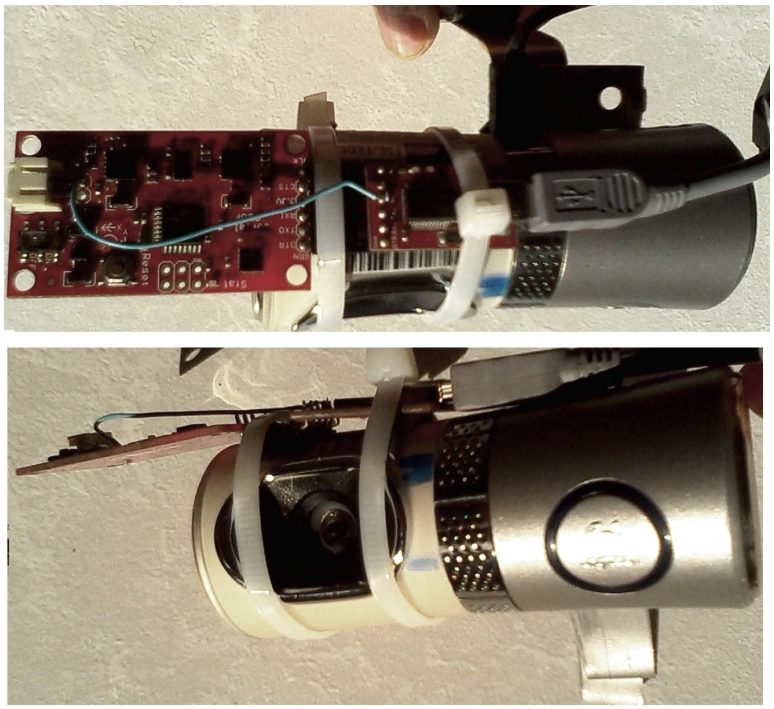

Figure 7.11: The Logitech QuickCam attached to 9DOF Razor IMU .

After the produtech robot setup with the camera and IMU. Using the augmented topological previously obtained (figure 7.10), four tours were made with the robot, at floor -1 in building I at FEUP. During these four tours, the *rosbag* tool (of ROS) was used to record five topics.

- $/topo\_output$ - The topic where the HySeLAM *topostate* component publish the semantic localization of the robot. In these tours, six place names were published: I-109; Corridor; I-110; I-111; I-108; Hall.
- $/driver\_laser\_navigation/laser\_scan$ - The topic where are published the LRF observations.
- $/driver\_laser\_security/laser\_scan$ - The topic where are published the LRF observations.
- $/imu\_data$ - The topic where are published the IMU observations and estimations.
- $/stereo/right/image\_rect\_color$ - The topic where are published rectified images gathered by the camera.

These four tours where acquired at different moments, with and without humans present in the scenario, and using slightly different paths.

The first two tours recorded data, without people present in the scene, where used to train the visual place recognition procedure. One of the recorded tour has 887 seconds long and it was acquired at midnight. The another one has 791 seconds long and was acquired 6 hours before.

Figure 7.12: In first two rows, images used during the learning stage are shown. Last row, images with people that were correctly classified with hLBP visual signature.

The other two tours recorded data, with people present in the environment, where used to test the accuracy of the visual place recognition procedure. One of these two recorded tours has 667

seconds long and it was acquired at 14 hours, after 4 days the first two tours. The another one has 689 seconds long and it was acquired at 17 hours, at same day.

Each one of these tours were done passing through these places: (s-0) FEUP I-109, (s-1) FEUP Corridor, (s-2) FEUP I-110, (s-3) FEUP I-111, (s-4) FEUP I-108, (s-5) FEUP Hall , figure 7.7. Inside of round brackets is the short name used to identify each place in the confusion matrix and ground truth graph. Figure 7.12 shows some images used to train the classifier (without people) and the images used to test the classifier (with people), which were correctly recognized by the procedure using the hLBP visual signature.

The computer used to train and test the classifier has a 2,16 GHz Intel Pentium Dual Core T4300 processor, 2GB of memory, and it is installed with a robotic operating system (ROS) (fuerte version) over the Ubuntu OS (12.04).

In order to evaluate the impact of including the depth and magnetic field information into the visual place recognition procedure, twelve types of visual signatures were built.

- hLBP - This visual signature is extracted using the LBPbyHSV-U8N-IRIS[SIFT] approach (section 6.3). This descriptor has 2360 bins.

- hLBP+ψ - This approach adds to the hLBP descriptor another bin ψ , which is filled with angular alignment to the North pole (observed by IMU compass). This descriptor has 2361 bins. During tests, 3 different weights were attributed to ψ bin, (1,2 and 10).

- hLBP+ψ+A_n - This approach adds to the hLBP+ψ descriptor another n bins, which are filled with distance values. In this approach, the camera field of view is divided into n beams. In each one of these beams is gathered all laser range measurements and the average distance is calculated for each beam. This descriptor has $2361+n$ bins.

- hLBP+[ψ]+A_n - This approach concatenates to the hLBP+A_n descriptor another 4 bins. These bins are filled with the magnitude of the magnetic field in the three axis component (x,y,z), and with full magnitude. This descriptor has $2361+n+4$ bins.

- sLBP-$(N)LRF_n$ - This approach slices the image into n columns (beams), where a average distance is calculated based on the LRF observations. In each one of these columns, the LBPbyHSV-U8N descriptor is extracted (section 6.3). Then, the sLBP-$(N)LRF_n$ descriptor is divided in 3 sections, in the first and second sections are summed the descriptors (LBPbyHSV-U8N) that are closer than 1 and 2 meters respectively, and on third section are summed the remain descriptors. This descriptor has 1416 bins $(472+472+472)$.

- sLBP-$(N)LRF_n$+ψ - This approach adds to the sLBP-$(N)LRF_n$ descriptor another bin ψ (heading). This descriptor has 1417 bins $(472+472+472+1)$.

- sLBP-$(A)LRF_n$ - This approach concatenates to the sLBP-$(N)LRF_n$ descriptor, the LBPbyHSV-U8N descriptor for all image. This descriptor has 1888 bins $(472+472+472+472)$.

- sLBP-$(A)LRF_n$+ψ - This approach concatenates to the sLBP-$(A)LRF_n$ descriptor another bin ψ (heading). This descriptor has 1889 bins $(472+472+472+472+1)$.
- sLBP-$LRF(S)_n$ - This approach concatenates to the sLBP-$(N)LRF_n$ descriptor, the uniform LBP histogram for all image (section 6.3). This descriptor has 1475 bins $(472+472+472+59)$.
- sLBP-$(S)LRF_n$+ψ - This approach concatenates to the sLBP-$(S)LRF_n$ descriptor another bin ψ (heading). This descriptor has 1476 bins $(472+472+472+59+1)$.
- sLBP-$(S)LRF_n$+$[\psi]$ - This approach concatenates to the sLBP-$(S)LRF_n$ descriptor another 4 bins. These bins are filled with the magnitude of the magnetic field in the three axis component (x,y,z), and with full magnitude. This descriptor has 1479 bins $(472+472+472+59+4)$.

The performance of these visual descriptors is summarized on table 7.1.

Table 7.1: Accuracy comparison for different procedures using depth and magnetic field information, with and without Markov chain based filter. Outside of the square brackets is presented the accuracy using a linear kernel and inside of the square brackets is presented the accuracy using a polynomial (second order) and a RBF kernel respectively.

Approach	time (ms)	standalone SVM	SVM + Markov	Learning time (s)
hLBP	35.637(0.030)	67.90 [68.68,68.16]	78.60 [82.24,**82.33**]	1172 [706,800]
hLBP+ψ				
|— {1,1 }	35.877(0.029)	69.17 [74.03,74.54]	82.33 [84.60,82.96]	911 [869,1235]
|— {1,2 }	35.863(0.017)	69.11 [73.38,74.23]	81.89 [**86.26**,83.65]	958 [768,1203]
|— {1,10}	35.943(0.055)	68.77 [70.88,70.93]	78.50 [80.52,81.91]	984 [1223,1275]
hLBP+ψ+A_1	36.262(0.030)	67.57[70.28,68.47]	81.30[82.96,81.74]	1133 [799,795]
hLBP+ψ+A_{20}	36.178(0.026)	71.44 [77.14,74.19]	86.45 [**88.21**,86.39]	970 [721,1378]
hLBP+$[\psi]$+A_{20}	36.188(0.025)	75.00 [77.40,76.89]	86.31 [86.72,86.78]	2017 [540,546]
sLBP-$LRF(N)_{20}$+ψ	30.907(0.010)	77.18 [70.77,70.62]	**86.83** [83.72,82.20]	1497 [800,753]
sLBP-$LRF(A)_{20}$	38.998(0.023)	74.25 [72.53,75.71]	86.99 [83.54,85.35]	1020 [1444,1486]
sLBP-$LRF(A)_{20}$+ψ	39.006(0.023)	77.29 [75.45,69.35]	**88.60** [85.00,78.34]	1420 [865,863]
sLBP-$LRF(S)_{20}$	39.130(0.043)	79.15 [78.21,78.63]	89.12 [88.66,89.71]	1456 [611,604]
sLBP-$LRF(S)_{20}$+ψ	39.030(0.031)	81.90 [81.23,81.38]	90.62 [91.23,91.32]	1890 [722,714]
sLBP-$LRF(S)_{20}$+$[\psi]$	38.039(0.007)	83.45 [83.66,83.50]	**92.06** [91.89,91.75]	1895 [432,439]
average	-	74.01 [74.84,74.35]	85.35 [85.81,84.95]	1332 [807,930]

These results show that it is possible to locate the robot using visual signatures without the need of mapping special features or objects. Indeed, it was shown that it is possible to locate the robot in the human word space by means of these visual signatures. Also, employing a Markov chains filter after the classifier has proved to be an efficient approach to increase visual place recognition accuracy; in average, an increase of 10% was verified.

An accuracy increase was verified when the magnetic field information was included into the visual signature. However, the magnetic field descriptor $[\psi]$ is more descriptive than ψ, because

it includes the information about visual signature orientation and also the magnetic field signature of the place. During these tests, the posture of the camera was static in terms of pitch and roll; for future work, however, posture correction using the accelerometer information should be considered.

The accuracy of visual place recognition procedure has increased when scene depth information was included into the visual signature. Nevertheless, clustering each image descriptor(LBPbyHSV-U8N) by a set of distances, as sLBP-$(S)LRF_n$, is more efficient than concatenate another descriptor with depth information, as done in hLBP+A_n. Figure 7.13 shows the confusion matrices obtained for these two approaches using the testing data. In this work a 2D depth information was considered, however as future work should be considered the use of a RGB-D camera to obtain 3D depth information.

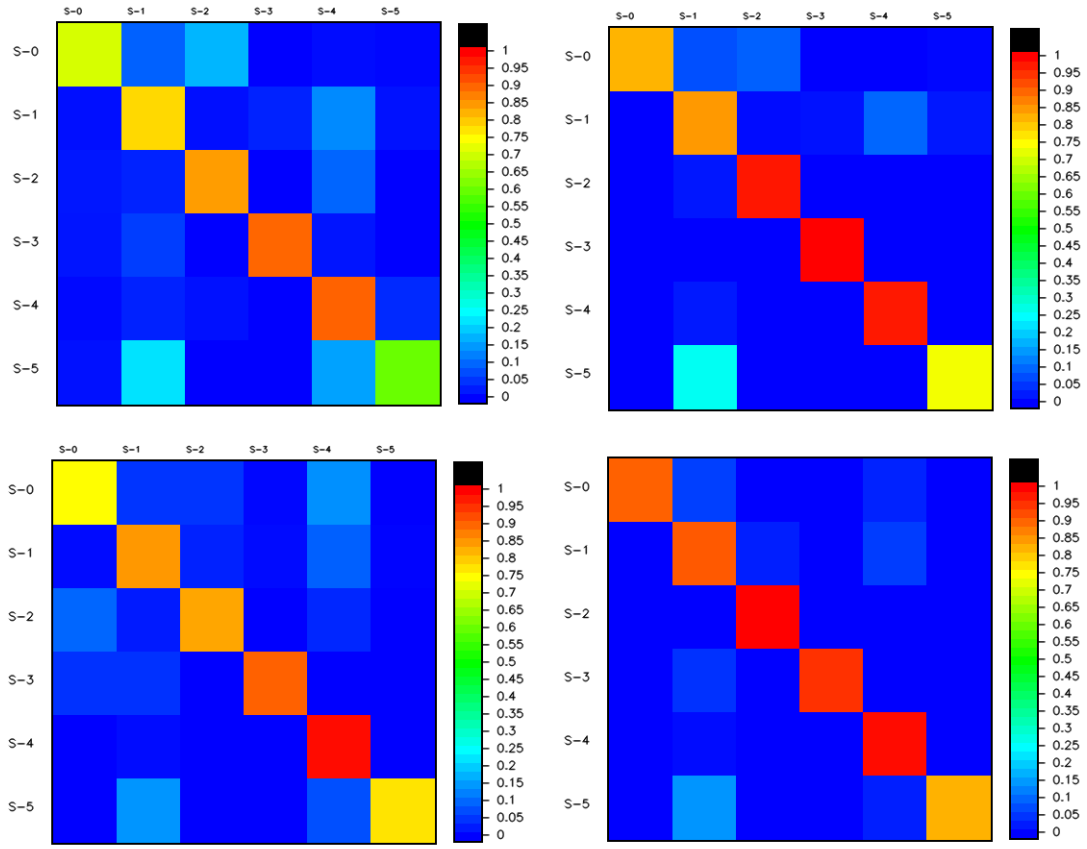

Figure 7.13: Confusion matrix comparison. On top the confusion matrices obtained for the hLBP+$[\psi]$+A_{20} approach, on the bottom, confusion matrices obtained for the for sLBP-$(S)LRF_n$+$[\psi]$ approach. At the left there is the classifier without Markov filter, at the right with Markov filter. These classifiers were using a linear kernel.

In order to help reading figure 7.13, the short identification for the places are remembered.

(s-0) FEUP I-109, (s-1) FEUP Corridor, (s-2) FEUP I-110, (s-3) FEUP I-111, (s-4) FEUP I-108, (s-5) FEUP Hall.

