Universitat de Girona

APPEARANCE-BASED MAPPING AND LOCALIZATION USING FEATURE STABILITY HISTOGRAMS FOR MOBILE ROBOT NAVIGATION

Eval Bladimir BACCA CORTES

Dipòsit legal: GI. 1196-2012
http://hdl.handle.net/10803/7xxx

Ph.D. Thesis

Appearance-based Mapping and Localization using Feature Stability Histograms for Mobile Robot Navigation

Presented by

Eval Bladimir Bacca Cortes

2012

Institute of Informatics and Applications

COMPUTER VISION AND ROBOTICS GROUP

Thesis submitted for the degree of

Ph.D. in Technology

Thesis Advisors:

Dr. Xavier Cufí	**Dr. Prof. Joaquim Salvi**
VICOROB	VICOROB
University of Girona	University of Girona

We, **Xavier Cufí** and **Joaquim Salvi**, professors of the **Computer Vision and Robotics** group at the *University of Girona*,

ATTEST:

That this thesis "*Appearance-based Mapping and Localization using Feature Stability Histograms for Mobile Robot Navigation*", submitted by **Eval Bladimir Bacca Cortes** for the degree of European Ph.D. in Technology, was carried under our supervision.

Signatures

Authorship Declaration

I hereby declare that this thesis contains no material which has been accepted for the award of any other degree or diploma in any university. To the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference has been made.

Bladimir Bacca Cortes.

Funding

This work has been partially funded by the Commission of Science and Technology of Spain (CICYT) through the coordinated projects DPI-2007-66796-C03-02, and the project RAIMON (Autonomous Underwater Robot for Marine Fish Farms Inspection and Monitoring) CTM2011-29691-C02-02 funded by the Spanish Ministry of Science and Innovation. The LASPAU-COLCIENCIAS grant 136-2008, the University of Valle contract 644-19-04-95, and the consolidated research group grant 2009SGR380.

Abstract

These days mobile robots are needed to interact within non-structured environments. They must deal with people, moving obstacles, perceptual aliasing, weather changes, occlusions, long term navigation and robot-human interaction in order to have high levels of autonomy from a decision-making point of view. These requirements are useful for service robots designed to conduct surveillance, inspect, deliver, clean and explore. Applications where robots need to collect sensor measurements from complex environments and extract meaningful information to achieve their tasks.

Simultaneous Localization and Mapping (SLAM) is considered essential for mobile robots immersed in real world applications requiring any prior information about the environment. Robotics community has been trying to solve the SLAM problem in many ways, and using appearance or metric information to represent the environment. This thesis is concerned with the problem of appearance-based mapping and localization for mobile robots in changing environments. This introduces our research question: How can a mobile robot update its internal representation of the environment and its location on it when the appearance of the environment is changing?

This work proposes an appearance-based mapping and localization method whose main contribution is the Feature Stability Histogram (FSH). The FSH is built using a voting schema, if the feature is re-observed, it will be promoted; otherwise it progressively decreases its corresponding FSH value. The FSH is based on the human memory model to deal with changing environments and long-term mapping and localization. The human memory model introduce concepts of Short-Term memory (STM), which retains information long enough to use it, and Long-Term memory (LTM), which retains information for longer periods of time or lifetime. If the entries in the STM are continuously rehearsed, they become part of the LTM (i.e. they become more stable). However, this work proposes a change in the pipeline of this model, allowing to any input be part of the STM or LTM considering the input strength (e.g. uncertainty, the Hessian value in the SURF descriptor, or the matching distance). The FSH stores the stability values of local features, stable features are only used for localization and mapping. This innovative feature management approach is able to cope with changing environments, long-term mapping and localization, and also contributes to the semantic environment representation.

The mobile robot perception system plays an important role in SLAM, this must provide reliable information of the robot environment, taking advantage of its surroundings in order to reconstruct a consistent representation of the environment. Taking into account requirements as precision, real-time operation, wide field of view, long-term landmark tracking and robustness to occlusions, this work considered a perception system composed by the combination of 2D Laser Range Finder (LRF) and an omnidirectional camera. Monocular vision sensors have limited field of view, for this reason they are prone to occlusions and limited feature-tracking. Omnidirectional vision solves this problems but it introduces additional non-linearity due the mirror projection model. Despite the fact 2D LRF are limited to planar motion, 2D LRF can be combined with omnidirectional vision sensors providing a sensor model with enhanced information of the environment. This work describes a sensor model based in the extrinsic calibration between a 2D LRF and an omnidirectional camera in order to extract 3D locations of vertical edges. The vertical edges are the features used to describe the appearance of the environment. Data association is considered of crucial importance for any SLAM algorithm, this work proposes a matching method based on the unified spherical model for catadioptric sensors. This matching process improves the Joint Compatibility Branch and Bound (JCBB) test for data association considering the local appearance of the environment vertical edges.

Experimental validation of this approach was conducted using two different SLAM algorithms, and with a long-term dataset collected during a period of 1 year. From the analysis of the experiments carried out the FSH model is able of: filtering out dynamic objects from laser scans and features present in the environment, increasing the map accuracy over the map updates, holding a model of the environment embedding the more stable appearance of the environment, increasing the localization accuracy over the map updates, dealing well with large environments, and reducing the data association effort in long-term runs.

Resumen

Actualmente, los robots móviles necesitan interactuar con ambientes no estructurados. Los robots móviles deben tratar con gente, obstáculos en movimiento, ambientes altamente similares, cambios climáticos, oclusiones, navegación a largo término y interacción humano-robot con el fin de tener altos niveles de autonomía en su toma de decisiones. Estos requerimientos son útiles en robots de servicio diseñados para llevar a cabo tareas de vigilancia, inspección, entrega de paquetes, limpieza y exploración. En estas aplicaciones los robots necesitan recolectar medidas de sus sensores en ambientes complejos y extraer información significativa para llevar a cabo sus tareas.

Localización y construcción de mapas de manera simultánea (SLAM) es considerada una tarea esencial para los robots móviles inmersos en aplicaciones reales sin requerir información previa acerca del ambiente. La comunidad experta en Robótica ha tratado de solucionar el problema de SLAM de diferentes maneras, y usando información métrica o basada en apariencia para representar el ambiente. Esta tesis se ocupa del problema de la localización y construcción de mapas basados en apariencia para robots móviles en ambientes complejos. Esto introduce la pregunta de investigación: ¿Cómo un robot móvil puede actualizar su representación interna del entorno y su localización en éste cuando la apariencia del ambiente es cambiante?

Este trabajo propone un método de localización y construcción de mapas basados en apariencia cuya principal contribución es el Histograma de Estabilidad de las Características (FSH). El FSH es construido usando un sistema de votos, si la característica en el ambiente es re-observada, esta será promovida; de lo contrario ésta progresivamente disminuye su correspondiente valor en el FSH. El FSH en el modelo de memoria humano para hacer frente a ambientes cambiantes y tareas de localización y construcción de mapas a largo término. El modelo de memoria humano introduce conceptos como memoria a corto plazo (STM), la cual retiene información el tiempo suficiente para ser usada, y la memoria a largo plazo (LTM), la cual retiene información por largos periodos de tiempo o de por vida. Si las características almacenadas en la STM son continuamente re-observadas, éstas harán parte de la LTM (i.e. estas características son más estables). Sin embargo, este trabajo propone un cambio en el proceso de memoria humana, permitiendo que cualquier entrada sea parte de la STM o de la LTM al considerar la intensidad de la característica de entrada (e.g. incertidumbre, el valor de la Hessiana en el descriptor SURF, o la distancia de correspondencia). El FSH almacena los valores de estabilidad de características locales, las cuales son solamente usadas para localización y construcción del mapa. Este innovador método de administrar las características del ambiente es capaz de hacer frente a entornos cambiantes, localización y construcción de mapas a largo término y también contribuye a una representación semántica del ambiente.

El sistema de percepción del robot juega un importante papel en SLAM, éste debe proveer de información certera del ambiente del robot, aprovechándose de los alrededores con el fin de construir una representación consistente del ambiente. Teniendo en cuenta requerimientos como precisión, operación en tiempo-real, amplio campo de visión, seguimiento de características por largo tiempo, robustez a oclusiones, este trabajo ha considerado un sistema de percepción compuesto por la combinación de un sensor de rango láser (LRF) en 2D y una cámara omnidireccional. Sensores de visión monocular tienen un limitado campo de visión, por esta razón son propensos a oclusiones y seguimiento de características limitado. La visión omnidireccional resuelve estos problemas per introduce otros como la no-linearidad debido al modelo de proyección en el espejo. A pesar del hecho que los LRF son limitados a movimientos planos, los LRF pueden ser combinados con la visión omnidireccional proveyendo un modelo de sensor con información mejorada del entorno. Este trabajo describe un modelo de sensor basado en la calibración extrínseca entre un LRF y una cámara omnidireccional con el fin de extraer la posición 3D de bordes verticales. Los bordes verticales son características usadas para describir la apariencia del ambiente. La asociación de datos es considerada de crucial importancia para cualquier algoritmo de SLAM, este trabajo propone un método para establecer la correspondencia entre características basado en el modelo unificado de proyección para un sensor catadióptrico. Este proceso de asociación mejora el método JCBB (Joint Compatibility Branch and Bound) considerando la apariencia local del ambiente como bordes verticales.

La validación experimental de este método fue realizada usando dos algoritmos diferentes de SLAM, y con un dataset adquirido durante un largo periodo de 1 año. Del análisis de los experimentos realizados el modelo FSH es capaz de: filtrar objetos dinámicos de los datos del sensor láser y de las características del ambiente, aumentar la precisión del mapa a lo largo de las actualizaciones del mapa, manteniendo un modelo del entorno embebido en la apariencia más estable del ambiente, incrementar la precisión en la localización a lo largo de la actualización de los mapas, tratando así con grandes ambientes y reduciendo el esfuerzo de asociación de datos en ejecuciones de largo término.

Acknowledgements

I would like to thank to my supervisors Xevi and Quim, whose helpful guidance and support encourage me during my PhD studies. They were always willing to help in any problem I could have.

I was very pleased to be part of the Computer Vision and Robotics Group (VICOROB), thanks to all of them who were available to offer their collaboration. Specially, I would show my gratitude to Javi, Joseph, Ricard, Tudor, Angelos (please, do not use your compass within a city downtown), Simone, Pere, Marc, Joseta, Jordi Freixenet and Lluís Magi for his support in the mechanical stuff.

Many thanks to secretaries of the institute (Anna, Montse and Rosa) who managed all the paper work needed for the trips.

It is a real pleasure to thank the colleagues from the MIS (Modélisation, Information et Systèmes) laboratory: Ashu for his friendship and teaching me of Indian culture, Sang for his technical support, Guillaume and Damien for their presence in my speaking and Pauline. I am specially mentioning to Professor El Mustapha Mouaddib for his guidance, collaboration and concern during my stay.

Many thanks to my colleague Eduardo Caicedo, who help a lot in the beginning of my studies taking care of the paper work of my grant, and as the main head of the Perception and Intelligent Systems group support me facing the Electrical and Electronic Engineering School directives.

Finally, but most importantly, this thesis would not have possible without the tremendous support of my family: to my wife always present in good and bad moments; to my father who was in charge of all my issues in Colombia; to my mother, my sister, my brother, my laws and sister in law who were aware of us; and specially to my little girl Catalina, seeing you every morning encourage me to go on.