Videos from these and other experiments are available at www.hyselam.com (dos Santos (2012)). The source code of the implemented HySeLAM extension can also be found there.

Chapter 8

Conclusions and Future work

With this work, a step forward was taken to include by natural means (voice and natural language) the human in the localization and mapping loop of robots. This will allow the robot to understand the environment as humans do, which in turn will allow them to work and collaborate alongside us. Also, a step forward was taken in granting the robot the capacity of recognizing a place at a glance and associating the place name directly to the recognition. However, several questions and problems remain open in making it possible for the robots to interact naturally with humans and understand the environment as humans do.

This final chapter is divided in two sections. Section 8.1 presents the overall conclusion obtained from this work. Section 8.2 presents the open questions and how this work can progress in the future.

8.1 Overall conclusion

Drove by the question *How can a SLAM approach be extended in order to include the human in the mapping process?*, this work is a step further towards enabling robots to interact with people by natural means (voice and natural language) and infer and acquire knowledge about the environment from this interaction.

One of the main contributions is the proposed semantic mapping extension for SLAM. The name of this extension is HySeLAM, which stands for Hybrid Semantic Localization and Mapping.

Unlike the common approaches to the SLAM problem (chapter 2), this extension enriches the mapping (SLAM) process by including the human in it.Also, this extension creates a tighter relationship between the words and their definition in terms of sensor observation, which will simplify the task of upper levels (reason/cognition).

A distinctive feature of this work is the division of the defining, classifying and mapping process into three categories metric, places and objects. This clear separation makes sense because:

- the knowledge acquired by a SLAM approach, the occupancy grid map, is much closer to the information provided by the robot sensors; it is very geometric and not easily human

readable and communicable but it is very useful for SLAM approaches (computationally tractable);

- human splits the environment into places, which are tagged by human words, Usually a physical definition of place is more abstract than an object (generally speaking an object has a clear delimitation), so the robot should be able to gather from the human interaction and from the occupancy grid map the geometric definition of each place and the human word that tags each place; and,

- object detection and object based localization are hard tasks which requires a high computational resources. A map that stores objects present in the environment and relates them to the topological and occupancy grid maps helps these tasks to select, based on the robot location, the best features for the localization and object detection process .

This separation of the mapping process in two categories has made it possible to build a category-specific extension composed of two layers with specific components and maps. This separation has simplified the practical implementation and made it possible to divide a complex problem into smaller ones.

In synthesis, with HySeLAM the mapping process is splitted in three layers: the metric layer which is managed by a SLAM approach, the topological layer where place definition and spatial symbol grounding happens, and the object mapping layer where objects are mapped and related metrically and semantically to topological and occupancy grid-map. This makes the approach more clear, more reliable and more implementation-friendly when compared to approaches that use a conceptual map like SLAM, as in Zender et al. (2007).

Also, unlike previous works (Galindo et al. (2005);Diard and Bessi (2008); Vasudevan and Siegwart (2008); Zender et al. (2007); and Pronobis and Jensfelt (2012)), the HySeLAM framework was designed to be compatible with any SLAM approach, which works with an occupancy grid map (2D or 3D). This simplifies the use of this extension over other systems which already have its own SLAM approach, because it only requires the connection to the occupancy grid-map and the state of the robot in that map.

An important aspect of HySeLAM framework is the use of attributed graph-based maps in the two layers. Attributed graphs are well-fitted for abstract representations, which are required when the human is included in the mapping process of objects and places.

Another innovation of HySeLAM, when compared to previous works, is the inclusion of relational tags such as at, inside, in, left, right, and top, into the edges of these maps and also the strengthening of this relation. This relational tags and its strength helps to:

- define what features are good for the localization process;

- make the internal maps more human-readable and communicable;

- render the reasoning about the representation of the environment a simpler task; and

- simplify the process of merging a human description into the internal maps.

Another distinctive feature of HySeLAM appears in the augmented topological map. This augmented topological map describing places and their connections includes both visual signatures for each place and real and virtual walls in the place definition, as well as real and virtual doors in the connection. Defining real and virtual walls/doors is useful to understand whether the place boundaries are real or not, or even to store physical barriers which can be invisible to the robot sensors, like glass walls or doors. During the tests of HySeLAM this has proved to be a useful feature.

During the design of this extension was found that is required a component to infer the first version of the augmented topological map. So, in order to infer the augmented topological map from the occupancy grid-map, an approach to make a discretization of a grid-map into a topological map was formalized.

This approach is another contribution of this work and it is named Gr2To, which stands for gridmap to topological map tool. When compared to other approaches which extract the topological map from an occupancy grid-map, the segmentation of the gridmap into places performed by the Gr2To approach is much closer to the segmentation done by a human. This happens because the majority of topological maps extracted from occupancy grid-maps are used for navigation planning, which requires other rules for place segmentation.

An important feature and innovation of the proposed Gr2To approach is the augmentation of the topological map with walls and doors extracted from the occupation gridmap. This approach finds in the occupation gridmap locations that have a higher probability of containing a door, which can then be confirmed using the human-robot interaction or a visual confirmation.

In order to validate the results obtained from Gr2To, three occupancy grid maps were given to a group of people, and the number of doors and places detected by the average of people and by the Gr2To were compared. Gr2To's results were similar to human results and were obtained in an acceptable amount of time (faster than the human average).

Another important contribution of this work is an approach that solves the problem related to the question *How to merge the description of a place provided by a human into the augmentation topological map?* In order to answer this question the problem was split in two smaller questions, as follows.

- *How can a description of a place given by a human be translated into an augmented topological map?*, and
- *How can these two augmented topological maps (one described by the human and other by Gr2To) merged to form a single map?*

The answer to the first question requires a two-stage approach: *speech-to-text* and Natural Language Processing. In the literature several solutions for both stages can be found. However, the first stage is a complex problem which is not totally solved.

Regarding this, it was found that all approaches require a dictionary of words in order to work. However, in the case of robots that need to learn with humans, sometimes new words are created, in general to tag a place. Therefore, another question was raised: *How can the speech-to-text approach work without a dictionary of words?* As no sound enough answer to this question was found, and this problem does not belong to the core of HySeLAM, a deep answer to this question was postponed for future work.

The first version found in this work for these two stages – *speech-to-text* and Natural Language Processing - is based on two tools – CMU Sphinx and Nooj – which are described in sections 5.1 and 7.2.

The second question, on the other hand, was deeply explored. The two topological maps, obtained from the human description and from the Gr2to place segmentation, are two attributed graphs. The best option to merge both graphs was found to be the graph matching theory. Therefore, a fitness function is proposed in order to quantify the matching quality, and two approaches – *TopoMergExplorer* and *TopoMerg* – were proposed so as to find the best matching between the two topological maps.

The first approach, *TopoMergExplorer*, finds/looks for in all space of possible combinations for the best matching between the two topological maps. This approach is not the most/more efficient because it uses all space of combinations. However, it ensures the best matching solutions to be found and was useful to validate the fitness function and to explore the influence of each parameter of the fitness function.

One conclusion obtained from the fitness function validation was that a larger number of attributes help to disambiguate the best matching solutions. Therefore, when the robot asks for a place description from a human, it should adopt strategies to get the maximum number of details from the human so as to reduce the number of best matchings or the uncertainty associated with best matching. However, these strategies should lead the human to use details that are understood by the robot.

As previously stated, the *TopoMergExplorer* does not optimize the search space and thus a lot of time is spent to find the best solution. Since the time spent was found to be too long and thus unacceptable during the human-robot interaction, a second approach – *TopoMerg* – is proposed. *TopoMerg* is based on the tree search concept with backtracking and it searched the best matching solution for the two topological maps, by considering the constrains imposed by the attributes of the edges and vertices of graphs. The algorithm starts with a draft version of the matching solution and then adds to the matching solution the pair of vertices which most increases the matching function. This algorithm proved to be much more efficient than *TopoMergExplorer*, albeit being susceptible to choosing local maximums.

In order to reduce the uncertainty of the matching solution, it was found that the robot should adopt an active behavior like going to other places and asking a human for another description or simply ask the name of that place. This will create new constraints for the graph matching which will reduce uncertainty of matching solution.

In synthesis, this is another innovation of this work. With these approaches it is possible to merge the description of a place provided by a human into the augmentation topological map. Another important innovation is the use of spatial relation tags (left, right, back and front) in the human description as an edge attribute in the fitness function. This way the robots are able to understand the human description about the place and infer the name of each segmented place.

In this thesis work a visual place recognition approach was explored, which mimics the humans capacity to recognize a place at a glance without requiring scene interpretation. This is described as a semantic localization procedure, which is used inside of the *TopoVisualSignatures* component, formalized in the HySeLAM framework (section 3.2). This semantic localization procedure locates the robot in a space defined by human words instead of a geometric space, as in conventional SLAM. The output of this procedure is redundant to the conventional localization provided by SLAM. This redundancy will help detect malfunctions, "kidnapping" situations, and reduce the starting time of a particle filter based SLAM.

In the context of visual place recognition two main contributions were made. A new global image descriptor was proposed, with the LBPbyHSV name (section 6.3). This descriptor is based on the LBP operator, and it was developed to include color information in the way to enrich the description. LBPbyHSV has proven to be the fastest global descriptor extractor, with high accuracy in the tested data. Another contribution of this work, was the use of a filter after the SVM classifier to constrain the probability, of robot state, flow according the topological map structure. The results proves that the filter increases significantly the system accuracy, (section 6.4).

In the same context, the inclusion of scene depth and magnetic field information into the visual signature was explored, (section 7.3.2). It was concluded that the inclusion of this information increases the visual place recognition accuracy and should be explored in the future.

This HySeLAM framework and the described components were implemented into a real robot. The results obtained were promising and made it possible to included the human in the mapping process. Although only the topological layer was explored in depth, it is shown that this extension is valid, opening up a new line work that uses a bottom-up approach.

The questions that were found during this work and which remain open are described in the next section.

8.2 Future Directions

The HySeLaM extension with its components (such as Gr2To, TopoMerg, TopoVisualRecognition) is the answer given by this work to the question *How can a SLAM approach be extended in order to include the human in the mapping process?* Besides introducing the human in the mapping process, HySeLaM creates a tighter relationship between the words and their definition in terms of sensor observation, which will simplify the task of upper levels (reason/cognition).

In this work, priority was given to investigating the actual implementation of the topological layer. Therefore, investigating a real implementation of the object mapping layer must be postponed for future work. In addition to this possible line of work, this thesis has opened several questions that must be answered in order for the HySeLaM extension to reach its maximum potential and enable the existence of a human, in all the mapping process, by natural interaction. The following questions are also presented as a future direction for research.

- *How can the visual place recognition procedure be improved?*

  To answer this question, six future lines of work were identified.

  - The LBPbyHSV has proved to be the fastest global descriptor extractor, with high accuracy in the tested data. However, the lower computational complexity allows that this descriptor extractor to be easily computed by a dedicated hardware, such as GPU or FPGA, removing this task from main CPU. This will make this procedure lighter with minimal CPU cost.
  - The results obtained in section 7.3.2 proved that inclusion of scene depth information into the visual signature enrich the scene description and increases the procedure accuracy. In these tests were used a LRF and monocular camera, as future work the use of a RGB-D or stereo vision system should be considered. These two sensors supplies 3D depth information, which can be used as gray scale image where the LBP histogram can be extracted as a descriptor. Another possible way is to apply the same concept used in LBPbyHSV, the color (in this case depth) segmentation, and extract a reacher descriptor.
  - Use of a multiclass SVM which adapts to posterior place probability distribution estimation, provided by the Markov chain filter. At this moment, the SVM classifier for each visual signature outputs an array of probabilities for all known classes (places); this output is used to update the transition probabilities of the Markov chain model, which is used to constrain the probability to flow according to the place connections. It is proposed that SVM, rather than testing all classes for each visual signature, should test only those classes (places) where a probability exists for the robot to be there on the moment of the visual signature acquisition.
  - Considering the use of Neuronal Networks as a classifier. Neuronal Networks based approaches are more complex and have more parameters to tune than SVM, but are more simple to manage new knowledge acquisition.
  - Considering a dynamic subdivision of the image, in the LBPbyHSV-U8N-IRIS approach, by using the vanish point concept, as illustrated in figure 6.16. The descriptor will be more descriptive about ceiling, walls and floor.
  - Considering an approach to detect doorway places and doors, as referred in section 4.4, this will help to detect important places where state transitions occurs.

- *How can a description of a place given by a human be translated into an augmented topological map?*

  During the construction of the answer to the question *How can a description of a place given by a human be merged into an augmented topological map?*, chapter 5, another question was found: *How can a description of a place given by a human be translated into an augmented topological map?* The answer to this last question should consider that the human will usually provide the description of the place vocally. Being so, two steps are required: voice-to-text and natural language processing (NPL).

  The first step, voice-to-text recognition, is a complex problem and an active research field. The most accurate approaches for translating the voice into text require a dictionary of words. This is a problem because the places can be tagged with words that are not very common. In the present approach, this is solved by providing the description of the place through a messenger system which uses natural textual communication. Therefore, the textual description is processed by NPL and merged into the topological map. Then, these words are introduced in the dictionary so the robot can understand them. Nevertheless, this is not the desired approach for a robot that should interact with humans in normal daily life. Then, another question arises: *How can this voice-to-text step work with uncommon words?* The answer to this question does not fit in theHySeLaM extension, albeit being an answer required by it.

  The second step, the processing of natural language (done by the HySeLaM component *TopoParser*), considers in the input a well-structured textual description of the place. However, this textual description can change from person to person and thus not be uniform and sometimes have gaps in sentences due to wrong conversion (voice to text). Therefore, in the context of human-robot interaction where the human is describing the place to the robot, another question arises: *What kind of approach should/can be used to uniform this textual description?*. In the author's opinion, the process of textual description standardization and gap filling should be done by upper layers at the reasoning/cognitive level. This gives birth to another question: *How shall robot architecture evolve, from HySeLaM, to a cognitive level?*

- *How can the system deal with the uncertainty introduced by the human in the mapping process?*

  The inclusion of the human in the mapping process helps the robot understand the environment as the humans. However, this introduces some challenges. For example, as referred in the introduction, humans are not always available to help; they can also be imprecise and may not be cooperative at all times (Burgard et al. (1999)). For the robot as with humans, the level of trust in the person providing the information should be find at the reasoning/cognitive layer.