Table of Contents

List of Figures

List of Tables

List of Algorithms

Glossary

EURON	European Robotics Research Network
CARE	Coordination Action for Robotics in Europe
EUROP	European Robotics Technology Platform
CCC	Computing Community Consortium
CRA	Computing Research Association
FP7-ICT	Seventh Framework Program – Information and Communication Technologies
EUROPA	European Robotic Pedestrian Assistant
UAV	Unmanned Aerial Vehicle
SLAM	Simultaneous Localization and Mapping
LRF	Laser Range Finder
FSH	Feature Stability Histogram
LTM	Long Term Memory
STM	Short Term Memory
MSIS	Mechanically Scanned Imaging Sonar
DVL	Doppler Velocity Log
MRU	Motion Reference Unit
FOG	Fiber-Optic Gyro
USBL	Ultra-Short Base Line
EKF	Extended Kalman Filter
UKF	Unscented Kalman Filter
EIF	Extended Information Filter
L1	Manhattan distance
L2	Euclidean distance
PHLAC	Polar High-Order Local Auto-correlation
DTMO	Detection and Tracking of Mobile Objects
HoC	Histogram of Oriented Cameras
WM	Working Memory
SIFT	Scale-Invariant Feature Transform
SURF	Speeded-Up Robust Features
DCT	Direct Cosine Transform
PCA	Principal Component Analysis
MSER	Maximally Stable Extremal Regions
COLD	COsy (Cognitive Systems for Cognitive Assistants) Localization Database
RANSAC	RANdom SAmple Consensus
LUT	Look-Up-Table
SVD	Singular-Value-Decomposition
DOF	Degrees Of Freedom
LLS	Linear Least Squares
MSE	Mean-Squared-Error

ICP	Iterative Closest Point
JCBB	Joint Compatibility Branch and Bound
FastSLAM	Factored solution to the SLAM problem
ESEIF	Exactly Sparse Extended Information Filter

CHAPTER 1

1. ***INTRODUCTION***

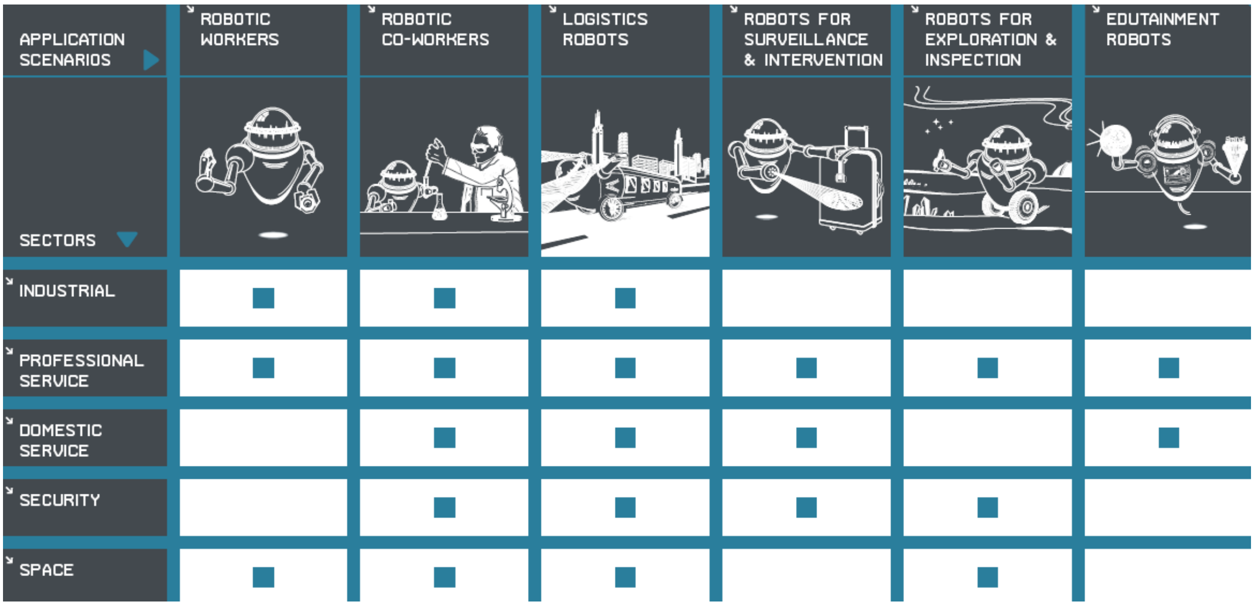
CONTENTS

This thesis addresses the problem of updating the robot internal representation of the environment when its appearance changes over time. This brings together aspects of mobile robots applied to service robotics, localization and mapping, and robot perception systems, which in this work is focused on combining omnidirectional vision and laser range finders. Section 1.1 presents the motivations behind this work. Section 1.2 describes the main objectives of this work. Finally, the structure of this thesis is presented with a short description of each chapter.

1.1. **MOTIVATION**

For centuries Science and Engineering have been working together in manufacturing (goods and tools), security and defense, and infrastructure construction (roads, railways, bridges, cities, dams, etc.) bringing great benefits for mankind. However, for many years, even after the industrial revolution, these economic service sectors did not take into account their impact to the environment and the human lifestyle, perturbing the natural balance that must exist between mankind and its surrounding environment. Throughout the XX century other big economic service sectors were added [The Robot Report, 2011]: healthcare, energy, logistics, entertainment and education. This growth of the responsibilities over the citizens, workers, environment, productivity monitoring, automating the energy acquisition and the dissemination of education at different levels, faces complex technical issues and limitations for mankind due the impossibility to have someone monitoring and / or building something in dangerous environments, or over long exhausting hours of work; dealing with the limited availability of specialized medical care at every home; guarantee the human survival in tough environments like volcanoes, oceans depths, mines, disaster areas, etc.; and handling with heavy loads repeatedly as in harbors, warehouses or stores.

The technical issues presented above points to scientifically interesting research directions. These research directions and their applications have been becoming part of the scientific community and governmental programs for innovation and science in the last decade. The strategic research agenda of the European Robotics Research Network (EURON) [EUROP, 2009] compiled the industry effort and academia knowledge in two big projects: The Coordination Action for Robotics in Europe (CARE) and The European Robotics Technology Platform (EUROP), in order to strength Europe's competitiveness in robotics research, development and global markets. The technical issues described earlier are mapped in Figure 1.1 taking into account sectors as: Industrial, Professional service, Domestic service, Security and Space. The application scenarios help in identifying the bottlenecks to progress, the scientific challenges and research activities to overcome the technological limitations and face the new responsibilities described above.



APPLICATION SCENARIOS ▶ SECTORS ▼	ROBOTIC WORKERS	ROBOTIC CO-WORKERS	LOGISTICS ROBOTS	ROBOTS FOR SURVEILLANCE & INTERVENTION	ROBOTS FOR EXPLORATION & INSPECTION	EDUTAINMENT ROBOTS
INDUSTRIAL	■	■	■			
PROFESSIONAL SERVICE	■	■	■	■	■	■
DOMESTIC SERVICE		■	■	■		■
SECURITY		■	■	■	■	
SPACE	■	■	■		■	

Figure 1.1 Application scenarios of robotics and economical sectors [EUROP, 2009].

According to Figure 1.1 service robotics covers a huge field of action which includes the professional service, domestic service and security sectors, and from the robotic co-workers to the edutainment robots in the application scenarios. In [EUROP, 2009] the application requirements to be fulfilled to successful overcome the scientific challenges are described: sustainability, configuration, adaptation, autonomy, positioning, manipulation and grasping, robot-robot and robot-human interaction, process quality, dependability, physical properties and standardization. Generally, the EURON document encourages the research of system architectures, especially the part of algorithmic software systems which in the end have great impact in the scientific challenges described earlier, instead of increasing the computing power or storage requirements.

On the other hand, the Computing Community Consortium (CCC) and the Computing Research Association (CRA) at USA issued the Roadmap for US Robotics [Computing Community Consortium, 2009] where the application sector was classified as follows: Manufacturing, Logistics, Medical robots, Health care, Services and International cooperation. The service robotics covers from Logistics to Services, having as main goals the cost reduction in the production chain (transportation cost, energy consumption and product handling), improving the economic reliability of the public health system, guarantee full coverage of the public health system (patient rehabilitation, recovery and remote monitoring), home assistance for full time workers and professional assistance (mining, rescue, automated harvesters for agriculture and forestry, and cleaning of large scale facilities). The CCC and CRA documents are focused on the creation of new jobs and companies using the technology in robotics as an important resource for emerging markets.

More recently, the Seventh Framework Program – Information and Communication Technologies (FP7-ICT) of the European Commission was issued [Juretzki, 2011] in the context of the Unit E5 (Cognitive systems, Interaction and Robotics), responding to the current demanding of urban service robots and the necessity of integration cognitive capabilities to the service robots. The last FP7-ICT call had an indicative budget of $ 73M€ and the next ICT call (2012) will have a budget of $ 82M€, showing an increase of 15% in a year-to-year comparison.

Nowadays, thanks to the governmental, academic and industrial initiatives presented above several research projects in the service robotics field have been developed. In the application sector of Logistics (see Figure 1.2), research projects as *DustBot* [Sant'Anna, 2006] developed a system for improving the management of urban hygiene based on a network of autonomous and cooperating robots, embedded in an Ambient Intelligence infrastructure; in [Systems, 2007] the European Robotic Pedestrian Assistant (EUROPA) is presented to develop the foundations for service robots designed to autonomously navigate in urban environments outdoors as well as in shopping malls and shops to provide various services to users including guidance, delivery, and transportation; the CityMobil project [Bouraoui et al., 2011] describes a urban transport system to organize the motorized traffic, reducing congestions, pollution and getting safer driving.

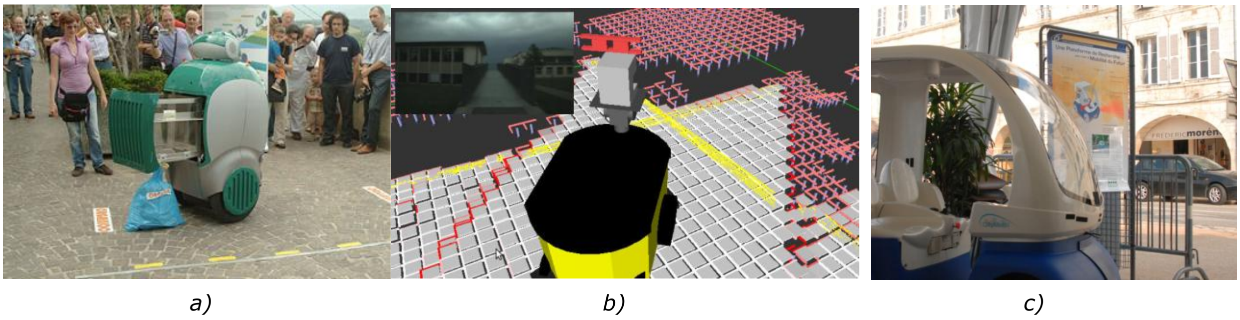


Figure 1.2 a) DustBot showcase. b) Traversability estimation in EUROPA project. c) CityMobil showcase, La Rochelle.

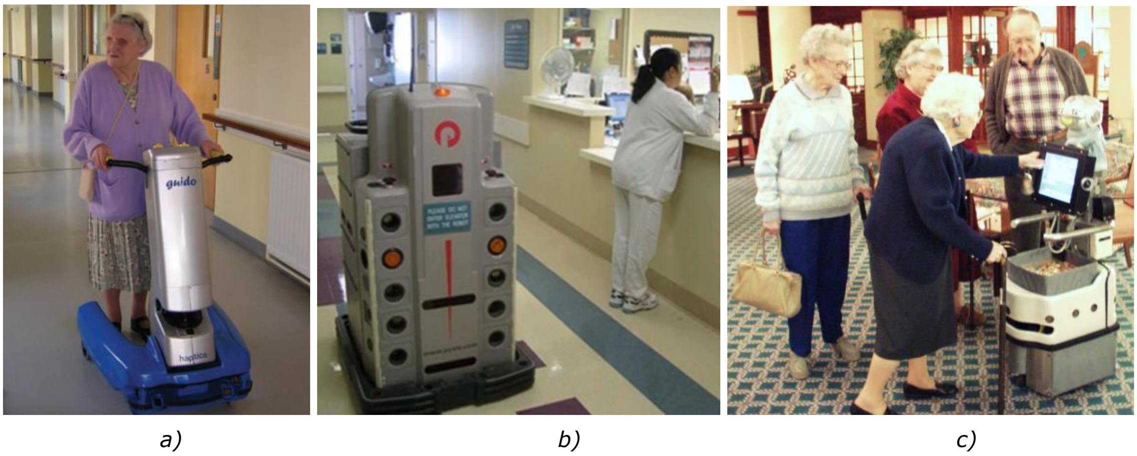


Figure 1.3 a) Smart walker GUIDO. b) RoboMed project. c) NurseBot project.