This level of trust and the level congruence between the knowledge store into HySeLaM and the input given by the human seems to be the key to find the right answer to this question. Also the use of several versions of maps with uncertainty associated or maps build with different levels of uncertainty associated to each vertex and edge seems to be a plausible direction, indeed this level of uncertainty can be useful for the reasoning/cognitive layer take decisions or verify the knowledge congruence.

- *How can be used the objects in HySeLAM localization and mapping process?*

  As said before the HySeLaM is an extension to the conventional SLAM which includes the human into the mapping process and which links words to a set of information acquired by the sensors. However, it can work as SLAM supervisor or work as concurrent process for global and relative localization. The capacity to detect features of higher level in the environment, such as objects, makes possible to use these objects as beacons/features for localization purposes. The use of objects as beacons/features is more stable than the features used in the conventional SLAM (low level features: lines, corners or visual features). The number of distinctive objects is much much more higher than the low level features. The use of this objects as beacons will help to increase the reliability of the localization estimation because the correct association of the observed beacons to the mapped one will be more stable. The localization process can be based on approaches such as EKF-SLAM or Fast-SLAM.

  Another novelty of HySeLAM is the inclusion/characterization of strength of the link (edge) between objects and objects/local. This characterization of strength, also described in this thesis as "movable", will help to decide what are the best objects used in localization process. However, this creates a new question *How can the value of this strength of the link be estimated*. Possible ways are:

  - a categorization list of movable objects which is learned in human-robot interaction;
  - a categorization list of movable objects which is learned by the history of the object position; and,
  - relating the estimation of the weight of the object to strength of the link;

  Another novelty of the HySeLAM formulation (section 3.2.3) considers that an object with another one on top is less “movable” than an equal object without any other on top of it. However, the way the strength of the primary link of main objects is affected by the “movable” value of the other objects that are related to it (right, inside) is not defined.

- *How can the robot architecture grow up from the HySeLaM until the reasoning/cognition level?*

  The HySeLAM is able to use information provided by a human to update the map of places and the map of objects. Also, it creates a tighter relationship between the words and the definition of this words in terms of sensors observation. Besides that, it supplies two maps

that are based on graphs which allows to abstract by words the information acquired by the sensors of the robot. This makes the HySeLAM very closer to the reasoning/cognition processes, figure 8.1.

Using this topological and objects maps, from HySeLAM, it is possible to build the block *Objects and place reasoning* which can use a set of rules and a conceptual map, similar to the proposed by Zender et al. (2007), to verify the congruence of the two maps. When an incongruence is detected by this block an active behavior from the robot can be activated in order to correct that. This activated behavior can be a composition of several procedures such as searching a human in the most likely place, activate the human-interaction, described the incongruence and ask to the human for a correct description of the map. Another option is the robot use other set of procedures and try capture information from the external world which helps correct the incongruence.

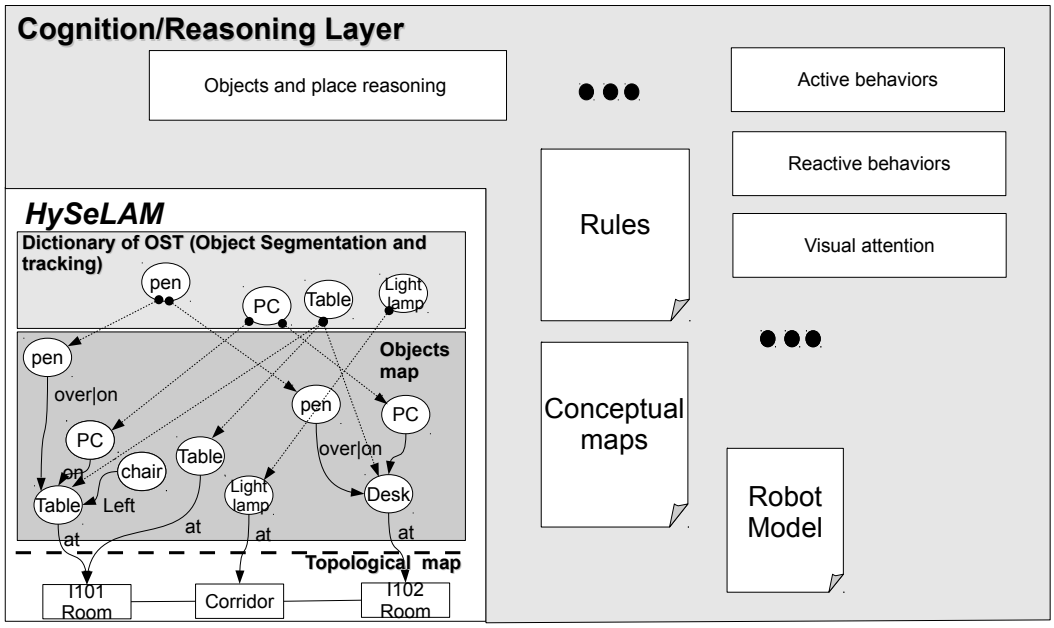

Figure 8.1: The HySeLAM maps are closer to the upper reasoning/cognition layer.

The topological layer HySeLAM is formalized with a procedure which notifies other components when it is in a place which does not have the human word that tags the place. This notification can be captured by the reactive behavior block and when the robot founds a human it can activate human-interaction and ask for the place name. This was done as draft version during the tests of the topological layer but this should formalized as a procedure of the reactive behavior block.

- *How can the the object dictionary of HySeLaM be updated from other knowledge acquired by other robots and/or available at the web?*

The mapping objects layer of HySeLAM considers a dictionary of objects which is used to described the generic objects. This generic objects are then linked to the vertices of the objects map. This link defines the generic proprieties of the object present in the environment (different color, size and so on). These proprieties and generic definition of the object is then used by a component for object detection and tracking.

The object detection and classification is hard task and still very active research field. There are several approaches that works well with one group of objects but does not work well with other group of objects. In synthesis, does not exist an approach for generic object detection and classification, this means that the object definition can change from one approach to other approach. So, the object definition found in the web or in other robots can change because they use different approaches for object detection or even different sensors. This can lead to the following problem, when we want a robot which reuses information learned by other robots, it will be required to considered different procedures for object detection which are dependent on the object definition. Another question emerges *How can these different procedures work concurrently in the same system with limited processing capacity?*

References

3DTKSite. 3DTK — The 3D Toolkit, 2013. URL http://slam6d.sourceforge.net/.

Abdel-Hakim, A. E. and Farag, A. A. CSIFT: A SIFT descriptor with color invariant characteristics. In *Computer Vision and Pattern Recognition*, pages 1978–1983. IEEE Computer Society Conference, 2006. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1640995.

Adão, T., Magalhães, L., Bessa, M., Coelho, A., Sousa, A., Rodrigues, N., Gonçalves, A., Rodrigues, R., Pereira, F., Moura, J. a., and Reis, L. P. ERAS – An Ontology-Based Tool for the Expeditious Reconstruction of Virtual Cultural Heritage Sites. In *In Atas do 20º Encontro Português de Computação Gráfica (20EPCG).*, pages 89–95, Instituto Politécnico de Viana do Castelo, Viana do Castelo, Portugal, 2012. URL http://paginas.fe.up.pt/~niadr/PUBLICATIONS/LIACC_publications_2011_12/pdf/CN13.pdf.

Aghili, F. Integrating IMU and landmark sensors for 3D SLAM and the observability analysis. *Intelligent Robots and Systems (IROS), 2010 IEEE*, pages 2025–2032, October 2010. doi: 10.1109/IROS.2010.5650359. URL http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5650359.

Alahi, A., Ortiz, R., and Vandergheynst., P. Freak: Fast retina keypoint. In *Computer Vision and Pattern Recognition (CVPR)*, pages 510–517. IEEE, 2012. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6247715.

Ancuti, C. and Bekaert, P. SIFT-CCH: Increasing the SIFT distinctness by Color Co-occurrence Histograms. In *Image and Signal Processing and Analysis, ISPA 2007. 5th International Symposium*, pages 130–135. IEEE, 2007. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4383677.

Andreopoulos, A. and Tsotsos, J. K. Active Vision for Door Localization and Door Opening using Playbot: A Computer Controlled Wheelchair for People with Mobility Impairments. In *2008 Canadian Conference on Computer and Robot Vision*, pages 3–10. IEEE, May 2008. ISBN 978-0-7695-3153-3. doi: 10.1109/CRV.2008.23. URL http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4562088.

Angeli, A., Doncieux, S., Meyer, J.-A., and Filliat, D. Incremental vision-based topological SLAM. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1031–1036. IEEE, September 2008. ISBN 978-1-4244-2057-5. doi: 10.1109/IROS.2008.4650675. URL http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4650675.

Asano, S., Maruyama, T., and Yamaguchi., Y. Performance comparison of FPGA, GPU and CPU in image processing. In *Field Programmable Logic and Applications,2009. FPL 2009.*

International Conference on. IEEE, 2009. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5272532.

Baeza-Yates, R. Challenges in the interaction of information retrieval and natural language processing. In Gelbukh, A., editor, *5th International Conference on Computational Linguistics and Intelligent Text Processing*, volume 2945 of *Lecture Notes in Computer Science*, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg. ISBN 978-3-540-21006-1. doi: 10.1007/b95558. URL http://www.springerlink.com/content/2c86t1ka6uk62087/.

Baeza-Yates, R. and Ribeiro-Neto, B. Modern information retrieval. *ACM press*, 463, 1999. URL ftp://mail.im.tku.edu.tw/seke/slide/baeza-yates/chap10_user_interfaces_and_visualization-modern_ir.pdf.

Bailey, D. *When push comes to shove: A computational model of the role of motor control in the acquisition of action verbs*. PhD thesis, University of California, 1997. URL http://ftp.icsi.berkeley.edu/~dbailey/diss.pdf.

Ballard, D. H. Generalizing the Hough transform to detect arbitrary shapes. *Pattern recognition*, (13.2):111–122, 1981. URL http://www.sciencedirect.com/science/article/pii/0031320381900091.

Banerji, S., Verma, A., and Liu., C. Novel color LBP descriptors for scene and image texture classification. In *15th International Conference on Image Processing, Computer Vision, and Pattern Recognition*, Las Vegas, Nevada, 2011. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.217.7194&rep=rep1&type=pdf.

Battiato, S., Farinella, G., Gallo, G., and Ravì, D. Spatial hierarchy of textons distributions for scene classification. *Advances in Multimedia Modeling. Springer Berlin Heidelberg*, pages 333–343, 2009. URL http://link.springer.com/chapter/10.1007/978-3-540-92892-8_35.

Bay, H., Tuytelaars, T., and Gool, L. V. Surf: Speeded up robust features. *Computer Vision–ECCV 2006, Springer Berlin Heidelberg*, pages 404–417, 2006. URL http://link.springer.com/chapter/10.1007/11744023_32.

Beetz, M., Stulp, F., Esden-Tempski, P., Fedrizzi, A., Klank, U., Kresse, I., Maldonado, A., and Ruiz, F. Generality and legibility in mobile manipulation. *Autonomous Robots*, 28(1):21–44, September 2009. ISSN 0929-5593. doi: 10.1007/s10514-009-9152-9. URL http://dl.acm.org/citation.cfm?id=1670712.1670769.

Beigi, H. *Fundamentals of speaker recognition*. Springer, 2009. ISBN 9780387775913. doi: 10.1007/978-0-387-77591-3. URL http://www.google.com/books?hl=en&lr=&id=qIMDvu3gJCQC&oi=fnd&pg=PR7&dq=Fundamentals+of+Speaker+Recognition&ots=N0HcsFwM1s&sig=GSbramnmO7zvAkCer1J02r4zPic.

Biederman, I. Aspects and extensions of a theory of human image understanding. *In Z. Pylyshyn (Ed.) Computational processes in human vision: An interdisciplinary perspective. (Pp. 370-428.) Norwood, NJ: Ablex*, 1988. URL http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Aspects+and+extensions+of+theory+of+human+image+understanding#0.

Bienenstock, E. and von der Malsburg, C. A Neural Network for Invariant Pattern Recognition. *Europhysics Letters (EPL)*, 4(1):121–126, July 1987. ISSN 0295-5075. URL http://stacks.iop.org/0295-5075/4/i=1/a=020.

Birchfield, S. T. Visual detection of lintel-occluded doors from a single image. In *2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–8. IEEE, June 2008. ISBN 978-1-4244-2339-2. doi: 10.1109/CVPRW.2008.4563142. URL http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4563142.

Bo, L., Ren, X., and Fox, D. Unsupervised feature learning for rgb-d based object recognition. *ISER, June*, 2012. URL http://researchweb.iiit.ac.in/~swagatika.panda/Linktopapers_read/bo_ren_fox_iser12.pdf.

Bohren, J., Rusu, R. B., Gil Jones, E., Marder-Eppstein, E., Pantofaru, C., Wise, M., Mosenlechner, L., Meeussen, W., and Holzer, S. Towards autonomous robotic butlers: Lessons learned with the PR2. In *2011 IEEE International Conference on Robotics and Automation*, pages 5568–5575. IEEE, May 2011. ISBN 978-1-61284-386-5. doi: 10.1109/ICRA.2011.5980058. URL http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5980058.

Bontcheva, K., Cunningham, H., Tablan, V., Maynard, D., and Hamza, O. Using GATE as an environment for teaching NLP. In *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics -*, volume 1, pages 54–62, Morristown, NJ, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1118108.1118116. URL http://dl.acm.org/citation.cfm?id=1118108.1118116.

Booij, O., Terwijn, B., Zivkovic, Z., and Krose, B. Navigation using an appearance based topological map. In *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pages 3927–3932. IEEE, April 2007. ISBN 1-4244-0602-1. doi: 10.1109/ROBOT.2007.364081. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4209699.