Regarding robotics co-working in the context of healthcare (see Figure 1.3), [Lacey & Rodriguez-Losada, 2008] presents the GUIDO project a smart walker developed by Haptica (Dublin, Ireland); it is a robotic walking aid that addresses the needs of the frail visually impaired, providing them with a safe means of taking exercise independently as well as navigation assistance in new environments; at Spain, HispaRob [Ministerio de Ciencia e Innovación, 2011] is a technological platform to improve robotic competitiveness in the European market, in this context [Centro Tecnológico CARTIF, 2011] proposed the RoboMed project to automate the patient medicine delivery in hospitals avoiding dosing errors and improving the medicament management. Aging is a global problem and taking care of elderly people suffering from chronic disorders in their everyday life is a challenging task; [Carnegie Mellon University, 2003] describes the NurseBot project, a mobile robot platform which provides intelligent reminding, tele-presence and health monitoring in order to provide a better quality of life to elderly people.

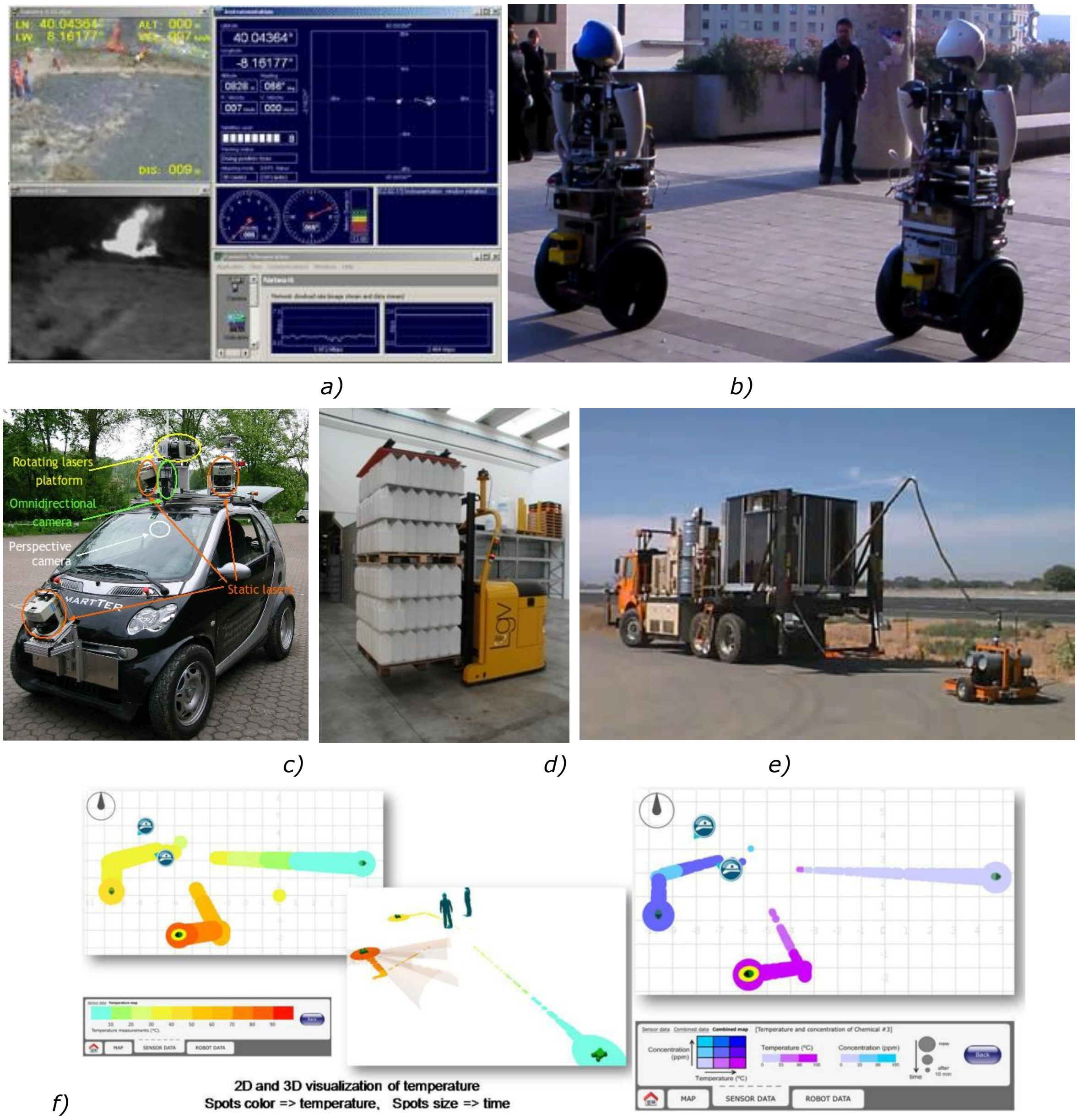


Figure 1.4 a) COMETS project. b) and c) URUS project (surveillance and transport of goods). d) Load handling. e) Route pavement cracks. f) GUARDIANS project.

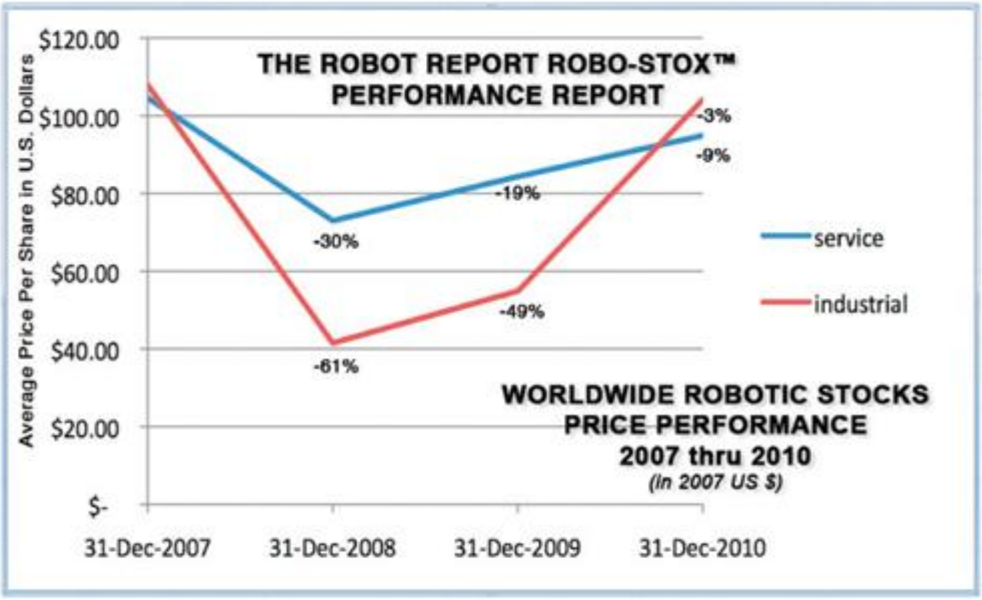
The professional service is an important sector with big impact day to day life of citizens. Robots for surveillance, inspection and exploration are part of the service robotics applications portfolio as depicted in Figure 1.4. Figure 1.4a shows the

COMETS project [Technologies, 2005] where a distributed control system for cooperative activities using heterogeneous Unmanned Aerial Vehicles (UAVs) is implemented for fire monitoring, traffic surveillance and terrain mapping applications. Figure 1.4b and 1.4c show the URUS project [Systems, 2007] focused on designing a network of robots that in a cooperative way interact with human beings and the environment for tasks of assistance, transportation of goods, and surveillance in urban areas. Handling heavy loads is a common task in store malls, warehouses and harbors where human capabilities are physically limited, and in long-term this task is unhealthy. Figure 1.4d shows an automated solution to this need where a skilled laser guided vehicle [C&D, 2011] is in charge of the transportation of different products and pallets, coming from different production lines, from and to the storage and trucking area. Inspection tasks have become in part of the current safety rules and regulations for any civil infrastructure. Figure 1.4e shows a robot used for sealing pavement crack, this maintenance task performed on highways, roads or streets. Figure 1.4f shows the GUARDIANS project [University, 2010] devoted on designing a robot swarm to assist fire fighters in searching a large warehouse. In all these situations, using mobile robots can enhance worker safety, reduce maintenance costs, and improve operational efficiency.

The application scenarios presented above are challenging tasks for mobile robots, since they have to deal with dynamic environments adapting their internal environmental representation (e.g. cluttered outdoor and indoor environments, change in appearance of domestic environments or parking lots); and they have to guarantee autonomous navigation and positioning (e.g. transporting goods or people, medicine delivery, load handling). These robot systems use complex control algorithms which allow them to navigate safely within difficult environments. One of the fundamental components of these control algorithms is the capability of localization and mapping.

Localization is the problem of estimating the robot position given a map, and mapping is the problem of creating a map of the environment given the true position of the robot. When neither the map nor the true robot position is known, the problem is called Simultaneous Localization and Mapping (SLAM) [Durrant-Whyte & Bailey, 2006]. The algorithms developed to solve the SLAM problem [Bailey & Durrant-Whyte, 2006] assume a static environment, however real-world environments as described above are not static, besides the applications mentioned require long-term operation and in many situations large-scale environments.

Summarizing, service robotics has a high impact in our daily lives, improving our quality of life, the efficient usage of resources and the environmental health. However, to achieve these goals great scientific challenges have to be overcome; one of them is improving the robot internal representation of the environment to deal with dynamic environments, which is a common need considering the application sectors described. Finally, according to the Robot Report [Tobe, 2011], there is an increasing need of industrial and service robots (see Figure 1.5), encouraging the robotics community to enable technologies based on cognitive capabilities, to increase the quality of service of robotics systems and expanding their range of application domains.



© 2010 The Robot Report. All rights reserved. Data Management: H. Rousso

Figure 1.5 Worldwide robotic stocks price performance.

1.2. OBJECTIVE

The SLAM problem is considered fundamental to build fully autonomous robotics systems in order to achieve a specific goal without requiring any prior knowledge of the environment. The real-world environments are complex and dynamic, and the robot must adapt to the new working conditions and fulfill its task. To do so, on-board sensors and their associated model are the information used to estimate the robot state and the map of the environment. The essential aim of this work is to enable mobile robots to update its internal representation of the environment to cope with long-term mapping and localization.

The evolution of the SLAM methods is linked to the sensor used and its model. 2D laser range finders (LRF) have been converted into fundamental sensors for mapping and localization. However, 2D LRF are limited to planar motion. Recently, rotation units combined to 2D LRF obtain 3D environmental representations, but these sensors are relatively slow and offer a delayed representation of the environment. Vision sensors have been used over many years due its rich representation of the environment. However, monocular or stereo vision setups have a limited field of view, affecting the robot mapping and localization due occlusions and limited feature-tracking. On the other hand, omnidirectional vision has received special attention recently due to its long-term landmark tracking, its wide field of view and its robustness to occlusions [Gaspar et al., 2007]. Combining these two sensors in order to extract salient features from the environment is one of the auxiliary techniques explored in this work.