Borenstein, J. and L.Fen. Measurement and Correction of Systematic Odometry Errors in Mobile Robots. *IEEE Transactions on Robotics and Automation*, 12(6):869–880, 1996.

Borenstein, J., Everett, H. R., and Feng, L. *Where am I? Sensors and methods for mobile robot positioning*. University of Michigan, 1996. URL http://www-personal.umich.edu/~johannb/position.htm.

Bosch, A., Zisserman, A., and Muoz, X. Scene classification using a hybrid generative/discriminative approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(4):712–727, 2008. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4359337.

Bosch, A., Zisserman, A., and Munoz, X. Representing shape with a spatial pyramid kernel. In *Proceedings of the 6th ACM international conference on Image and video retrieval*, pages 401–408. ACM Press, 2007. URL http://dl.acm.org/citation.cfm?id=1282340.

Branavan, S. and Chen, H. Reinforcement learning for mapping instructions to actions. In *Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*. Association for Computational Linguistics, 2009. URL http://dl.acm.org/citation.cfm?id=1687892.

Brooks, A. and Bailey, T. HybridSLAM : Combining FastSLAM and EKF-SLAM for reliable mapping. In And, H. C., And, M. M., and Murphey, T. D., editors, *Algorithmic Foundation of Robotics VIII, Selected Contributions of the Eight International Workshop on the Algorithmic Foundations of Robotics, WAFR 2008, Guanajuato, Mexico, December 7-9, 2008*, pages 1–16. Springer, 2008.

Broz, F., Kose-Bagci, H., Nehaniv, C., and Dautenhahn, K. Learning behavior for a social interaction game with a childlike humanoid robot. In *Social Learning in Interactive Scenarios Workshop, Humanoids 2009*, Paris, 2009.

Brunskill, E., Kollar, T., and Roy, N. Topological Mapping Using Spec- tral Clustering and Classification. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3491–3496, 2007.

Bugmann, G., Klein, E., Lauria, S., and Kyriacou, T. Corpus-based robotics: A route instruction example. In *Proceedings of Intelligent Autonomous Systems*, pages 96–103, 2004.

Burgard, W., Cremers, A. B., Fox, D., Hähnel, D., Lakemeyer, G., Schulz, D., Steiner, W., and Thrun, S. Experiences with an interactive museum tour-guide robot. *Artificial Intelligence*, 114 (1-2):3–55, October 1999. ISSN 00043702. doi: 10.1016/S0004-3702(99)00070-3.

Burghouts, G. and Geusebroek, J. Performance evaluation of local colour invariants. *Computer Vision and Image Understanding*, 113(1):48–62, 2009. URL http://www.sciencedirect.com/science/article/pii/S1077314208001008.

Buschka, P. and Saffiotti, A. A Virtual Sensor for Room Detection. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 637–642, 2002.

Buschka, P. and Saffiotti, R. Some Notes on the Use of Hybrid Maps for Mobile Robots. In *IN Proceedings of the 8th International Conferences on Intelligent Autonomous Systems (IAS) 547–556*, pages 547 – 556, 2004. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.62.4227.

Byeong-Soon, R. and Hyun Seung, Y. Integration of reactive behaviors and enhanced topological map for robust mobile robot navigation. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 29(5):474–485, 1999. ISSN 10834427. doi: 10.1109/3468.784174. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=784174.

Cadena, C., Galvez-Lopez, D., Ramos, F., Tardos, J. D., and Neira, J. Robust place recognition with stereo cameras. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5182–5189. IEEE, October 2010. ISBN 978-1-4244-6674-0. doi: 10.1109/IROS.2010.5650234. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5650234.

Caetano, T. S., Caelli, T., and Barone., D. A. C. Graphical models for graph matching. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on.*, volume 2. IEEE, 2004. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1315201.

Caetano, T. and McAuley, J. Learning graph matching. *Pattern Analysis and Machine Intelligence, IEEE Transactions*, 31.6:1048–1058, 2009. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4770108.

Calonder, M., Lepetit, V., Strecha, C., and Fua, P. Brief: Binary robust independent elementary features. In *Computer Vision–ECCV 2010*, pages 778–792. Springer Berlin Heidelberg, 2010. URL http://link.springer.com/chapter/10.1007/978-3-642-15561-1_56.

Canny, J. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions*, (6):679–698, 1986. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4767851.

Cao, Y., Wang, C., Li, Z., Zhang, L., and Zhang, L. Spatial-bag-of-features. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3352–3359. IEEE, June 2010. ISBN 978-1-4244-6984-0. doi: 10.1109/CVPR.2010.5540021. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5540021.

Cassandras, Christos G., Lafortune, S. *Introduction to Discrete Event Systems*. 2008. ISBN 978-0-387-33332-8. URL http://www.springer.com/engineering/robotics/book/978-0-387-33332-8.

Chang, C.-C. and Lin, C.-J. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(2):27, 2011. URL http://dl.acm.org/citation.cfm?id=1961199.

Chang, C., Siagian, C., and Itti, L. Mobile robot vision navigation & localization using gist and saliency. *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on. IEEE*, 2010. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5649136.

Chang, F. and Pei, S. Color constancy via chromaticity neutralization: From single to multiple illuminants. In *Circuits and Systems (ISCAS), 2013 IEEE International Symposium*, pages 2808–2811. IEEE, 2013. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6572462.

Chang, P. and Krumm, J. Object recognition with color cooccurrence histograms. In *Computer Vision and Pattern Recognition,IEEE Computer Society Conference*. IEEE, 1999. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=784727.

Chatila, R. and Laumond, J. Position referencing and consistent world modeling for mobile robots. In *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, volume 2, pages 138–145. Institute of Electrical and Electronics Engineers. doi: 10.1109/ROBOT.1985.1087373. URL http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1087373.

Chen, Z., Li, Y., and Birchfield, S. T. Visual detection of lintel-occluded doors by integrating multiple cues using a data-driven Markov chain Monte Carlo process. *Robotics and Autonomous Systems*, 59(11):966–976, 2011. URL http://apps.webofknowledge.com/full_record.do?product=UA&search_mode=GeneralSearch&qid=1&SID=V2nG5lAgpL4bp54fdiC&page=1&doc=2&cacheurlFromRightClick=no.

Choi, J., Choi, M., and Chung, W. K. Incremental topological modeling using sonar gridmap in home environment. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3582–3587. IEEE, October 2009. ISBN 978-1-4244-3803-7. doi: 10.1109/IROS.2009.5354247. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5354247.

Choi, J., Choi, M., Nam, S. Y., and Chung, W. K. Autonomous topological modeling of a home environment and topological localization using a sonar grid map. *Autonomous Robots*, 30(4): 351–368, March 2011. ISSN 0929-5593. doi: 10.1007/s10514-011-9223-6. URL http://www.springerlink.com/content/m12548156847ntl6/.

chul woo Kang and chan gook Park. Attitude Estimation with Accelerometers and Gyros Using Fuzzy Tuned Kalman Filter. In *proceedings of the european control conference 2009*, pages 3713–3718, 2009. ISBN 9789633113691.

Civera, J., Galvez-Lopez, D., Riazuelo, L., Tardos, J. D., and Montiel, J. M. M. Towards semantic SLAM using a monocular camera. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1277–1284. IEEE, September 2011. ISBN 978-1-61284-456-5. doi: 10.1109/IROS.2011.6094648. URL http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6094648&contentType=Conference+Publications&searchField=Search_All&queryText=semantic+slam.

Conte, D., Foggia, P., Sansone, C., and Vento, M. Thirty years of graph matching in pattern recognition. *International journal of pattern recognition and artificial intelligence*, 18(03):265–298, 2004. URL http://www.worldscientific.com/doi/abs/10.1142/S0218001404003228.

Cordella, L. P., Foggia, P., Sansone, C., and Vento, M. An improved algorithm for matching large graphs. *3rd IAPR-TC15 workshop on graph-based representations in pattern recognition*, pages 149–159, 2001. URL http://pdf.aminer.org/000/348/645/efficient_algorithms_for_matching%_attributed_graphs_and_function_described_graphs.pdf.

Cristianini, N. and Shawe-Taylor, J. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000. ISBN 9780521780193. doi: http://dx.doi.org/10.1017/CBO9780511801389. URL http://www.google.com/books?hl=pt-PT&lr=&id=_PXJn_cxv0AC&oi=fnd&pg=PR9&dq=An+introduction+to+support+vector+ma-+chines+and+other+kernel-based+learning+methods&ots=xQUj1I-u1g&sig=-NoQpF53giDugyEahStgac4Prmc.

Cummins, M. and Newman, P. FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance. *The International Journal of Robotics Research*, 27(6):647–665, June 2008. ISSN 0278-3649. doi: 10.1177/0278364908090961. URL http://ijr.sagepub.com/content/27/6/647.short.

Cummins, M. and Newman, P. Fab-map: Appearance-based place recognition and mapping using a learned visual vocabulary model. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 3–10, 2010. URL http://machinelearning.wustl.edu/mlpapers/paper_files/icml2010_CumminsN10.pdf.

Dalal, N. and Triggs, B. Histograms of oriented gradients for human detection. *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference*, 1, 2005. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1467360.

Davison, A. J. and Murray., D. W. Simultaneous localization and map-building using active vision. *Pattern Analysis and Machine Intelligence, IEEE Transactions*, 24.7:865–880, 2002. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1017615.

Dellaert, F. and Bruemmer, D. Semantic SLAM for Collaborative Cognitive Workspaces. In *AAAI Fall Symposium Series 2004: Workshop on The Interaction of Cognitive Science and Robotics: From Interfaces to Intelligence*, 2004.

Diard, J. and Bessi, P. Bayesian Maps : Probabilistic and Hierarchical Models for Mobile Robot Navigation. *Star*, pages 153–175, 2008.

Dijkman, R., Dumas, M., and García-Bañuelos, L. Graph matching algorithms for business process model similarity search. *Process Management. Springer Berlin Heidelberg*, pages 48–63, 2009. URL http://link.springer.com/chapter/10.1007/978-3-642-03848-8_5.

Dirk, H., Burgard, W., Fox, D., and Thrun, S. An Efficient FastSLAM Algorithm for Generating Maps of Large-Scale Cyclic Environments from Raw Laser Range Measurements. In *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on. Vol. 1. IEEE*, 2003.

Dissanayake, M., Newman, P., Clark, S., Durrant-Whyte, H., and Csorba, M. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241, June 2001. ISSN 1042296X. doi: 10.1109/70.938381. URL http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=938381.

dos Santos, F. N. HySeLAM - Hybrid Semantic Localization and Mapping, 2012. URL hyselam.fbnsantos.com.

Duan, S., Zhang, J., Roe, P., and Towsey, M. A survey of tagging techniques for music, speech and environmental sound. *Artificial Intelligence Review*, October 2012. ISSN 0269-2821. doi: 10.1007/s10462-012-9362-y. URL http://link.springer.com/10.1007/s10462-012-9362-y.

Duchenne, O., Bach, F., Kweon, I. S., and Ponce, J. A tensor-based algorithm for high-order graph matching. *Pattern Analysis and Machine Intelligence, IEEE Transactions*, 33(12):2383–2395, 2011. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5871640.

Duckett, T. and Nehmzow, U. Exploration of unknown environments using a compass, topological map and neural network. In *Proceedings 1999 IEEE International Symposium on Computational Intelligence in Robotics and Automation. CIRA'99 (Cat. No.99EX375)*, pages 312–317. IEEE, 1999. ISBN 0-7803-5806-6. doi: 10.1109/CIRA.1999.810067. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=810067.

Durrant-Whyte, H. and Bailey, T. Simultaneous localization and mapping: part I. *IEEE Robotics & Automation Magazine*, 13(2):99–110, June 2006. ISSN 1070-9932. doi: 10.1109/MRA.2006.1638022. URL http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1638022.

Dzifcak, J., Scheutz, M., Baral, C., and Schermerhorn, P. What to do and how to do it: Translating natural language directives into temporal and dynamic logic representation for goal management and action execution." In Robotics and Automation. In *Robotics and Automation, ICRA'09. IEEE International Conference*, pages 4163–4168, 2009.

Ebner, M. How does the brain arrive at a color constant descriptor? *Advances in Brain, Vision, and Artificial Intelligence*, 2007a. URL http://link.springer.com/chapter/10.1007/978-3-540-75555-5_9.

Ebner, M. Why Color Constancy Improves for Moving Objects. *Biosignals*, pages 193–198, 2012. URL http://stubber.math-inf.uni-greifswald.de/~ebner/resources/uniG/movingCCconf.pdf.

Ebner, M. *Color constancy*. Wiley - IS&T Series in imaging science and technology, 2007b. ISBN 9780470058299. URL http://www.google.com/books?hl=pt-PT&lr=&id=WVKJST7zE8cC&oi=fnd&pg=PR7&dq=M.+Ebner,+Color+Constancy.+Hoboken,+NJ:+Wiley,+2007&ots=287mp1Fkpp&sig=oa9H6lTixc5YUeGUNUtoWFftPZk.

Ebner, M. Color constancy based on local space average color. *Machine Vision and Applications*, 20(5):283–301, 2009. URL http://link.springer.com/article/10.1007/s00138-008-0126-2.

Elfes, A. *Occupancy grids: a probabilistic framework for robot perception and navigation*. Phd, Carnegie Mellon University, 1989. URL http://www.citeulike.org/user/syberspaz/article/2501955.

Eliazar, A. and Parr, R. DP-SLAM: Fast, Robust Simultaneous Localization and Mapping Without Predetermined Landmarks. In *International Conference on Artificial Intelligence (IJCAI)*, volume 3, 2003.

Endres, F., Hess, J., Engelhard, N., Sturm, J., Cremers, D., and Burgard, W. An evaluation of the RGB-D SLAM system. *Robotics and Automation (ICRA), 2012 IEEE International Conference*, pages 1691–1696, 2012. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6225199.

Endres, F. RGBDSLAM site. URL http://www.informatik.uni-freiburg.de/~endres/rgbdslam.html.

Escrig, T. What are the benefits of Artificial Intelligence in robotics? URL http://robohub.org/what-are-the-benefits-of-artificial-intelligence-in-robotics/.