The SLAM algorithms usually perform the *prediction-observation-update* cycle. Prediction step depends of the motion model assumed, observation step concerns with the perception system and its model, and the update step is related with the filtering technique and it is high linked with data association. The latter is closely related with the number of landmarks present in the map. The SLAM algorithms consider all the map features equally important, but in long-term operation there is the opportunity to classify the map features according to the number of times a feature is re-observed. Here is where this work introduces the Feature Stability Histogram (FSH) concept, which is a histogram based on a voting schema, if the feature is re-observed, it will increase its corresponding FSH value; otherwise this value is progressively decreased. This research work proposes this feature management approach to deal with changing environments and long-term operation, since the higher FSH value, the more stable a map feature is.

The integration of the FSH in the SLAM framework introduces new challenging issues such as the feature classification criteria to being a more stable feature or not, or the voting schema to implement in order to take into account the feature reliability. This work proposes a straightforward method to solve these questions based on the feature strength (e.g. its uncertainty, the Hessian value in the SURF descriptor [Bay et al., 2008], or the matching distance).

Incorporating salient features extracted from the combination of a 2D LRF and an omnidirectional camera and the FSH approach within a SLAM framework provides advantages that will be explored in this thesis.

1.3. STRUCTURE OF THE THESIS

The research work presented in this thesis is structured as follows:

- **Chapter 2** – Presents the state of the art of the SLAM algorithms and special attention is given to algorithms concerning with long-term operation and dynamic environments. Afterwards, a review of the different features used to represent the environments is introduced. At the end of this chapter a description of the robotic platform used is presented.
- **Chapter 3** – Describes the feature extraction and environment modeling used in this work. First the LRF sensor is considered; then the edge detection into the omnidirectional image is presented. Next, the extrinsic calibration between the omnidirectional camera and the LRF is introduced. Finally, the sensor model is proposed and tested using the FastSLAM algorithm [Montemerlo & Thrun, 2003].
- **Chapter 4** – The FSH approach is presented considering: the original human memory model, the modifications made to it in order to overcome the original limitations, the probabilistic foundations of the FSH model to be able to integrate it into a SLAM solution, the LTM/STM feature classification criteria, the feature pruning approach and the voting schema are explained, some map building considerations are discussed, last the FSH experimental validation is presented.
- **Chapter 5** – Presents the long-term SLAM experiments conducted using the dataset described in Section 2.5.2. This Chapter includes qualitative and quantitative results.
- **Chapter 6** – The conclusions are given, as well as the future research work, the main contributions of this work, the list of publications and the scientific collaborations.
- **Appendix A** – The Jacobians to estimate the corner uncertainty on the image plane are computed. This is an important part of Chapter 3.

CHAPTER 2

2. *BACKGROUND*

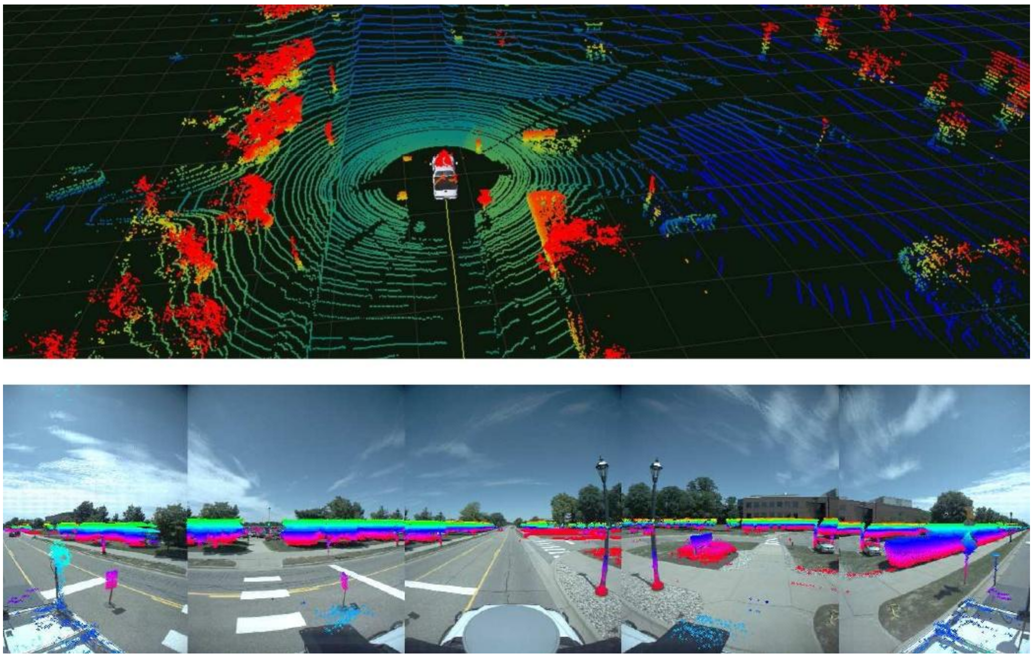
CONTENTS

Over two decades a solution for the SLAM problem (2D or 3D) has been the focus of a lot of research in the robotics community. Application fields of SLAM range from: service, industrial and professional co-workers, security, inspection and space sectors. However, the application fields mentioned also require an autonomous system deployed for long-term operation without human intervention. Section 2.1 gives an overview of the SLAM problem and the implications of long-term operation. The most meaningful SLAM techniques are described in Section 2.2. The state of the art of lifelong (long-term) mapping and localization is described in Section 2.3. An overview of the most common environment representation techniques is shown in Section 2.4. Section 2.5 presents the mobile robot platform used as well as its perception system. Finally, Section 2.6 presents an overview of this work.

2.1. INTRODUCTION

SLAM is a fundamental task for an autonomous robot. The knowledge of the environment, and robot pose on it is essential to guarantee accurate and safe navigation. Initially, in real world applications robots do not have any knowledge of the environment. In this situation, robots have to solve the localization and mapping problem simultaneously, the former estimates the motion of the robot given a map, and the latter builds an environmental representation from observations given the robot motion. The SLAM problem is challenging since observations gathered from the robot sensors and the robot motion are corrupted by noise.

A solution of the SLAM problem involves finding the robot motion and the measurement model. However, the latter has a big impact in the effectiveness and reliability of the solution, because the measurement model determines how the environment is represented, how likely is to re-observe the environment features and how the uncertainty is handled since it depends on the robot sensors used (or a combination of sensors) according with the application needs. For instance, Figure 2.1 shows different kind of applications where the application dictates the set of sensors used. Figure 2.1a shows an autonomous outdoor navigation system [Pandey et al., 2011] which was build to develop technologies for future automotive safety systems; this system involves using panoramic cameras, a 3D laser scanner, radar and IMU sensors in order to estimate the vehicle motion and accurate environmental representations. Figure 2.1b shows the set-up of an inspection dam application [Ridao et al., 2010,] using and underwater robot, which has a set of different sort of sensors compared with the ground vehicles: a mechanically scanned imaging sonar (MSIS), a Doppler velocity log (DVL), a motion reference unit (MRU) assisted with a fiber-optic gyro (FOG), a ultra-short base line (USBL) with a DGPS unit and a camera; this perception system is needed to build an accurate mosaic reconstruction of the wall of a dam to evaluate the state of the concrete or asses the quantity of vegetal residuals obstructing the water flow. Figure 2.1c shows the mapping an localization module of the URUS project [Systems, 2009], where 2D LRF, onboard vision systems and a networked camera system is used to interact with human beings and the environment for tasks of assistance, transportation of goods, and surveillance in urban areas. Figure 2.1d shows a mining exploration system [Thrun et al., 2004] where a robot is used to access to dangerous underground places with structural soundness; this system is composed basically with a 3D LRF depending on these measurements to fully autonomous operation since wireless communication are unreliable.



a)

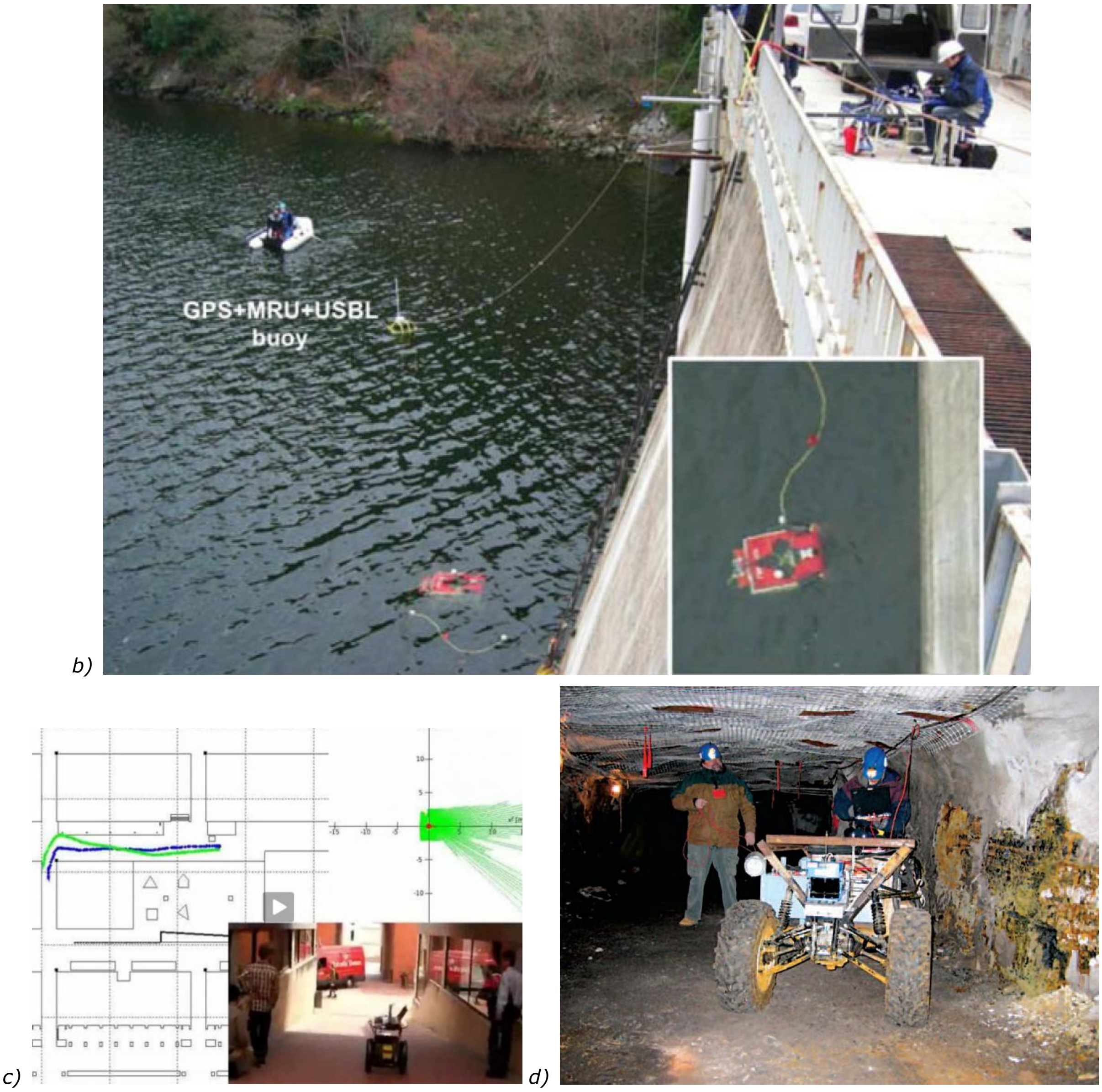


Figure 2.1 SLAM applications. a) Autonomous outdoor navigation [Pandey et al., 2011]. b) Dam inspection [Ridao et al., 2010,]. c) Urban robot navigation [Systems, 2009]. d) Mining exploration [Thrun et al., 2004].

The relationship between the SLAM solution and the perception system is so strong that it allows classifying a SLAM algorithm as: visual based, range and bearing based, feature based and appearance based. Most of the real world SLAM applications shown in Figure 2.1 and those shown in Chapter 1 of this document require long-term operation with little or no human intervention. Many solutions of the SLAM problem presented in Section 2.2 assume a static environment representation, such that moving objects or in general the change of the environment is considered as noise or outliers that should be rejected by the SLAM algorithm [Durrant-Whyte & Bailey, 2006] [Bailey & Durrant-Whyte, 2006].

2.2. **SLAM TECHNIQUES**

In this section the SLAM problem formulation is introduced as a probabilistic estimation, furthermore different kind of SLAM approaches are briefly presented, finally the map representation of the environment is described.

2.2.1. *Problem Formulation*

For self-containment, the SLAM problem formulation is introduced here. The notation followed is the same used in [Durrant-Whyte & Bailey, 2006] and [Bailey & Durrant-Whyte, 2006], which can be summarized as shown in Figure 2.2. In Figure 2.2, $\boldsymbol{x_t}$ is the current robot position at time *t*; a control vector $\boldsymbol{u_t}$ is applied to drive the robot from $\boldsymbol{x_{t-1}}$ to $\boldsymbol{x_t}$ at time *t*; at each position $\boldsymbol{x_t}$ the robot can make an observation $\boldsymbol{z_{t,\,j}}$ corresponding to the *j-th* feature or landmark of the environment $\boldsymbol{m_j}$. Grouping these data, the following sets can be defined: a state vector composed by all past robot positions $\boldsymbol{X_{0:t}} = \{\boldsymbol{x_0}, \ldots, \boldsymbol{x_t}\}$; a sequence of control inputs $\boldsymbol{U_{0:t}} = \{ \boldsymbol{u_1}, \ldots, \boldsymbol{u_t}\}$; the landmarks gathered so far $\boldsymbol{M} = \{ \boldsymbol{m_1}, \ldots, \boldsymbol{m_N}\}$ and corresponding landmark measurements $\boldsymbol{Z_{0:t}} = \{\boldsymbol{z_1}, \ldots, \boldsymbol{z_N}\}$.

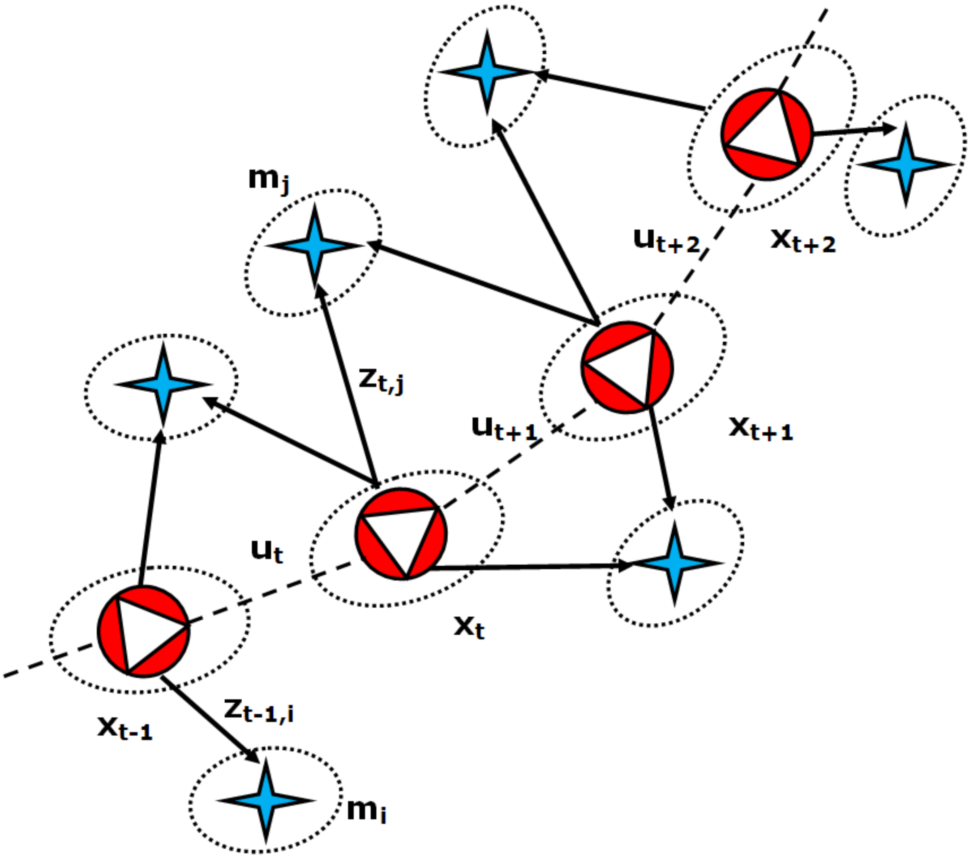


Figure 2.2 Brief description of the SLAM problem and notation.

The SLAM problem solution normally assumes the robot has not any knowledge of the surrounding environment, however the control input sequence and the initial position are the only prior information known by the robot. Using this information in conjunction with the measurements $\boldsymbol{Z}$ sequentially captured by the robot, the aim of the SLAM problem is to build a map of the environment $\boldsymbol{M}$ and simultaneously estimate the position $\boldsymbol{x_t}$ of the robot. Given the correlations existent between the robot position and the landmarks, the concurrent estimation of the map environment and the robot pose is a challenging problem. In addition, control inputs $\boldsymbol{U}$ are not noise free, and different kind of sensors have diverse type of noise sources, if these noise sources are not kept in mind, the result is wrong robot position and map estimations. For this reason, probabilistic formulation of the SLAM problem is popular, since explicitly model different sensors and noise sources [Thrun et al., 2005].

A probabilistic formulation of the SLAM problem requires the estimation of the joint posterior density of the robot pose and the map, given the sensor measurements and the control inputs collected so far. This is expressed in Equation 2.1, which used the Bayes rule applied to the target posterior $P(\boldsymbol{x_t}, \boldsymbol{M} \mid \boldsymbol{Z_{0:t}}, \boldsymbol{U_{0:t}}, \boldsymbol{x_o})$.

$$P(\boldsymbol{x_t},\boldsymbol{M}|\boldsymbol{Z}_{0:t},\boldsymbol{U}_{0:t},\boldsymbol{x_0}) = \alpha P(\boldsymbol{z_t}|\boldsymbol{x_t},\boldsymbol{M})P(\boldsymbol{x_t},\boldsymbol{M}|\boldsymbol{Z}_{0:t-1},\boldsymbol{U}_{0:t},\boldsymbol{x_0}) \qquad 2.1$$

Where, α is a normalizing constant; $P(\boldsymbol{z_t} \mid \boldsymbol{x_t}, \boldsymbol{M})$ is known as the sensor model and describes the probability to get an observation given the robot pose and the map; and $P(\boldsymbol{x_t}, \boldsymbol{M} \mid \boldsymbol{Z_{0:t-1}}, \boldsymbol{U_{0:t}}, \boldsymbol{x_0})$ is known as the time update equation which gives the recursive formulation to the SLAM problem. Using the law of total probability and the Bayes rule, the time update equation can be expressed as Equation 2.2 shows.

$$P(\boldsymbol{x_t},\boldsymbol{M}|\boldsymbol{Z}_{0:t-1},\boldsymbol{U}_{0:t},\boldsymbol{x_0}) = \int P(\boldsymbol{x_t}|\boldsymbol{x_{t-1}},\boldsymbol{u_t})P(\boldsymbol{x_{t-1}},\boldsymbol{M}|\boldsymbol{Z}_{0:t-1},\boldsymbol{U}_{0:t-1},\boldsymbol{x_0})d\,\boldsymbol{x_{t-1}} \qquad 2.2$$

Where, $P(\boldsymbol{x_t} \mid \boldsymbol{x_{t-1}}, \boldsymbol{u_t})$ is known as the motion model describing the state transition between the position $\boldsymbol{x_{t-1}}$ to $\boldsymbol{x_t}$ given a control input $\boldsymbol{u_t}$; and, $P(\boldsymbol{x_{t-1}}, \boldsymbol{M} \mid \boldsymbol{Z_{0:t-1}}, \boldsymbol{U_{0:t-1}}, \boldsymbol{x_0})$ is the posterior one time step earlier. From equations 2.1 and 2.2, the probabilistic SLAM problem formulation is depicted in Equation 2.3.

$$P(\boldsymbol{x_t},\boldsymbol{M}|\boldsymbol{Z}_{0:t},\boldsymbol{U}_{0:t},\boldsymbol{x_0}) = \alpha P(\boldsymbol{z_t}|\boldsymbol{x_t},\boldsymbol{M}) \int P(\boldsymbol{x_t}|\boldsymbol{x_{t-1}},\boldsymbol{u_t})P(\boldsymbol{x_{t-1}},\boldsymbol{M}|\boldsymbol{Z}_{0:t-1},\boldsymbol{U}_{0:t-1},\boldsymbol{x_0})d\,\boldsymbol{x_{t-1}} \qquad 2.3$$

Solving Equation 2.3 implies to know three probability distributions: the sensor model, the motion model and the previous posterior. The following sections describe different solutions to the SLAM problem expressed in Equation 2.3.

2.2.2. *Kalman Filter-based SLAM*

Table 2.1 Summary of Kalman filter based solutions to the SLAM problem.