Eshera, M. A. and Fu, K.-S. An Image Understanding System Using Attributed Symbolic Representation and Inexact Graph-Matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(5):604–618, September 1986. ISSN 0162-8828. URL http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=4767835.

Espinace, P., Kollar, T., Roy, N., and Soto, A. Indoor scene recognition by a mobile robot through adaptive object detection. *Robotics and Autonomous Systems*, 61(9):932–947, 2013. URL http://www.sciencedirect.com/science/article/pii/S0921889013000821.

Fabrizi, E. and Saffiotti, A. Extracting topology-based maps from gridmaps. In *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, volume 3, pages 2972–2978. IEEE. ISBN 0-7803-5886-4. doi: 10.1109/ROBOT.2000.846479. URL http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=846479.

Fan, P., Men, A., Chen, M., and Yang, B. Color-SURF: A surf descriptor with local kernel color histograms. In *Network Infrastructure and Digital Content, NIDC 2009. IEEE International Conference*, pages 726–730, 2009. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5360809.

Figueira, D., Lopes, M., Ventura, R., and Ruesch, J. From Pixels to Objects : Enabling a spatial model for humanoid social robots. In *ICRA 2009*, 2009.

Fischler, M. and Elschlager, R. The representation and matching of pictorial structures. *Computers, IEEE Transactions on*, 1973. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1672195.

Forsyth, D. A. and Ponce, J. *Computer vision: a modern approach*. Prentice Hall Professional Technical Reference, 2002. ISBN 0130851981. URL http://www.inria.fr/centre/paris-rocquencourt/actualites/computer-vision-a-modern-approach.

Galindo, C., Saffiotti, A., Coradeschi, S., Buschka, P., Fernandez-Madrigal, J., and Gonzalez, J. *Multi-hierarchical semantic maps for mobile robotics*. IEEE, 2005. ISBN 0-7803-8912-3. doi: 10.1109/IROS.2005.1545511. URL http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1545511.

Galindo, C., Fern, J.-a., and Gonz, J. Robot Task Planning using Semantic Maps. *Robotics and Autonomous Systems*, 56(2008):955–966, 2008.

Garrette, D. and Klein, E. An extensible toolkit for computational semantics. In *Eighth International Conference on Computational Semantics*, pages 116–127, January 2009. ISBN 978-90-74029-34-6. URL http://dl.acm.org/citation.cfm?id=1693756.1693770.

Ge, R. and Mooney., R. J. A statistical semantic parser that integrates syntax and semantics. *Proceedings of the Ninth Conference on Computational Natural Language Learning. Association for Computational Linguistics*, 2005. URL http://dl.acm.org/citation.cfm?id=1706546.

Gijsenij, A., Gevers, T., and Weijer, J. V. D. Computational color constancy: Survey and experiments. *Image Processing, IEEE Transactions*, 20(9):2475–2489, 2011. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5719167.

Gijsenij, A., Lu, R., and Gevers, T. Color constancy for multiple light sources. *Image Processing, IEEE Transactions*, 21(2):697–707, 2012. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5986707.

Giugno, R. and Shasha, D. Graphgrep: A fast and universal method for querying graphs. *Pattern Recognition, 2002. Proceedings. 16th International Conference on. Vol. 2. IEEE,*, 2002. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1048250.

Gold, S. and Rangarajan, A. A graduated assignment algorithm for graph matching. *Pattern Analysis and Machine Intelligence, IEEE Transactions*, 18(4):377–388, 1996. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=491619.

Grauman, K. and Leibe, B. Visual Object Recognition. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 5(2):1–181, April 2011a. ISSN 1939-4608. doi: 10.2200/S00332ED1V01Y201103AIM011. URL http://www.morganclaypool.com/doi/abs/10.2200/S00332ED1V01Y201103AIM011.

Grauman, K. and Leibe, B. *Visual Object Recognition*. Morgan and Claypool Publishers, 2011b. ISBN 1598299689. URL http://books.google.com/books?id=lAQGBvdm3UsC&pgis=1.

Grisetti, G. and Kummerle, R. Hierarchical optimization on manifolds for online 2D and 3D mapping. *Robotics and Automation (ICRA), 2010 IEEE International Conference on. IEEE,*, 2010. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5509407.

Grisetti, G., Stachniss, C., and Burgard, W. Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters. *IEEE Transactions on Robotics*, 23(1):34–46, February 2007. ISSN 1552-3098. doi: 10.1109/TRO.2006.889486. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4084563.

Gu, S. and Chen, Q. Building topological map using omnidirectional image matching. In *Proceedings 2011 International Conference on System Science and Engineering*, pages 282–287. IEEE, June 2011. ISBN 978-1-61284-351-3. doi: 10.1109/ICSSE.2011.5961914. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5961914.

Hadas, K.-G., Fainekos, G. E., and Pappas., G. J. Translating structured english to robot controllers. *Advanced Robotics*, 22(12):1343–1359, 2008.

Hahnel, D. and Burgard, W. An efficient FastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements. *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on. Vol. 1. IEEE*, 2003. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1250629.

Harnad, S. The symbol grounding problem. *Physica D: Nonlinear Phenomena*, 42(1-3):335–346, June 1990. ISSN 01672789. doi: 10.1016/0167-2789(90)90087-6. URL http://dx.doi.org/10.1016/0167-2789(90)90087-6.

Harris, C. and Stephens, M. A combined corner and edge detector. *Alvey vision conference*, 15:50, 1988. URL http://csce.uark.edu/~jgauch/library/Features/Harris.1988.pdf.

He, X., Zemel, R. S., and Mnih, V. Topological map learning from outdoor image sequences. *Journal of Field Robotics*, 23(11-12):1091–1104, November 2006. ISSN 15564959. doi: 10.1002/rob.20170. URL http://doi.wiley.com/10.1002/rob.20170.

Hein, C. M., Engels, S., Kishkinev, D., and Mouritsen, H. Robins have a magnetic compass in both eyes. *Nature*, 471(7340):E11–2; discussion E12–3, March 2011. ISSN 1476-4687. doi: 10.1038/nature09875. URL http://dx.doi.org/10.1038/nature09875.

Hijikata, S., Terabayashi, K., and Umeda, K. *A simple indoor self-localization system using infrared LEDs*. IEEE, June 2009. ISBN 978-1-4244-6313-8. doi: 10.1109/INSS.2009.5409955. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5409955.

Hordley, S. D. Scene illuminant estimation: past, present, and future. *Color Research & Application*, 31(4):303–314, 2006. URL http://onlinelibrary.wiley.com/doi/10.1002/col.20226/abstract.

Horn, B. K. P. *Robot Vision*. MIT Press, mit electr edition, 1986. ISBN 0-262-08159-8. URL http://www.google.com/books?hl=pt-PT&lr=&id=jpX9Lrxn58MC&oi=fnd&pg=PR7&dq=Robot+vision&ots=ta6Uv5To1h&sig=DRUKMuM1-UXZ6VS5xubjTUVDDzM.

Hornung, A., Wurm, K., Bennewitz, M., and Stachniss, M. OctoMap: an efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, pages 1–18, 2013. URL http://link.springer.com/article/10.1007/s10514-012-9321-0.

Hu, J. and Guo, P. Spatial local binary patterns for scene image classification. In *2012 6th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT)*, pages 326–330. IEEE, March 2012. ISBN 978-1-4673-1658-3. doi: 10.1109/SETIT.2012.6481936. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6481936.

Huang, S. and Wang, Z. Sparse local submap joining filter for building large-scale maps Sparse local submap joining filter for building large-scale maps. *IEEE Transactions on Robotics*, 2008.

Husin, Z., Shakaff, A., Aziz, A., Farook, R., Jaafar, M., Hashim, U., and Harun, A. Embedded portable device for herb leaves recognition using image processing techniques and neural network algorithm. *Computers and Electronics in Agriculture*, 89(null):18–29, November 2012. ISSN 01681699. doi: 10.1016/j.compag.2012.07.009. URL http://dx.doi.org/10.1016/j.compag.2012.07.009.

Jae Gon, A. and Heung Seok, J. R-Map: A hybrid map created by maximal rectangles. *Control Automation and Systems (ICCAS), 2010 International Conference on*, pages 1336–1339. URL http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5670300.

Jebari, I., Bazeille, S., Battesti, E., Tekaya, H., Klein, M., Tapus, A., Filliat, D., Meyer, C., Ieng, S.-H., Benosman, R., Cizeron, E., Mamanna, J.-C., and Pothier, B. *Multi-sensor semantic mapping and exploration of indoor environments*. IEEE, April 2011a. ISBN 978-1-61284-482-4. doi: 10.1109/TEPRA.2011.5753498. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5753498.

Jebari, I., Bazeille, S., and Filliat, D. Combined Vision and Frontier-Based Exploration Strategies for Semantic Mapping 2 Related Work 3 Semantic Mapping 4 Multi-Objective Exploration for Semantic Mapping. *Informatics in Control, Automation and Robotics*, 133 LNEE(VOL.2): 237–244, 2011b.

Jensfelt, P., Zender, H., and Kruijff, G.-j. M. From Labels to Semantics : An Integrated System for Conceptual Spatial Representations of Indoor Environments for Mobile Robots. In *ICRA Workshop: Semantic Information in Robotics,*, 2007.

Joo, K., Lee, T.-K., Baek, S., and Oh, S.-Y. Generating topological map from occupancy grid-map using virtual door detection. In *IEEE Congress on Evolutionary Computation*, pages 1–6. IEEE, July 2010. ISBN 978-1-4244-6909-3. doi: 10.1109/CEC.2010.5586510. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5586510.

Juan, L. and Gwun, O. A comparison of sift, pca-sift and surf. *International Journal of Image Processing (IJIP)*, 3(4):143–152, 2009. URL http://www.doaj.org/doaj?func=fulltext&aId=509434.

Jurie, F. and Triggs, B. Creating efficient codebooks for visual recognition. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference*, pages 604–610, 2005. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1541309.

Kai-Bo, D. and Keerthi; S. Sathiya. Which Is the Best Multiclass SVM Method? An Empirical Study. In *Proceedings of the Sixth International Workshop on Multiple Classifier Systems. Lecture Notes in Computer Science*, 2005.

Kalman, R. A new approach to linear filtering and prediction problems. *Journal of basic Engineering*, 82(Series D):35–45, 1960. URL http://160.78.24.2/Public/Kalman/Kalman1960.pdf.

Kasra Khosoussi, P. H. D. T. SLAM with Prior Knowledge (Artificial Landmarks), 2009. URL http://saba.kntu.ac.ir/eecd/aras/students/kasra/research.html.

Ke, Y. and Sukthankar, R. PCA-SIFT: A more distinctive representation for local image descriptors. *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference*, 2:II–506–II–513, 2004. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1315206.

Kim, G., Chung, W., Kim, K., M., Han, S., and Shinn, R. H. The autonomous tour-guide robot jinny. *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference*, 4:3450–3455, 2004. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1389950.

Kirschvink, J. L., Kobayashi-Kirschvink, A., Diaz-Ricci, J. C., and Kirschvink, S. J. Magnetite in human tissues: A mechanism for the biological effects of weak ELF magnetic fields. *Bioelectromagnetics*, 13(S1):101–113, 1992. ISSN 0197-8462. doi: 10.1002/bem.2250130710. URL http://doi.wiley.com/10.1002/bem.2250130710.

Klippenstein, J. and Zhang, H. *Performance evaluation of visual SLAM using several feature extractors*. IEEE, October 2009. ISBN 978-1-4244-3803-7. doi: 10.1109/IROS.2009.5354001. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5354001.

Kohlbrecher, S., Meyer, J., von Stryk, O., and Klingauf, U. A Flexible and Scalable SLAM System with Full 3D Motion Estimation. In *Proc. IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*, pages 155–160. IEEE, 2011a.

Kohlbrecher, S., Meyer, J., von Stryk, O., and Klingauf, U. A flexible and scalable slam system with full 3d motion estimation. *Proc. IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*, pages 155–160, 2011b. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6106777.

Kollar, T., Tellex, S., Roy, D., and Roy, N. Toward understanding natural language directions. *Human-Robot Interaction (HRI), 2010 5th ACM/IEEE International Conference*, pages 259–266, 2010a. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5453186.

Kollar, T., Tellex, S., Roy, D., and Roy, N. Grounding verbs of motion in natural language commands to robots. *International Symposium on Experimental Robotics.*, ([1] T. Kollar, S. Tellex, D. Roy, and N. Roy, “Grounding verbs of motion in natural language commands to

robots," International Symposium on Experimental Robotics., 2010.), 2010b. URL http://people.csail.mit.edu/stefie10/publications/tkollar_iser2010.pdf.

Kortenkamp, D. TRACLabs in new facility, 2012. URL http://traclabs.com/2011/03/traclabs-in-new-facility/.

Kosecká, J. and Li, F. Vision based topological Markov localization. *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference*, 2:1481–1486, 2004. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1308033.

Kröse, B., Vlassis, N., Bunschoten, R., and Motomura, Y. A Probabilistic Model for Appearance-Based Robot Localization. In *First European Symposium On Ambience Intelligence (EUSAI)*, pages 264 – 274, 2000. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.31.9122.

Kruijff, G.-j. M., Zender, H., Jensfelt, P., and Christensen, H. I. Situated Dialogue and Spatial Organization : What , Where . . . and Why ? *International Journal of Advanced Robotic Systems*, 4, 2007.

Kuipers, B. Modeling Spatial Knowledge. *Cognitive Science*, 2:129 – 153, 1978. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.29.1915.

Kuipers, B. and Byun, Y.-t. A Robot Exploration and Mapping Strategy Based on a Semantic Hierarchy of Spatial Representations. *Journal Of Robotics And Autonomous Systems*, 8:47 – 63, 1991. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.52.1472.

Labbani-Igbida, O., Charron, C., and Mouaddib, E. M. Haar invariant signatures and spatial recognition using omnidirectional visual information only. *Autonomous Robots*, 30(3):333–349, February 2011. ISSN 0929-5593. doi: 10.1007/s10514-011-9222-7. URL http://www.springerlink.com/index/10.1007/s10514-011-9222-7.

Land, E. and McCann, J. Lightness and retinex theory. *Journal of the Optical society of America*, 61(1):1–11, 1971. URL http://mccannimaging.com/Retinex/Retinex_files/L&M1971.pdf.

Lau, B., C. Sprunk, and Burgard, W. Improved Updating of Euclidean Distance Maps and Voronoi Diagrams. In *IEEE International Conference on Intelligent RObots and Systems (IROS)*, Taipei, Taiwan, 2010.

Lazaros, N., Sirakoulis, G. C., and Gasteratos, A. Review of Stereo Vision Algorithms: From Software to Hardware. *International Journal of Optomechatronics*, 2(4):435–462, November 2008. ISSN 1559-9612. doi: 10.1080/15599610802438680. URL http://dx.doi.org/10.1080/15599610802438680.

Lazebnik, S., Schmid, C., and Ponce, J. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference*, 2:2169–2178, 2006. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1641019.

Lee, Y.-J. and Song, J.-B. Autonomous Salient Feature Detection through Salient Cues in an HSV Color Space for Visual Indoor Simultaneous Localization and Mapping. *Advanced Robotics*, 24(11):1595–1613, July 2010. ISSN 01691864.

doi: 10.1163/016918610X512613. URL http://www.ingentaconnect.com/content/vsp/arb/2010/00000024/00000011/art00004?token=005a19fd44bffbacf41333c4a2f7a316a593b2b67283e3f3b3e4f6d62222c227e37253033297673ed90

Lee, Y.-J. and Song, J.-B. Three-dimensional iterative closest point-based outdoor SLAM using terrain classification. *Intelligent Service Robotics*, 4(2):147–158, February 2011. ISSN 1861-2776. doi: 10.1007/s11370-011-0087-6. URL http://www.springerlink.com/content/05110747147h4u47/.

LeMaster, E. and Rock, S. A local-area GPS pseudolite-based Mars Navigation System. *IEEE 10th International Conference on Advanced Robotics, Budapest, Hungary*, (August):1–6, 2001. URL http://www.gmat.unsw.edu.au/pseudolite/papers/pl0023.pdf.

Leonard, J. and Durrant-Whyte, H. Simultaneous map building and localization for an autonomous mobile robot. In *Proceedings IROS '91:IEEE/RSJ International Workshop on Intelligent Robots and Systems '91*, pages 1442–1447. IEEE, 1991. ISBN 0-7803-0067-X. doi: 10.1109/IROS.1991.174711. URL http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=174711.

Leonardis, A. and Bischof, H. Robust recognition using eigenimages. *Computer Vision and Image Understanding*, 2000. URL http://www.sciencedirect.com/science/article/pii/S1077314299908305.

Leordeanu, M. and Hebert, M. A spectral technique for correspondence problems using pairwise constraints. *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference*, 2:1482–1489, 2005. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1544893.

Leutenegger, S., Chli, M., and Siegwart, R. Y. BRISK: Binary robust invariant scalable keypoints. In *Computer Vision (ICCV), 2011 IEEE International Conference*, pages 2548–2555. IEEE, 2011. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6126542.

Lin, K.-H. and Wang., C.-C. Stereo-based simultaneous localization, mapping and moving object tracking. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference*, pages 3975–3980. IEEE, 2010. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5649653.

Linde, O. and Lindeberg, T. Object recognition using composed receptive field histograms of higher dimensionality. In *Pattern Recognition, 2004. ICPR 17th International Conference on. Vol. 2. IEEE*, 2004a. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.140.4019.

Linde, O. and Lindeberg, T. Object recognition using composed receptive field histograms of higher dimensionality. In *Pattern Recognition, 2004. ICPR 17th International Conference on. Vol. 2*. IEEE, 2004b. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1333965.

Lindeberg, T. Feature detection with automatic scale selection. *International journal of computer vision*, 1998. URL http://link.springer.com/article/10.1023/A:1008045108935.

Liu, F., Liu, A., Wang, M., and Yang, Z. *Robust and Fast Localization Algorithm for Data Matrix Barcode*. IEEE, November 2010. ISBN 978-1-4244-8683-0. doi: 10.1109/ICOIP.2010.299. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5663102.

Liu, M., Scaramuzza, D., Pradalier, C., Siegwart, R., and Chen, Q. Scene recognition with omnidirectional vision for topological map using lightweight adaptive descriptors. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 116–121. IEEE, October 2009. ISBN 978-1-4244-3803-7. doi: 10.1109/IROS.2009.5354131. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5354131.

Lovász, L. and Plummer, M. *Matching Theory*. Akadémiai Kiadó, Budapest, Also published as Vol. 121 of the North-Holland Mathematics Studies, North-Holland Publishing, Amsterdam, 1986. ISBN 0-444-87916-1. URL http://www.elsevier.com/wps/find/bookdescription.cws_home/501947/description#description.

Lowe, D. G. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110., 2004. URL http://link.springer.com/article/10.1023/B:VISI.0000029664.99615.94.

Lv, Z. and Zhang, Z. *Build 3D Laser Scanner Based on Binocular Stereo Vision*. IEEE, March 2011. ISBN 978-1-61284-289-9. doi: 10.1109/ICICTA.2011.158. URL http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5750689.

MacMahon, M., Stankiewicz, B., and Kuipers, B. Walk the talk: Connecting language, knowledge, and action in route instructions. In *American Association for Artificial Intelligence*, 2006.

Mäenpää, T. and Pietikäinen, M. Classification with color and texture: jointly or separately? *Pattern Recognition*, 37(8):1629–1640, 2004. URL http://www.sciencedirect.com/science/article/pii/S0031320303004321.

Maimone, M., Cheng, Y., and Matthies, L. Two years of Visual Odometry on the Mars Exploration Rovers. *Journal of Field Robotics*, 24(3):169–186, March 2007. ISSN 15564959. doi: 10.1002/rob.20184. URL http://doi.wiley.com/10.1002/rob.20184.

Marocco, D., Cangelosi, A., Fischer, K., and Belpaeme, T. Grounding Action Words in the Sensorimotor Interaction with the World: Experiments with a Simulated iCub Humanoid Robot. *Frontiers in neurorobotics*, 4, January 2010. ISSN 1662-5218. doi: 10.3389/fnbot.2010.00007. URL http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2901088&tool=pmcentrez&rendertype=abstract.

Marr, D. and Hildreth, E. Theory of edge detection. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, (207.1167):187–217, 1980. URL http://rspb.royalsocietypublishing.org/content/207/1167/187.short.

Martínez Mozos, O., Triebel, R., Jensfelt, P., Rottmann, A., and Burgard, W. Supervised semantic labeling of places using information extracted from sensor data. *Robotics and Autonomous Systems*, 55(5):391–402, 2007.

Mataric, M. J. *A Distributed Model for Mobile Robot Environment-Learning and Navigation*. Msc, MIT, May 1990. URL http://dl.acm.org/citation.cfm?id=889361.

Matas, J., Chum, O., Urban, M., and Pajdla, T. Robust wide-baseline stereo from maximally stable extremal regions. *Image and vision computing*, 2004. URL http://www.sciencedirect.com/science/article/pii/S0262885604000435.

Matuszek, C., Fox, D., and Koscher, K. Following directions using statistical machine translation. *Proceeding of the 5th ACM/IEEE international conference on Human-robot interaction - HRI '10*, page 251, 2010. doi: 10.1145/1734454.1734552. URL http://portal.acm.org/citation.cfm?doid=1734454.1734552.

Matuszek, C., Herbst, E., Zettlemoyer, L., and Fox, D. Learning to parse natural language commands to a robot control system. In Cynthia Matuszek, Evan Herbst, Luke Zettlemoyer, D. F., editor, *Proc. of the 2012 International Symposium on Experimental Robotics*, pages 1–14, Québec City, Canada, 2012. URL http://homes.cs.washington.edu/~lsz/papers/mhzf-iser12.pdf.

McKay, B. Practical graph isomorphism. 1981. URL http://cs.anu.edu.au/people/bdm/nauty/pgi.pdf.

Meagher, D. Geometric modeling using octree encoding. *Computer graphics and image processing*, 1982. URL http://www.sciencedirect.com/science/article/pii/0146664X82901046.

Meger, D., Forssen, P.-E., Lai, K., Helmer, S., McCann, S., Southey, T., Baumann, M., Little, J. J., and Lowe, D. G. Curious george: An attentive semantic robot. *Robotics and Autonomous Systems (RAS)*, (56(6)):503–511, 2008. URL http://www.sciencedirect.com/science/article/pii/S0921889008000316.

Meng, X. and Wang, Z. Rapid Scene Categorization Using Novel Gist Model. In *2010 2nd International Conference on Information Engineering and Computer Science*, pages 1–4. IEEE, December 2010. ISBN 978-1-4244-7939-9. doi: 10.1109/ICIECS.2010.5677699. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5677699.

Messmer, B. and Bunke, H. A new algorithm for error-tolerant subgraph isomorphism detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions*, (20.5):493–504., 1998. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=682179.

Miettinen, M., Ohman, M., Visala, A., and Forsman, P. *Simultaneous Localization and Mapping for Forest Harvesters*. IEEE, April 2007. ISBN 1-4244-0602-1. doi: 10.1109/ROBOT.2007.363838. URL http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4209143.

Miihlig, M., Gienger, M., and Steil, J. J. Human-Robot Interaction for Learning and Adaptation of Object Movements. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4901–4907, Taipei, Taiwan, 2010. ISBN 9781424466764.

Mikolajczyk, K. and Schmid, C. Indexing based on scale invariant interest points. *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on. IEEE,*, pages 525–531, 2001. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=937561.

Mikolajczyk, K. and Schmid, C. An affine invariant interest point detector. *Computer Vision—ECCV 2002*, 2002. URL http://link.springer.com/chapter/10.1007/3-540-47969-4_9.

Mikolajczyk, K. and Schmid, C. A performance evaluation of local descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions*, 27(10):1615–1630, 2005. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1498756.

Mikolajczyk, K. and Tuytelaars, T. A comparison of affine region detectors. *International journal of computer vision*, 1(2):43–72, 2005. URL http://link.springer.com/article/10.1007/s11263-005-3848-x.

Modayil, J. and Kuipers, B. Autonomous development of a grounded object ontology by a learning robot. In 1999., A. P. M. P., editor, *In Proceedings of the national conference on Artificial intelligence (Vol. 22, No. 2, p. 1095)*, Cambridge, MA; London;, 2007. URL http://www.aaai.org/Papers/AAAI/2007/AAAI07-174.pdf.

Montemerlo, M., Thrun, S., Koller, D., and Wegbreit, B. FastSLAM: A factored solution to the simultaneous localization and mapping problem. *AAAI/IAAI*, 2002. URL http://www.aaai.org/Papers/AAAI/2002/AAAI02-089.pdf.

Mozos, O. M. *Semantic Labeling of Places with Mobile Robots*. Phd, Albert Ludwigs Universitat Freiburg, 2008.

Murase, H. and Nayar, S. Visual learning and recognition of 3-D objects from appearance. *International journal of computer vision*, 1995. URL http://link.springer.com/article/10.1007/BF01421486.

Murillo, A., Košecká, J., Guerrero, J., and Sagüés, C. Visual door detection integrating appearance and shape cues. *Robotics and Autonomous Systems*, 56(6):512–521, June 2008. ISSN 09218890. doi: 10.1016/j.robot.2008.03.003. URL http://dl.acm.org/citation.cfm?id=1377036.1377146.

Murphy, K., Torralba, A., Eaton, D., and Freeman, W. Object detection and localization using local and global features. *Toward Category-Level Object Recognition*, pages 382–400, 2006.

Myung, H., Jeon, H., Jeong, W., and Bang, S. Virtual door-based coverage path planning for mobile robot. *Advances in Robotics*, pages 197–207, 2009. URL http://www.springerlink.com/index/48108M8J1607N370.pdf.

Newman, P. and Leonard, J. Explore and return: Experimental validation of real-time concurrent mapping and localization. *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference*, 2:1802–1809, 2002. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1014803.

Nieto-Granda, C., Rogers, J. G., Trevor, A. J. B., and Christensen, H. I. Semantic map partitioning in indoor environments using regional analysis. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'10)*, pages 1451–1456, Taipei, Taiwan, 2010. ISBN 9781424466764. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5650575.

Nistér, D., Naroditsky, O., and Bergen, J. Visual odometry. *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on. IEEE*, 1:652–659, 2004. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1315094.

Nourbakhsh, I., Clay, K., and Thomas, W. The mobot museum robot installations: A five year experiment. *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on. IEEE,*, 3:3636–3641, 2003. doi: 10.1109/IROS.2003.1249720. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1249720http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1249720.

Nowak, E., Jurie, F., and Triggs, B. Sampling strategies for bag-of-features image classification. *Computer Vision–ECCV 2006*, 2006. URL http://link.springer.com/chapter/10.1007/11744085_38.

Nüchter, A. *3D robotic mapping: the simultaneous localization and mapping problem with six degrees of freedom*. Springer, 2009. URL http://www.google.com/books?hl=pt-PT&lr=&id=7gOspY2t1Q0C&oi=fnd&pg=PA1&dq=3D+Robotic+Mapping,&ots=GDwLtE7k9W&sig=26YK4TRave1PcHqaytY1zBmI5XY.

Nüchter, A. and Hertzberg, J. Towards semantic maps for mobile robots. *Robot. Auton. Syst.*, 56(11):915–926, November 2008. ISSN 0921-8890. doi: 10.1016/j.robot.2008.08.001. URL http://portal.acm.org/citation.cfm?id=1453261.1453481.

Oberlander, J. and Uhl, K. A region-based SLAM algorithm capturing metric, topological, and semantic properties. In *IEEE International Conference on Robotics and Automation*, pages 1886–1891, 2008. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4543482.

Ojala, T., Pietikäinen, M., and Harwood, D. A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, 29(1):51–59, 1996. URL http://www.sciencedirect.com/science/article/pii/0031320395000674.

Ojala, T., Pietikainen, M., and Maenpaa., T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions*, 24(2):971–987., 2002. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1017623.