KF	**Motion Model**	**Time Update**	**Sensor Model and Measurement Update**
EKF	$\boldsymbol{x_t} = \boldsymbol{f}(\boldsymbol{x_{t-1}},\boldsymbol{u_t}) + \boldsymbol{w_k}$	$\bar{\boldsymbol{x}}_{\boldsymbol{t}} = \boldsymbol{f}(\boldsymbol{x_{t-1}},\boldsymbol{u_t})$ $\overline{\boldsymbol{P_t}} = \nabla \boldsymbol{f}\boldsymbol{P_{t-1}}\nabla \boldsymbol{f^T} + \boldsymbol{R_t}$	$\boldsymbol{z_t} = \boldsymbol{h}(\boldsymbol{x_t},\boldsymbol{M}) + \boldsymbol{v_t}$ $\boldsymbol{S_t} = \nabla \boldsymbol{h}\overline{\boldsymbol{P_t}}\nabla \boldsymbol{h}^T + \boldsymbol{Q_t}$ $\boldsymbol{K_t} = \overline{\boldsymbol{P_t}}\nabla \boldsymbol{h}^T\boldsymbol{S}_t^{-1}$ $\boldsymbol{x_t} = \bar{\boldsymbol{x}}_{\boldsymbol{t}} + \boldsymbol{K}_t(\boldsymbol{z_t} - \boldsymbol{h}(\bar{\boldsymbol{x}}_{\boldsymbol{t}}))$ $\boldsymbol{P}_t = \overline{\boldsymbol{P_t}} - \boldsymbol{K_t}\boldsymbol{S_t}\boldsymbol{K_t^T}$
UKF	$\boldsymbol{\chi_t} = \boldsymbol{f}(\boldsymbol{\chi}_{t-1},\boldsymbol{u}_t)$ $\boldsymbol{\chi}_{t-1} = (\boldsymbol{x_{t-1}},\boldsymbol{x_{t-1}} + \alpha,\boldsymbol{x_{t-1}} - \alpha)$ $\alpha = \sqrt{(n + (\beta^2(n+k) - n))\boldsymbol{P}_{t-1}}$ n – Number of sigma points β and k – Scaling parameters	$\bar{\boldsymbol{x}}_t = \sum_{i=0}^{2n} w_m^i \boldsymbol{\chi}_t^i$ $\boldsymbol{P_t} = \sum_{i=0}^{2n} w_c^i(\boldsymbol{\chi}_t^i - \bar{\boldsymbol{x}}_t)(\boldsymbol{\chi}_t^i - \bar{\boldsymbol{x}}_t)^T + R_t$ $w_m^0 = \frac{\beta^2(n+k) - n}{n + (\beta^2(n+k) - n)}$ $w_c^0 = w_m^0 + (1 - \beta^2)$ $w_m^i = w_c^i = \frac{1}{2(n + (\beta^2(n+k) - n))}$	$\boldsymbol{Z}_t = \boldsymbol{h}(\bar{\boldsymbol{x}}_t)$ $\bar{\boldsymbol{z}}_{\boldsymbol{t}} = \sum_{i=0}^{2n} w_m^i \boldsymbol{Z}_t^i$ $\boldsymbol{S_t} = \sum_{i=0}^{2n} w_c^i(\boldsymbol{Z}_t^i - \bar{\boldsymbol{z}}_t)(\boldsymbol{Z}_t^i - \bar{\boldsymbol{z}}_t)^T + \boldsymbol{Q_t}$ $\overline{\boldsymbol{P}_t^{x,z}} = \sum_{i=0}^{2n} w_c^i(\boldsymbol{\chi}_t^i - \bar{\boldsymbol{x}}_t)\,(\boldsymbol{Z}_t^i - \bar{\boldsymbol{z}}_t)^T$ $\boldsymbol{K_t} = \overline{\boldsymbol{P}_t^{x,z}}\boldsymbol{S}_t^{-1}$ $\boldsymbol{x_t} = \bar{\boldsymbol{x}}_{\boldsymbol{t}} + \boldsymbol{K}_t(\boldsymbol{z_t} - \bar{\boldsymbol{z}}_{\boldsymbol{t}})$ $\boldsymbol{P}_t = \overline{\boldsymbol{P_{t-1}}} - \boldsymbol{K_t}\boldsymbol{S_t}\boldsymbol{K_t^T}$
EIF	$\boldsymbol{x_{t-1}} = \boldsymbol{\Omega}_{t-1}^{-1}\boldsymbol{\xi}_{t-1}$ $\boldsymbol{x_t} = \boldsymbol{f}(\boldsymbol{x_{t-1}},\boldsymbol{u_t})$	$\overline{\boldsymbol{\Omega}_t} = (\nabla \boldsymbol{f}\boldsymbol{\Omega}_{t-1}^{-1}\nabla \boldsymbol{f}^T + \boldsymbol{R}_t)^{-1}$ $\bar{\boldsymbol{\xi}}_t = \overline{\boldsymbol{\Omega}_t}\boldsymbol{f}(\boldsymbol{x_{t-1}},\boldsymbol{u_t})$ $\bar{\boldsymbol{x}}_{\boldsymbol{t}} = \boldsymbol{f}(\boldsymbol{x_{t-1}},\boldsymbol{u_t})$	$\boldsymbol{z_t} = \boldsymbol{h}(\boldsymbol{x_t}) + \boldsymbol{v_t}$ $\boldsymbol{\Omega}_t = \overline{\boldsymbol{\Omega}_t} + \nabla \boldsymbol{h}^T\boldsymbol{Q}_t^{-1}\nabla \boldsymbol{h}$ $\boldsymbol{\xi}_t = \bar{\boldsymbol{\xi}}_t + \nabla \boldsymbol{h}^T\boldsymbol{Q}_t^{-1}(\boldsymbol{z_t} - \boldsymbol{h}(\bar{\boldsymbol{x}}_{\boldsymbol{t}})$ $+ \nabla \boldsymbol{h}\bar{\boldsymbol{x}}_{\boldsymbol{t}})$

In general, the Kalman filter based solutions to the SLAM problem approximates the joint posterior (Equation 2.1) as a Gaussian distribution as well as the noise sources. This section briefly describes three popular Kalman filter based approaches: the *Extended Kalman Filter* (EKF) [Thrun et al., 2005], the *Unscented*

Kalman Filter (UKF) [Julier, 1997] and the *Extended Information Filter* (EIF) [Thrun et al., 2003]. Table 2.1 summarizes these SLAM solutions in terms of the motion model, the sensor model, the time update equations and the linearization method applied to the motion and sensor model.

The EKF does not assume a linear state transition and measurement functions, if these are non-linear functions they are linearized using a Taylor series expansion. EKF represents the joint posterior density by a multivariate Gaussian distribution. However, it strongly depends on the state uncertainty, since the higher the state uncertainty, the bigger is the posterior density distortion and the approximation errors.

For this reason, another linearization method is applied in the UKF algorithm using a weighted statistical linear regression process. The basic idea behind the UKF is to extract a set of points around the state mean (the sigma points), compute their exact value using the state transition function and then the linearized Gaussian is computed from the extracted sigma points.

The measurement update of EKF and UKF is a complex step, since it requires matrix inversion which could be expensive from the computational point of view. This motivates the canonical parameterization of the Gaussian belief which is considered as the dual of the Gaussian representation based in their moments. The canonical representation is given by the information matrix (the inverted covariance matrix) and the information vector (the information matrix multiplied by the state mean). The pipeline process is the same as in the EKF and UKF; however the EIF presents some comparative advantages as the simplicity for computing conditional probabilities or the simplicity of the Kalman updating equations. Despite these advantages, a drawback of the EIF is the inversion of the information matrix to recover the robot state.

The formulation of the SLAM problem involves some important issues as: convergence and consistency, scalability and data association. Convergence and consistency are related with linearization, from this point of view UKF computes the joint posterior with better accuracy than EKF, but the underlying problem of linearization of the state transition and measurement equations is not completely avoided. One of the well known problems of Kalman filters family is scalability; that is the ever increasing number of landmarks, such that soon or later there are not enough computer resources to update the map and the robot state in real time. This is an important problem for service robots, which have to operate over long periods of time. The data association problem is crucial to find correlations between the current features and the landmarks already stored in the map. According with Table 2.1 re-observing landmarks is fundamental to improve the robot state and the map. However, Kalman filters family work on a single joint posterior hypothesis, which means that incorrect associations of landmarks cause important or dramatic errors in the robot state and the map.

Multi-hypothesis tracking and considering the state transition, the measurement function and the joint posterior as non-Gaussian distributions are part of the particle filter framework, which is introduced in the next section.

2.2.3. *Particle Filter SLAM*

Particle filter techniques represent the state estimate probability distribution in a non-parametric form, compared with the Gaussian representation of the Kalman based techniques. In this case, the state estimate is represented through a set of samples; despite the fact that it is an approximated representation, using a suitable number of samples a wide number of probability distributions can be represented.

According with this idea and keeping in mind that the observations are conditionally independent; the Equation 2.3 can be factored in a different way as depicted in Equation 2.4.

$$P(\boldsymbol{x_t}, \boldsymbol{M}|\boldsymbol{Z_{0:t}}, \boldsymbol{U_{0:t}}, \boldsymbol{x_0}) = P(\boldsymbol{x_t}|\boldsymbol{Z_{0:t}}, \boldsymbol{U_{0:t}}, \boldsymbol{x_0}) \prod_{n=1}^{N} P(\boldsymbol{m}_n|\boldsymbol{x_{0:t}}, \boldsymbol{Z_{0:t}}) \qquad 2.4$$

Where, *N* is the total number of landmarks. Equation 2.4 states a high dimensional state-space if a set of samples want to be drawn from this distribution. Normally a Rao-Blackwellisation [Durrant-Whyte & Bailey, 2006] procedure is applied in order to reduce the state-space dimensionality. This is achieved representing analytically the rightmost part of Equation 2.4 and sampling the leftmost part of Equation 2.4.

In general form, the basic algorithm of a particle filter is as follows:

1. A state sample $\boldsymbol{x}_t^k$ is drawn from the motion model $P(\boldsymbol{x_t} \mid \boldsymbol{x_{t-1}}, \boldsymbol{u_t})$:

$$\boldsymbol{x}_t^k \sim \gamma(\boldsymbol{x_t}|\boldsymbol{x}_{t-1}^k, \boldsymbol{u_t}) \qquad 2.5$$

   Where, $\boldsymbol{x}_{t-1}^k$ is the posterior estimate for the robot pose at *t-1* for the *k-th* particle. Given the robot position, all the observed features for each particle are updated and those non-observed features are copied without modification.

2. The measurements are incorporated computing the importance factors $w_t^k = P(\mathbf{Z}_\mathrm{t}|\mathbf{x}_\mathrm{t}^\mathrm{k})$, which in general are defined as the ratio of the target distribution and the proposal distribution. The target distribution is the true posterior distribution $P(\boldsymbol{x}_t|\boldsymbol{x}_{0:t}, \boldsymbol{Z}_{0:t})$. The proposal distribution approximates the target distribution, and it is used to generate the initial set of particles (e.g. Equation 2.5).
3. The joint posterior approximation of a particle filter occurs in the re-sampling process. Here, a low variance sampler algorithm draws with replacement *N* particles from the particle set. The probability of drawing a particle depends on its importance weight. Such that, at the end, the particles are distributed approximately as the posterior distribution.

Particle filters have mapped large areas [Montemerlo & Thrun, 2003] keeping in mind multiple map hypothesis, which is well suited for the SLAM problem. However, some known drawbacks are [Thrun et al., 2005]: memory use, particle deprivation and scalability.

2.2.4. *Appearance-based SLAM*

Appearance-based methods for mapping and localization have gained increasing attention in recent years. These mapping techniques can be classified according to the environmental representation method used. Table 2.2 reviews some remarkable studies of appearance-based mapping and localization, where the type of sensor used was also kept in mind.

Appearance-based mapping approaches model the environment extracting feature descriptors, which are well known in the computer vision community as SIFT and SURF features [Angeli et al., 2008] [Segvic et al., 2009] [Newman et al., 2006] [Zivkovic et al., 2007] [Andreasson et al., 2008] [Dayoub et al., 2011], Discrete Cosine Transforms (DCT) [Porta & Krose, 2004], multidimensional histograms (color, edge, texture, gradient and rank) [Zhou et al., 2003], and Fourier transforms [Gross et al., 2005]. All these methods use L1 (Manhattan distance) or L2 (Euclidean distance) metrics to match feature descriptors, and in general terms the mapping and localization process is done assuming a static environment, where its local appearance is not updated over time.

Other approaches take advantage of the type of sensor used (omnidirectional or standard camera, LRF, etc.) and environmental structures surrounding the mobile robot in order to represent the environmental appearance. Intuitively, the appearance-based model of the environment describes the environment as it is, taking advantage of its natural features. That is the case of [Sujan et al., 2006] which uses image information content implementing quad-tree decomposition to find interesting places, or [Linaker & Ishikawa, 2006] which proposes a new descriptor focused on omnidirectional images called Polar High-Order Local Auto-correlation (PHLAC) and tested using a particle filter framework. An interesting case is reported in [Goedemé et al., 2007] and [Murillo et al., 2007] which use omnidirectional vision and vertical lines as landmarks projected on the image plane. These landmarks have promising results because in omnidirectional vision they can be tracked for long baselines, decreasing the probability of being occluded and increasing the probability of being visible.

In contrast to the above approaches, which use one sensor, [Gallegos & Rives, 2010] uses sensor fusion between an omnidirectional camera and a 2D LRF. This approach takes advantage of the metric information provided by the LRF and mixes it with the omnidirectional vision [Bacca et al., 2010].

Table 2.2 Review of appearance-based mapping and localization approaches.

Appearance-based Mapping and Localization		
Rich Sensor Information	**Environment Representation**	
	Classical	**Closer to Environment Appearance**
• Standard camera [Angeli et al., 2008] [Porta & Krose, 2004] [Zhou et al., 2003] [Segvic et al., 2009] [Sujan et al., 2006] • LRF (2D and 3D) [Magnusson et al., 2009] [Nieto et al., 2007] [Nachter & Hertzberg, 2008] • Standard camera + LRF [Mozos et al., 2006] • Omni-camera [Gaspar et al., 2007] [Linaker & Ishikawa, 2006] [Goedemé et al., 2007] [Gross et al., 2005] [Dayoub et al., 2011] [Murillo et al., 2007] [Zivkovic et al., 2007] [Andreasson et al., 2008] • Omni-camera + LRF [Gallegos & Rives, 2010]	• Scale-Invariant Feature Transform (SIFT) [Angeli et al., 2008] [Segvic et al., 2009] [Newman et al., 2006] [Zivkovic et al., 2007] [Andreasson et al., 2008] • Speeded-Up Robust Features (SURF) [Dayoub et al., 2011] • Direct Cosine Transform (DCT) [Porta & Krose, 2004] • Principal Component Analysis (PCA) [Gaspar et al., 2007] • Multidimensional Histograms [Zhou et al., 2003] • Fourier [Gross et al., 2005] • Normal Distribution Transform (NDT) [Magnusson et al., 2009]	• Quad-tree [Sujan et al., 2006] • Object saliency and Iterative Closest Point (ICP) [Nieto et al., 2007] [Nachter & Hertzberg, 2008] • Machine learning approaches [Mozos et al., 2006] • Polar Higher-order Local Auto-Correlations (PHLAC) [Linaker & Ishikawa, 2006] • Vertical edges [Goedemé et al., 2007] [Murillo et al., 2007] [Gallegos & Rives, 2010]

The goal of appearance-based methods is to use rich information of color, texture, or environment structure in order to find an association between two data sets, such that reducing the false positives in robot localization. This technique has been successful used in [Cummins & Newman, 2009] to map and detect loop closing situations over large and complex areas. In the context of the SLAM problem, appearance-based methods are useful to predict a possible landmark association which afterwards could be metrically validated. In addition, close a large loop in Kalman based and Particle Filters techniques is a very challenging task; however, appearance-based methods provides additional discrimination information which can be taken into account to improve the detection of loop closures.

2.2.5. *Map Representation*

Once the robot state and the map of the environment given by the observed landmarks are estimated, the robot's internal representation of the environment can be essentially described in four categories:

1. *Topological* – Topological maps are compact, consume less computer memory, can be stored in efficient data structures, and speed up the navigation process. They use graphs for environmental modeling, where each vertex often contain appearance descriptions of the environment as well as other relevant information; and each edge indicates a traversable path between two environment locations, they often include information obtained from odometry, epipolar geometry or in general the resulting change in the position given by the motion model.
2. *Metric* – Metric maps represent the environment using specialized features which depends on the sensor model and the distance measurement method used. Features such as lines, planes, points and curves are often used to represent the map. This kind of maps has the advantage to provide direct information about the free collision trajectories for safe robot navigation.
3. *Hybrid* – Hybrid map representation combines the better of two worlds: the topological and metric maps. Topological maps can represent large areas in a compact form, which makes them a suitable solution for large scale mapping situations. Metric maps directly consider the state and landmark uncertainty providing the means to estimate spatial relationships between the topological maps elements.
4. *Occupancy Grids* – Occupancy grid map representation generates a map able for path planning and navigation, and generally they are computed after solving the SLAM problem since occupancy grid maps require all the measurements and the robot path estimate. Occupancy grid maps divides the space into many grid cells of fixed size, then each cell contains basically three possible values: one for occupied, zero for free and p (e.g. 0.5) for unknown.

Choosing a topological representation of the environment encourages the identification of key areas and the paths to reach them. Metric and hybrid map representations depends on the map features, that is, the sensor model; in this case, given the high dimensionality of the SLAM problem feature extraction is better than raw sensor data. Occupancy grids use raw sensor data (LRF in most cases) providing local and global estimation of the free space.

2.3. **LIFELONG MAPPING AND LOCALIZATION**

These days mobile robots are needed to interact within non-structured environments. They must deal with people, moving obstacles, perceptual aliasing, weather changes, occlusions and robot-human interaction in order to have high levels of autonomy from a decision-making point of view, and to resolve mapping, localization and navigation issues as well as possible. These requirements are useful for service robots designed to conduct surveillance, inspect, deliver, clean and explore. In addition to localization, mapping and navigation problems, they have to guarantee a high level of autonomy through long-term navigation using stable features, which can be extracted from the environment structure or detected using artificial landmarks.

Typical techniques to solve the SLAM problem assume a static environment, and they build a map without taking into account the real-world conditions described above. This drives us to the initial question stated in the abstract: How can a mobile robot update its internal representation of the environment and its location on it when the appearance of the environment is changing?. In the two past decades many approaches to resolve the SLAM problem have been proposed, and basically the last eight years the SLAM problem in dynamic environments has been increasing the attention of the robotics community. Table 2.3 presents a review of the most relevant techniques with respect to SLAM in dynamic environments.

Table 2.3 Review of SLAM techniques in dynamic environments.

SLAM in Dynamic Environments	
Classification	**Approaches**
Memory management models	[Dayoub et al., 2011] STM/LTM human memory model to update reference views in hybrid topological maps. [Labbe & Michaud, 2011] WM/LTM human memory model for loop closure detection. [Abrate et al., 2010] Uses recency weighted averaging to detect changes in the environment.
Landmark visibility and rating	[Pirker et al., 2011] Uses the Histogram of Oriented Cameras descriptor to extract landmark visibility and reduce the matching candidates. [Hochdorfer & Schlegel, 2009] Quantify landmarks contribution to localize the robot based on their visibility and rating. [Hochdorfer et al., 2009] Basically same approach described in [Hochdorfer & Schlegel, 2009] but it uses DBSCAN clustering algorithm for better results. [Andrade-Cetto & Sanfeliu, 2002] Combines the landmark strength and EK filtering for map updating.
Detecting/using dynamic objects	[Meyer-Delius et al., 2010] Uses temporary maps built from unexpected objects in the environment. [Burgard et al., 2007] Probabilistic framework to identify spurious measurements and then filter them out from the map process. [Wang et al., 2003] Probabilistic framework for the detection and tracking of mobile objects and the SLAM problem. [Montesano et al., 2005] Detection and tracking of mobile objects is included in the SLAM process. Based on the work presented in [Wang et al., 2003]
Dynamic changes integrated in SLAM	[Glover et al., 2010] Fuses the probabilistic local feature based data association method of FAB-MAP with the pose cell filtering and experience mapping of RAT-SLAM. [Milford & Wyeth, 2009] Presents the RAT-SLAM approach which is a biologically inspired solution for the SLAM problem.
Pruning methods	[Kretzschmar et al., 2010] Discards observations that do not provide relevant new information with respect to the map built so far. Observation removal is based on the entropy reduction caused. [Konolige & Bowman, 2009] Learns cluster of views that represent a persistent visual environment and it removes those views which are redundant.
Multiple map representations	[Biber & Duckett, 2009] Uses a multiple map representation over time using the median of recency instead of the recency weighed averaging to update the map.

The most common strategy to deal with dynamic environments is *detecting dynamic objects* and considering them as spurious measurements. A seminal work using 3D LRF in outdoor environments was proposed in [Wang et al., 2003], here the detection and tracking of mobile objects (DTMO) is performed in advance to SLAM. The probabilistic framework proposed in [Wang et al., 2003] is based on estimating the constant velocity motion of mobile objects and differentiating static from moving observations. In contrast, [Burgard et al., 2007] proposes a probabilistic framework based on occupancy grid maps estimating for each individual beam whether of or not it has been reflected by a dynamic object. Once these laser beams are identified, they are filtered out from the range registering process. This technique also learns quasi-static environmental configurations clustering local grid maps. [Burgard et al., 2007] and [Wang et al., 2003] do not use the dynamic changes in the environment in the SLAM process, however [Meyer-Delius et al., 2010] temporary integrates the observations caused by semi-static objects in the SLAM process. Nevertheless, the aim of [Meyer-Delius et al., 2010] is not generating a consistent representation of the environment, since temporary maps are discarded as soon as the current observation cannot be explained with the current temporary map.

Service robots are normally operated by users who are not skilled robotics programmers. Thus, a service robot has to learn relevant information of the environment where it is deployed. One way to do it is through *landmark visibility* and rating. In this context, [Andrade-Cetto & Sanfeliu, 2002] developed and EKF-based map building system which incorporates the measurement of landmark strength and quality allowing the elimination of unreliable observations. However, the removal criteria were based on user supplied strength and quality thresholds. In [Hochdorfer & Schlegel, 2009] and [Hochdorfer et al., 2009] the landmark quality is quantified based on its contribution to the ability of the robot self-localization. In these approaches the landmark uncertainty and its observability are taken into account to quantify the landmark quality, and then used to avoid the ever growing number of landmarks. A clustering step is performed to indentify the landmark spatial distribution, this computes their observability. Another interesting idea is presented in [Pirker et al., 2011], where the landmark visibility is computed using the Histogram of Oriented Cameras (HoC) in order to reduce the average of matching candidates per frame.

A logical consequence of landmark rating is *pruning* of unreliable or useless landmarks. In this sense, [Kretzschmar et al., 2010] is focused on removing observations which do not provide relevant information with respect to the map built so far. To do so, the entropy of an observation given the past measurements is computed to obtain its information gain, if this information gain is zero the observation is discarded. In [Kretzschmar et al., 2010] a graph-based SLAM solution is used and the observations are the node positions. In [Konolige & Bowman, 2009] a lifelong mapping solution is proposed using the FrameSLAM approach [Konolige et al., 2009] and a view deletion model based on removing views with low matching measures. In spirit this work is similar to [Burgard et al., 2007], but in this case learning clusters of views is done to represent similar and persistent places of the environment. These pruning techniques have the tendency to introduce delayed map updates, since the optimization of large maps could take more time than the observation time window.

A characteristic which has gained attention for lifelong mapping is *multiple map representations*. A seminal contribution on this context was [Burgard et al., 2007] and afterwards [Konolige & Bowman, 2009]. Nevertheless, a particularly different approach is done in [Biber & Duckett, 2009], where the environment is simultaneously represented at multiple timescales and at each timescale a different learning rate is used to obtain a map of the environment. However, a drawback of [Biber & Duckett, 2009] is that the timescales were experimentally found, which limits the robot autonomy compared with [Burgard et al., 2007] and [Biber & Duckett, 2009].

In general, the approaches described so far solve the SLAM problem independently from the detection and tracking of dynamic changes in the environment. Decomposing the whole problem in two separate estimations is understandable given the high dimensionality of the estimation problem, and in many situations the dynamic objects do not provide relevant localization information for the robot. However, in [Milford & Wyeth, 2009] the RatSLAM algorithm is presented and its system architecture is inspired on the rat hippocampus place cells. RatSLAM is composed by an experience space, each experience is associated with a local view cell and each local view cell is associated with a set of pose cells. In RatSLAM, the map is maintained up to date due continuously experience adding. The data association method in RatSLAM is strongly dependent on lighting conditions, on the other hand FAB-MAP [Cummins & Newman, 2009] is a probabilistic appearance-based mapping technique with a reliable data association approach, as a result in [Glover et al., 2010] is presented a hybrid mapping system combining the better of both algorithms but integrated them in one SLAM solution.

Biologically or psychologically inspired models have been used over many years in the robotics community. For instance, the human memory model proposed by [Atkinson & Shiffrin, 1968] or [Baddeley, 2003], and the weighed recency averaging approach are good examples of *memory management models* inspired on psychological and neuroscience models. In [Abrate et al., 2010] the weighted recency averaging approach is used to update the map of the environment; however the results presented only consider simulations and small environments. In [Dayoub et al., 2011] the Atkinson and Shiffrin memory model [Atkinson & Shiffrin, 1968] is considered in order to update the reference view of a particular place. However, they assume the robot is able to self-localize using other means, since their main goal is to maintain the reference views of the topological map up to date. Using the human memory model proposed in [Baddeley, 2003] a real-time loop closure detection approach is presented in [Labbe & Michaud, 2011] evaluating the number of times a locations has been matched or consecutively viewed. This evaluation is done using a Bayesian filter to estimate the probability that the current topological location matches one of an already visited stored in the Working Memory (WM).