Oliva, A. and Torralba, A. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3):145–175, 2001. URL http://link.springer.com/article/10.1023/A:1011139631724.

Oliva, A. and Torralba, A. Building the gist of a scene: the role of global image features in recognition. *Progress in Brain Research*, 155, 2006. URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.88.7631.

Paris, S. and Gloti, H. Pyramidal Multi-level Features for the Robot Vision@ICPR 2010 Challenge. In *2010 20th International Conference on Pattern Recognition*, pages 2949–2952. IEEE, August 2010. ISBN 978-1-4244-7542-1. doi: 10.1109/ICPR.2010.1143. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5595908.

Paul, R. and Newman, P. FAB-MAP 3D: Topological mapping with spatial and visual appearance. *2010 IEEE International Conference on Robotics and Automation*, pages 2649–2656, May 2010. doi: 10.1109/ROBOT.2010.5509587. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5509587.

Pelillo, M. Replicator equations, maximal cliques, and graph isomorphism. *Neural Computation*, 1999. URL http://www.mitpressjournals.org/doi/abs/10.1162/089976699300016034.

Petry, M., Moreira, A. P., and Reis, L. P. Increasing Illumination Invariance of SURF Feature Detector through Color Constancy. *Progress in Artificial Intelligence. Springer Berlin Heidelberg*, pages 259–270, 2013. URL http://link.springer.com/chapter/10.1007/978-3-642-40669-0_23.

Petry, M. R. *A Vision-based Approach Towards Robust Localization for Intelligent Wheelchairs*. Phd, 2013.

Pham, T.-T., Mulhem, P., Maisonnasse, L., Gaussier, E., and Lim, J.-H. Visual graph modeling for scene recognition and mobile robot localization. *Multimedia Tools and Applications*, 60 (2):419–441, September 2010. ISSN 1380-7501. doi: 10.1007/s11042-010-0598-8. URL http://link.springer.com/10.1007/s11042-010-0598-8.

Pieraccini, R. *The voice in the machine: building computers that understand speech*. The MIT Press., 2012. URL http://www.google.com/books?hl=en&lr=&id=Edxx8hcTjzMC&oi=fnd&pg=PR7&dq=The+Voice+in+the+Machine.+Building+Computers+That+Understand+Speech&ots=e2SfxklB7J&sig=eMdHlBguwmEH6Jx9OPsFzUWWeBs.

Pietikäinen, M., Zhao, G., Ahonen, T., and Hadid, A. *Computer vision using local binary patterns*. Springer, 2011. ISBN 9780857297471. URL http://books.google.com/books?hl=en&lr=&id=wBrZz9FiERsC&oi=fnd&pg=PR3&dq=Computer+Vision+Using+Local+Binary+Patterns&ots=XnbUlLac5o&sig=S8IhyAN0c3W59tTZy-ab6k-TXew.

Pinto, M., Moreira, A. P., Matos, A., and Santos, F. Fast 3D Matching Localisation Algorithm. *Journal of Automation and Control Engineering,*, 1(2):110–115 ISSN:2301–3702, 2013a.

Pinto, M., Moreira, A. P., and Matos, A. *Robots Localisation in Indoor and Service Scenarios: Two Approaches: Landmark-Based and Three-Dimensional Map-Based*. LAP LAMBERT Academic Publishing, 2013b. ISBN 3659336475. URL http://www.amazon.com/Robots-Localisation-Indoor-Service-Scenarios/dp/3659336475.

Posada, L.-f., Nierobisch, T., Hoffmann, F., and Bertram, T. Image signal processing for visual door passing with an omnitdirectional camera. In *In Proc. Int. Conf. on Computer Vision Theory and Applications*, 2009.

Priyantha, N. B., Chakraborty, A., and Balakrishnan, H. The cricket location-support system. *6th ACM International Conference on Mobile Computing and Networking (ACM MOBICOM), Boston,*, 2000(August), 2000. URL http://dl.acm.org/citation.cfm?id=345917.

Pronobis, A., Caputo, B., Jensfelt, P., and Christensen, H. I. A discriminative approach to robust visual place recognition. In *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference*, pages 3829–3836. IEEE, 2006. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4059003.

Pronobis, A., Martinez Mozos, O., Caputo, B., and Jensfelt, P. Multi-modal Semantic Place Classification. *The International Journal of Robotics Research*, 29(2-3):298–320, December

2009. ISSN 0278-3649. doi: 10.1177/0278364909356483. URL http://ijr.sagepub.com/cgi/content/abstract/29/2-3/298.

Pronobis, A. *Semantic Mapping with Mobile Robots*. Phd, KTH Royal Institute of Technology, 2011.

Pronobis, A. and Caputo, B. Confidence-based cue integration for visual place recognition. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference*, pages 2394–2401. IEEE, 2007. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4399493.

Pronobis, A. and Caputo, B. COLD: COsy Localization Database. *The International Journal of Robotics Research (IJRR)*, 28:588–594, 2009.

Pronobis, A. and Jensfelt, P. Large-scale semantic mapping and reasoning with heterogeneous modalities. *2012 IEEE International Conference on Robotics and Automation*, pages 3515–3522, May 2012. doi: 10.1109/ICRA.2012.6224637. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6224637.

Pronobis, A., Mozos, O. M., and Caputo., B. SVM-based discriminative accumulation scheme for place recognition. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference*, pages 522–529. IEEE, 2008. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4543260.

Qu, X., Yao, M., Gu, Q., and Zhang, J. Adaptive Subspace Based Online PCA Algorithm for Mobile Robot Scene Learning and Recognition. In *2011 Third International Conference on Intelligent Human-Machine Systems and Cybernetics*, volume 1, pages 205–209. IEEE, August 2011. ISBN 978-1-4577-0676-9. doi: 10.1109/IHMSC.2011.56. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6038182.

Quelhas, P. and Monay, F. Modeling scenes with local descriptors and latent aspects. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on. Vol. 1*. IEEE, 2005. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1541347.

Quintas, J., Menezes, P., and Dias, J. Cloud Robotics: Towards Context Aware Robotic Networks. *Proc. of the 16th IASTED International Conference on Robotics, Pittsburgh, USA.*, 2011. URL http://www.actapress.com/Abstract.aspx?paperId=452861.

Rady, S., Wagner, A., and Badreddin, E. Building efficient topological maps for mobile robot localization: An evaluation study on COLD benchmarking database. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 542–547. IEEE, October 2010. ISBN 978-1-4244-6674-0. doi: 10.1109/IROS.2010.5649670. URL http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5649670.

Ramos, F., Upcroft, B., Kumar, S., and Durrant-Whyte, H. A Bayesian approach for place recognition. *Robotics and Autonomous Systems*, 60(4):487–497, 2012. URL http://www.sciencedirect.com/science/article/pii/S0921889011002053.

Ranganathan, A. Pliss: Detecting and labeling places using online change-point detection. *Robotics: Science and Systems VI*, 2010. URL http://roboticsproceedings.org/rss06/p24.pdf.

Ranganathan, A. PLISS: labeling places using online changepoint detection. *Autonomous Robots*, 32(4):351–368, January 2012. ISSN 0929-5593. doi: 10.1007/s10514-012-9273-4. URL http://link.springer.com/10.1007/s10514-012-9273-4.

Ranganathan, A. and Dellaert, F. Semantic Modeling of Places using Objects. In *Robotics: Science and Systems*, 2007.

Rasiwasia, N. and Vasconcelos, N. Scene classification with low-dimensional semantic spaces and weak supervision. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference*, pages 1–5. IEEE, 2008. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4587372http://mplab.ucsd.edu/wp-content/uploads/cvpr2008/Conference/data/papers/032.pdf.

Rasiwasia, N. and Vasconcelos, N. Holistic context models for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 34(5):902–17, May 2012. ISSN 1939-3539. doi: 10.1109/TPAMI.2011.175. URL http://www.ncbi.nlm.nih.gov/pubmed/21844625.

Rasolzadeh, B., Bjorkman, M., Huebner, K., and Kragic, D. An Active Vision System for Detecting, Fixating and Manipulating Objects in the Real World. *The International Journal of Robotics Research*, 29(2-3):133–154, August 2009. ISSN 0278-3649. doi: 10.1177/0278364909346069. URL http://ijr.sagepub.com/cgi/doi/10.1177/0278364909346069.

Regier, T. The acquisition of lexical semantics for spatial terms: A connectionist model of perceptual categorization. 1992. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.48.9613&rep=rep1&type=pdf.

Reiser, U., Connette, C., Fischer, J., Kubacki, J., Bubeck, A., Weisshardt, F., Jacobs, T., Parlitz, C., Martin, H., and Verl, A. Care-O-bot R 3 - Creating a product vision for service robot applications by integrating design and technology. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1992–1998, 2009. ISBN 9781424438044.

Robles-Castro, J., Duchen-Sanchez, G., and Takahashi, H. *Improving object position estimation based on non-linear mapping using Relevance Vector Machine*. IEEE, February 2011. ISBN 978-1-4244-9558-0. doi: 10.1109/CONIELECOMP.2011.5749355. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5749355.

Roy, A. H. and Nicholas. The Robotics Data Set Repository (Radish), 2003.

Roy, D. Semiotic schemas: A framework for grounding language in action and perception. *Artificial Intelligence*, 2005. URL http://www.sciencedirect.com/science/article/pii/S0004370205001037.

Rublee, E. and Rabaud, V. ORB: an efficient alternative to SIFT or SURF. In *Computer Vision (ICCV), 2011 IEEE International Conference*. IEEE, 2011. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6126544.

Ruiz-del Solar, J., Chown, E., Plöger, P., Pellenz, J., Neuhaus, F., Dillenberger, D., Gossow, D., and Paulus, D. *Mixed 2D/3D Perception for Autonomous Robots in Unstructured Environments*, volume 6556 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. ISBN 978-3-642-20216-2. doi: 10.1007/978-3-642-20217-9. URL http://www.springerlink.com/content/l2480t3151820u13/.

Rusu, R. B. *Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments*. PhD thesis, Institut für Informatik der Technischen Universität München, 2009.

Sandini, G., Metta, G., and Vernon, D. The iCub Cognitive Humanoid Robot : An Open-System Research Platform for Enactive Cognition Enactive Cognition : Why Create a Cognitive Humanoid. In Berlin, S.-V., editor, *50 years of artificial intelligence*, pages 359–370. 2007. ISBN 3-540-77295-2.

Santo, M. D., Foggia, P., Sansone, C., and Vento, M. A large database of graphs and its use for benchmarking graph isomorphism algorithms. *Pattern Recognition Letters*, 2003. URL http://www.sciencedirect.com/science/article/pii/S0167865502002532.

Santos, F. *Determinação em tempo real da posição e atitude de um veículo móvel aéreo com múltiplos receptores GPS de baixo custo*. Dissertação para obtenção do grau de mestre em eng. electrotécnica e de computadores, Universidade técnica de Lisboa - Instituto Superior Técnico, 2007.

Sarfraz, M. and Hellwich, O. Head Pose Estimation in Face Recognition Across Pose Scenarios. *VISAPP (1)*, 2008. URL http://scholar.google.pt/scholar?hl=pt-PT&as_sdt=0,5&q=Head+Pose+estimation+in+face+recognition+across+pose+scenarios#0.

Scaramuzza, B. D. and Fraundorfer, F. Visual Odometry. *IEEE Robotics & Automation Magazine*, 11(December):80–92, 2011.

Schellewald, C. *Convex mathematical programs for relational matching of object views*. Phd thesis, University of Mannheim., 2005. URL http://d-nb.info/975862782/34.

Schmidt, J., Wong, C., and Yeap, W. A Split and Merge Approach to Metric-Topological Map-Building. In *18th International Conference on Pattern Recognition (ICPR'06)*, pages 1069–1072. IEEE, 2006. ISBN 0-7695-2521-0. doi: 10.1109/ICPR.2006.176. URL http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1699710.

Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. The MIT Press, 2005. ISBN 0262201623. URL http://www.amazon.com/gp/product/0262201623/102-8479661-9831324?v=glance&n=283155&n=507846&s=books&v=glance.

Shanming, Y. E. and Malik, J. Object Detection in RGB-D Indoor Scenes. Technical report, Electrical Engineering and Computer Sciences University of California at Berkeley, 2013. URL http://www.eecs.berkeley.edu/Pubs/TechRpts/2013/EECS-2013-3.pdf.

Shapiro, L. and Brady, J. M. Feature-based correspondence: an eigenvector approach. *Image and vision computing*, 1992. URL http://www.sciencedirect.com/science/article/pii/0262885692900433.

Shi, W. and Samarabandu, J. Investigating the Performance of Corridor and Door Detection Algorithms in Different Environments. In *2006 International Conference on Information and Automation*, pages 206–211. IEEE, December 2006. ISBN 1-4244-0554-8. doi: 10.1109/ICINFA.2006.374113. URL http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=4250203.

Shimizu, N. and Haas, A. Learning to follow navigational route instructions. *Int'l Joint Conf. on Artificial Intelligence (IJCAI)*, 2009. URL http://www.aaai.org/ocs/index.php/IJCAI/IJCAI-09/paper/viewFile/462/921.

Shokoufandeh, A., Keselman, Y., Demirci, M. F., Macrini, D., and Dickinson, S. Many-to-many feature matching in object recognition: a review of three approaches. *Computer Vision, IET,*, 6(6):500–513, 2012. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6400403.

Siciliano, B. and Khatib, O. *Springer handbook of robotics*. Springer, 2008. ISBN 9783540239574. URL http://books.google.com/books?hl=en&lr=&id=Xpgi5gSuBxsC&oi=fnd&pg=PR53&dq=Springer+Handbook+of+Robotics&ots=lTmgX9f43P&sig=wlcQ-bnexvR1rbKFAcuR3a9zD0s.

Siegwart, R., Arras, K. O., Bouabdallah, S., Burnier, D., Froidevaux, G., Greppin, X., Jensen, B., Lorotte, A., Mayor, L., Meisser, M., Philippsen, R., Piguet, R., Ramel, G., Terrien, G., and Tomatis, N. Robox at Expo.02: A large-scale installation of personal robots. *Robotics and Autonomous Systems*, 42(3-4):203–222, March 2003. ISSN 09218890. doi: 10.1016/S0921-8890(02)00376-7. URL http://linkinghub.elsevier.com/retrieve/pii/S0921889002003767.

Silberztein, M. NooJ: A Linguistic Annotation System For Corpus Processing. In *Proceedings of HLT/EMNLP on Interactive Demonstrations* -, pages 10–11, Morristown, NJ, USA, October 2005. Association for Computational Linguistics. doi: 10.3115/1225733.1225739. URL http://dl.acm.org/citation.cfm?id=1225733.1225739.

Siskind, J. Grounding the lexical semantics of verbs in visual perception using force dynamics and event logic. *Journal of Artificial Intelligence Res. 15, 31–90*, 2011. URL http://arxiv.org/abs/1106.0256.

Sivic, J. and Zisserman, A. Video Google: a text retrieval approach to object matching in videos. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pages 1470–1477 vol.2. IEEE, 2003. ISBN 0-7695-1950-4. doi: 10.1109/ICCV.2003.1238663. URL http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=1238663.

Smith, R. C. and Cheeseman, P. On the Representation and Estimation of Spatial Uncertainty. *The International Journal of Robotics Research*, 5(4):56–68, December 1986. ISSN 0278-3649. doi: 10.1177/027836498600500404. URL http://ijr.sagepub.com/cgi/content/abstract/5/4/56.

Smith, S. and Brady, J. SUSAN—A new approach to low level image processing. *International journal of computer vision*, 1997. URL http://link.springer.com/article/10.1023/A:1007963824710.

Sobreira, H., Santos, F., ALves, H., and Moreira, A. P. Localizing an NXT Lego Robot using infra-red beacons. In *EPIA2011 - 15th Portuguese Conference on Artificial Intelligence*, 2011.

Srinivasa, S. S., Ferguson, D., Helfrich, C. J., Berenson, D., Collet, A., Diankov, R., Gallagher, G., Hollinger, G., Kuffner, J., and Weghe, M. V. HERB: a home exploring robotic butler. *Autonomous Robots*, 28(1):5–20, November 2009. ISSN 0929-5593. doi: 10.1007/s10514-009-9160-9. URL http://www.springerlink.com/index/10.1007/s10514-009-9160-9.

Srinivasan, N. *Feature Based Landmark Extraction for Real Time Visual SLAM*. IEEE, October 2010. ISBN 978-1-4244-8093-7. doi: 10.1109/ARTCom.2010.10. URL http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=5655579.

Sugita, Y. and Tani, J. Learning semantic combinatoriality from the interaction between linguistic and behavioral processes. *Adaptive Behavior*, 2005. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.130.9505&rep=rep1&type=pdf.

Szeliski, R. *Computer Vision: Algorithms and Applications (Texts in Computer Science)*. Springer, 2010. ISBN 1848829345. URL http://www.amazon.com/Computer-Vision-Algorithms-Applications-Science/dp/1848829345.

Tellex, S., Kollar, T., and Dickerson, S. Approaching the symbol grounding problem with probabilistic graphical models. *AI Magazine*, 32(4):64–76, 2011. URL http://dspace.mit.edu/handle/1721.1/73542.

Thrun, S., Beetz, M., Bennewitz, M., Burgard, W., Cremers, A. B., Dellaert, F., Fox, D., Rosenberg, C., Roy, N., Schulte, J., and Schulz, D. Probabilistic Algorithms and the Interactive Museum Tour-Guide Robot Minerva. *Journal of Robotics Research*, pages 972–99, 2000.

Thrun, S. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, February 1998. ISSN 00043702. doi: 10.1016/S0004-3702(97)00078-7. URL http://dl.acm.org/citation.cfm?id=275387.275389http://www.scribd.com/doc/64310081/Learning-Metric-Topological-Maps-for-Indoor-Mobile-Robot-Navigation.

Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., Lau, K., Oakley, C., Palatucci, M., Pratt, V., Stang, P., Strohband, S., Dupont, C., Jendrossek, L.-e., Koelen, C., Markey, C., Rummel, C., Niekerk, J. V., Jensen, E., Alessandrini, P., Bradski, G., Davies, B., Ettinger, S., Kaehler, A., Nefian, A., and Mahoney, P. Stanley : The Robot that Won the DARPA Grand Challenge. *Journal of Field Robotics*, 23 (April):661–692, 2006. doi: 10.1002/rob.

Tikhanoff, V., Cangelosi, A., and Metta, G. Integration of Speech and Action in Humanoid Robots: iCub Simulation Experiments. *IEEE Transactions on Autonomous Mental Development*, 3(1): 17–29, March 2011. ISSN 1943-0604. doi: 10.1109/TAMD.2010.2100390. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5672581.

Torralba, A. and Murphy, K. Context-based vision system for place and object recognition. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference*, pages 273–280. IEEE, 2003. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1238354.

Torsello, A. and Hancock, E. Computing approximate tree edit distance using relaxation labeling. *Pattern Recognition Letters*, 2003. URL http://www.sciencedirect.com/science/article/pii/S0167865502002556.

Trajković, M. and Hedley, M. Fast corner detection. *Image and Vision Computing*, 1998. URL http://www.sciencedirect.com/science/article/pii/S0262885697000565.

Tsai, W.-H. and Fu, K.-S. Subgraph error-correcting isomorphisms for syntactic pattern recognition. *Systems, Man and Cybernetics, IEEE Transactions*, 1:48–62, 1983. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6313029.

Tsai, W. and Fu, K. Error-correcting isomorphisms of attributed relational graphs for pattern analysis. *Systems, Man and Cybernetics, IEEE Transactions*, 9(12):757–768, 1979. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4310127.

Ullmann, J. An algorithm for subgraph isomorphism. *Journal of the ACM (JACM)*, 1976. URL http://dl.acm.org/citation.cfm?id=321925.

Ulrich, I. and Nourbakhsh, I. Appearance-based place recognition for topological localization. *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference*, 2:1023–1029, 2000. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=844734.

Umeyama, S. An eigendecomposition approach to weighted graph matching problems. *Pattern Analysis and Machine Intelligence, IEEE Transactions*, 10(5):695–703, 1988. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6778.

Unsal, D. and Demirbas, K. Estimation of deterministic and stochastic IMU error parameters. *Proceedings of the 2012 IEEE/ION Position, Location and Navigation Symposium*, pages 862–868, 2012. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6236828.

van de Sande, K. E. A., Gevers, T., and Snoek, C. G. M. Evaluating color descriptors for object and scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 32(9): 1582–96, September 2010. ISSN 1939-3539. doi: 10.1109/TPAMI.2009.154. URL http://www.ncbi.nlm.nih.gov/pubmed/20634554.

Van De Weijer, J., Gevers, T., and Gijsenij., A. Edge-based color constancy. *Image Processing, IEEE Transactions*, 16(9):2207–2214., 2007. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4287009.

Vapnik, V. Statistical learning theory. 1998. URL http://www.citeulike.org/group/664/article/106699.

Vasudevan, S. and Siegwart, R. Bayesian space conceptualization and place classification for semantic maps in mobile robotics. *Robotics and Autonomous Systems*, 56(6):522–537, June 2008. ISSN 09218890. doi: 10.1016/j.robot.2008.03.005. URL http://linkinghub.elsevier.com/retrieve/pii/S092188900800033X.

Viswanathan, P., Southey, T., Little, J., and Mackworth, A. Place Classification Using Visual Object Categorization and Global Information. In *2011 Canadian Conference on Computer and Robot Vision*, pages 1–7. IEEE, May 2011. ISBN 978-1-61284-430-5. doi: 10.1109/CRV.2011.8. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5957535.

Vogel, A. and Jurafsky, D. Learning to follow navigational directions. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics.*, pages 806–814, 2010. URL http://dl.acm.org/citation.cfm?id=1858764.

Waibel, M., Beetz, M., and Civera, J. Roboearth. *Robotics & Automation Magazine, IEEE*, 2(18): 69–82, 2011. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5876227.

Wang, Y., Chen, D., and Shi, C. Autonomous Navigation Based on a Novel Topological Map. In *2009 Asia-Pacific Conference on Information Processing*, pages 254–257. IEEE, July 2009. ISBN 978-0-7695-3699-6. doi: 10.1109/APCIP.2009.71. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5197044.

Weijer, V. D., Joost, T. G., and Bagdanov, A. D. Boosting color saliency in image feature detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions*, 28(1):150–156, 2006. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1542040.

Wilhelms, J. and Gelder, A. V. Octrees for faster isosurface generation. *ACM Transactions on Graphics (TOG)*, 1992. URL http://dl.acm.org/citation.cfm?id=130882.

Wilson, P. and Fernandez, J. Facial feature detection using Haar classifiers. *Journal of Computing Sciences in Colleges*, 21(4):127–133, 2006. URL http://dl.acm.org/citation.cfm?id=1127416.

Wilson, R. and Hancock, E. Structural matching by discrete relaxation. *Pattern Analysis and Machine Intelligence, IEEE Transactions*, 19(6):634–648, 1997. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=601251.

Wiltschko, W., Traudt, J., Güntürkün, O., Prior, H., and Wiltschko, R. Lateralization of magnetic compass orientation in a migratory bird. *Nature*, 419(6906):467–70, October 2002. ISSN 0028-0836. doi: 10.1038/nature00958. URL http://dx.doi.org/10.1038/nature00958.

Winograd, T. *Procedures as a representation for data in a computer program for understanding natural language*. Phd dissertation, Massachusetts Institute of Technology., January 1971. URL http://dspace.mit.edu/handle/1721.1/7095.

Wolf, J., Burgard, W., and Burkhardt, H. Using an image retrieval system for vision-based mobile robot localization. *Image and Video Retrieval*, 2002. URL http://link.springer.com/chapter/10.1007/3-540-45479-9_12.

Wong, A., You, M., and Chan, S. An algorithm for graph optimal monomorphism. *Systems, Man and Cybernetics, IEEE Transactions*, 20(3):628–638, 1990. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=57275.

Wu, J. and Rehg, J. M. Where am I: Place instance and category recognition using spatial PACT. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference*, 2008. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4587627.

Wu, J. and Rehg, J. M. CENTRIST: A Visual Descriptor for Scene Categorization. *IEEE transactions on pattern analysis and machine intelligence*, 33(8):1489–1501, December 2010. ISSN 1939-3539. doi: 10.1109/TPAMI.2010.224. URL http://www.ncbi.nlm.nih.gov/pubmed/21173449.

Wu, J., Christensen, H. I., and Rehg, J. M. Visual Place Categorization: Problem, dataset, and algorithm. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4763–4770. IEEE, October 2009. ISBN 978-1-4244-3803-7. doi: 10.1109/IROS.2009.5354164. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5354164.

Wyk, M. V. A RKHS interpolator-based graph matching algorithm. *Pattern Analysis and Machine Intelligence, IEEE Transactions*, 24(7):988–995, 2002. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1017624.

Xing, L. and Pronobis, A. Multi-cue discriminative place recognition. *Multilingual Information Access Evaluation II. Multimedia Experiments., Springer Berlin Heidelberg*, pages 315–323., 2010. URL http://link.springer.com/chapter/10.1007/978-3-642-15751-6_41.

Xu, G. *Gps: Theory, Algorithms and Applications*. Springer, 2003. ISBN 3540678123. URL http://books.google.com/books?id=aRKPAXBt174C&pgis=1.

Xu, L. and King, I. A PCA approach for fast retrieval of structural patterns in attributed graphs. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transaction*, 31(5):812–817, 2001. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=956043.

Zaslavskiy, M., Bach, F., and Vert, J.-P. A path following algorithm for the graph matching problem. *Pattern Analysis and Machine Intelligence, IEEE Transactions*, 31(12):2227–2242, 2009. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4641936.

Zeki, S. A Vision of the Brain. 1993. URL http://brain.oxfordjournals.org/content/116/4/local/back-matter.pdf.

Zender, H., Jensfelt, P., Mozos, O. M., Kruijff, G.-j. M., and Wolfram Burgard. An Integrated Robotic System for Spatial Understanding and Situated Interaction in Indoor Environments. *Artificial Intelligence*, pages 1584–1589, 2007.

Zender, H., Mozos, O. M., Jensfelt, P., Kruijff, G.-J. M., and Burgard, W. Conceptual spatial representations for indoor mobile robots. *Robotics and Autonomous Systems (RAS)*, 56(6): 493–502, 2008. URL http://www.sciencedirect.com/science/article/pii/S0921889008000304.

Zhang, H., Dou, L., Fang, H., and Chen, J. Autonomous indoor exploration of mobile robots based on door-guidance and improved dynamic window approach. In *2009 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 408–413. IEEE, December 2009. ISBN 978-1-4244-4774-9. doi: 10.1109/ROBIO.2009.5420681. URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5420681.

Zhou, F. and la Torre, F. D. Deformable Graph Matching. Technical report, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, 2013. URL http://www.f-zhou.com/gm/2013_CVPR_DGM.pdf.

Zhou, F. and Torre, F. e. D. l. T. Factorized graph matching. *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE*, pages 127–134, 2012. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6247667.

Zhou, J. and Shi, J. Effect of Facility Geometry on RFID Localization Accuracy. *ASME Conference Proceedings*, 2009(43611):267–275, January 2009. URL http://link.aip.org/link/abstract/ASMECP/v2009/i43611/p267/s1.

Zhou, L., Zhou, Z., and Hu, D. Scene classification using a multi-resolution bag-of-features model. *Pattern Recognition, Elsivier*, 46(1):424–433, 2012. URL http://www.sciencedirect.com/science/article/pii/S0031320312003330.

Zhou, L., Hu, D., and Zhou, Z. Scene recognition combining structural and textural features. *Science China Information Sciences*, 56(7):1–14, October 2011. ISSN 1674-733X. doi: 10.1007/s11432-011-4421-6. URL http://link.springer.com/10.1007/s11432-011-4421-6.

Zivkovic, Z., Bakker, B., and Krose, B. Hierarchical Map Building and Planning based on Graph Partitioning. In *Proc. of IEEE International Conference on Robotics and Automation*, pages 803–809, 2006.