The lifelong SLAM approaches reviewed in this section can integrate the dynamism of the environment in the estimation process or not. Moreover, given the high number of SLAM solutions available nowadays, it is worth designing and implementing lifelong SLAM methods in a way that it can be applied to the current SLAM solutions.

2.4. ENVIRONMENT MODELING

As described in the first section of this chapter, the perception system plays a crucial role in how the SLAM problem is solved. Furthermore, the perception system defines how the environment is modeled and what kind of resources can be used for data association, or loop closure situations. The robot perception system yields a considerably high amount of raw data, if it is directly processed the data high dimensionality will make any estimation procedure unfeasible for real-time operation. On the other hand, feature extraction allows a trade-off between detecting relevant information from the environment and keeping low the amount of sensor data and the computational time required for processing this information.

The features extracted from sensor data range between low-level to high-level features. For instance, corners, circles and lines extracted from 2D range data can be considered as low-level features; general image descriptors, object descriptors, vertical edges image descriptors and image histograms can be considered as high-level features. In this work, the LRF and the omnidirectional camera are the two main sensors used. The following two sections will describe the most common features used with these kinds of sensors.

2.4.1. *Range Data Features*

Laser range finders have been used over many years in the robotics community. They provide direct range and heading measures centered in the robot frame, resolving the scale problem presented if the scene is analyzed using a sequence of images. Range data can be 2D or 3D, 2D range data is directly captured from a LRF, but 3D range data is captured using a standard LRF mounted on a pan/tilt mechanism.

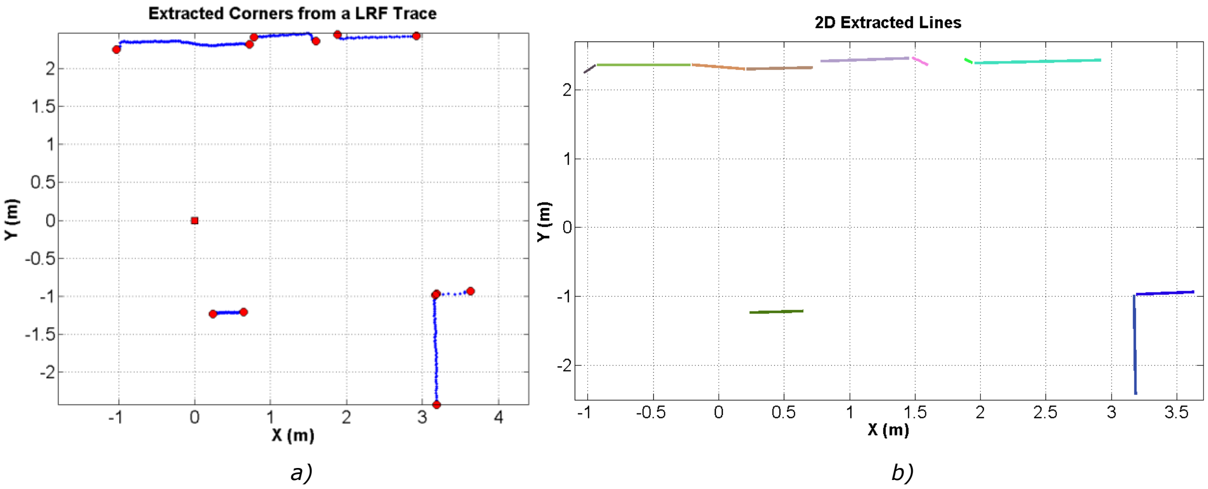


Figure 2.3 a) Detected corners on a LRF trace. Circle-shaped marks represent the extreme points of the homogeneous regions. The robot frame origin is represented by the rectangular-shaped point. b) Line segments estimated from the same LRF trace.

Typical features extracted from 2D range data include corners, circles and line segments. Corner features can be estimated as a tracking-like problem [Castellanos & D., 1999], which can be formulated using a Kalman filter. Figure 2.3a shows an example of a LRF trace and the detected corners represented in the circle-shaped marks. Afterwards, line segments can be estimated using the sub-divided groups of laser points using and iterative line fitting algorithm. A detailed comparison of line segment extraction algorithms can be found in [Nguyen et al., 2005]. Split-and-merge [Pavlidis & Horowitz, 1974] is the most popular algorithm to accomplish this task. In this implementation, the basic algorithm was modified to increase its robustness. One way to do that is modifying the cost function, in this work the cost function was implemented using the dot product of each line segment point and the line segment normal. Figure 2.3b shows the line segments obtained.

However, 2D range data is constrained to planar motions which in indoor environments are not an issue, but in outdoor environments this assumption is not longer valid. Although, 3D range data allows overcomes this constraint, the amount of data to analyze is increased considerably. For this reason, fast plane segmentation techniques have been implemented to reduce the dimensionality problem. In [Nachter & Hertzberg, 2008] a plane segmentation technique based on the normal computed from raw data is presented (Figure 2.4). Nevertheless, this method requires an exhaustive processing of the raw data. On the other hand, plane segmentation techniques based on cell decomposition are becoming popular [Weingarten & Siegwart, 2005] [Xiao et al., 2011]. In this case, the 3D range point cloud is decomposed by regular cells, a local plane is fitted and then a recursive region growing algorithm is used to fuse similar planes [Weingarten & Siegwart, 2005]. The latter technique does not estimate the resulting plane parameters due the region growing algorithm used, but in [Xiao et al., 2011] a new recursive region growing algorithm is proposed in order to obtain the final plane parameters.

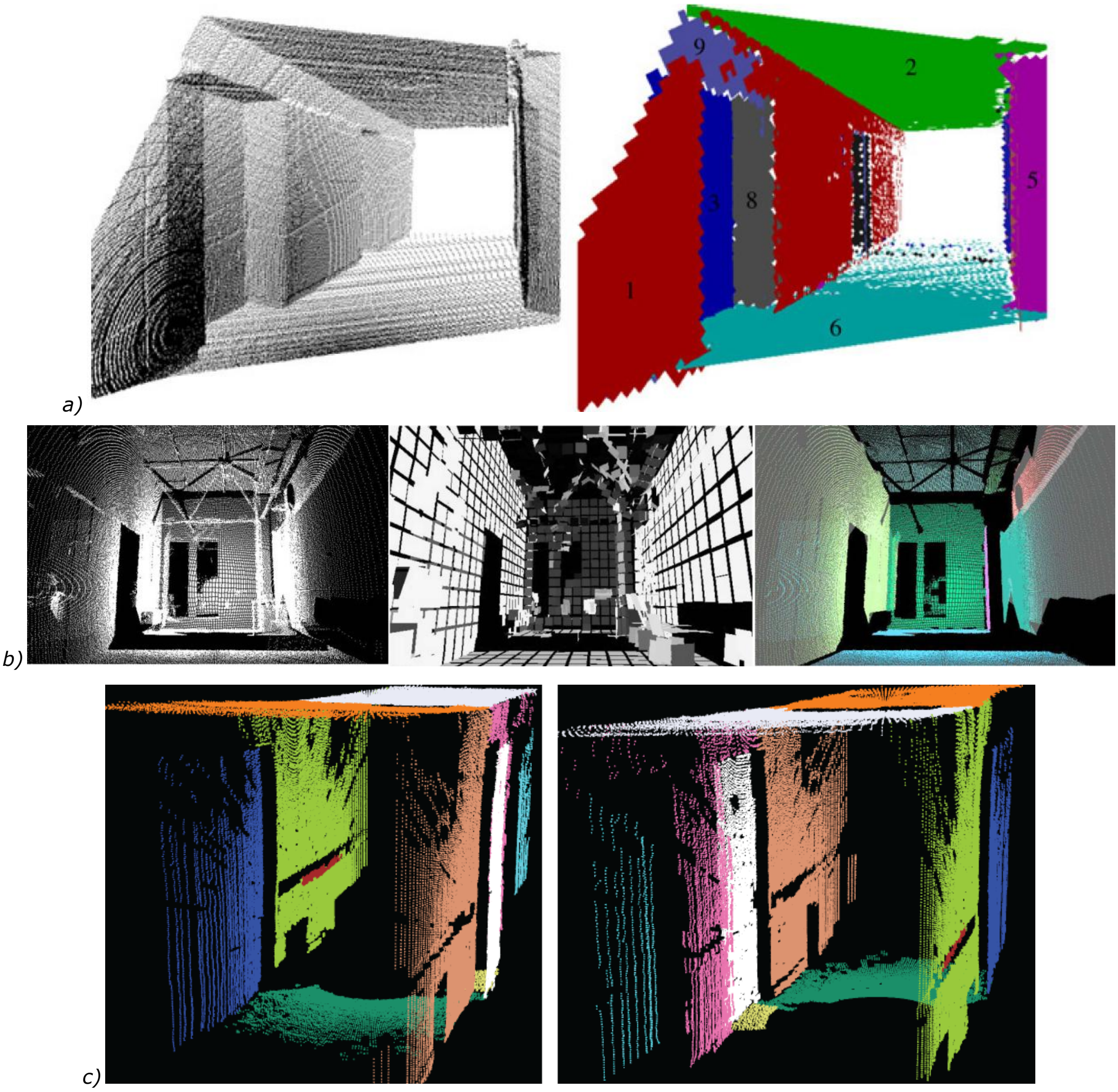


Figure 2.4 a) 3D range point cloud and extracted planes [Nachter & Hertzberg, 2008]. b) Plane segmentation proposed in [Weingarten & Siegwart, 2005]. c) Plane segmentation proposed in [Xiao et al., 2011].

2.4.2. *Image Features*

2D or 3D range data provide direct geometric information of the environment and a reasonable level of the environment structure. Nevertheless, a vision system provides richer appearance information as color, texture, edges (horizontal, vertical and curved) and shapes. Despite the fact that geometric information is difficult to extract and often requires more than two consecutive views to obtain accurate measurements, vision plays an important role in SLAM. Table 2.4 shows a general classification of some image features extraction techniques widely used. Table 2.4 divides the feature extraction techniques used in standard images, and those used in omnidirectional images.

The most relevant image feature is its color, texture and shapes. A common way to quantify the color distribution along the image is using histograms. For instance, image retrieval systems have been using histogram matching for long time due their compact representation and their invariance to small rotations and translations. In [Zhou et al., 2003] not only color histograms are extracted, but edge, texture and gradient histograms are also computed for appearance-based robot mapping and localization. However, in general histograms are not invariant when important movements are involved. In addition, histograms define a global property of image; they do not allow extract interesting points.

Table 2.4 General classification of image features in standard and omnidirectional images.

Features in standard images		**Features in omnidirectional images**	
Classification	**Approaches**	**Classification**	**Approaches**
Histogram-based	*[Zhou et al., 2003]* Multidimensional histograms (color, edge, texture, gradient and rank) applied to robot localization.	*Scale space based*	*[Scaramuzza et al., 2009]* Vertical edge descriptor in omnidirectional images. [Hansen et al., 2007] Scale space images as the solution to the heat diffusion equation on the sphere which is implemented in the frequency domain using spherical harmonics. [Lourenco et al., 2010] Enhanced SIFT detector by introducing radial distortion into the scale space computation. [Arican & Frossard, 2010] Scale invariant feature for omnidirectional images.
Corner based	*[Harris & Stephens, 1988]* Harris corner detector. *[Rosten & Drummond, 2006]* FAST		
Scale space based	*[Lowe, 2004]* SIFT descriptor. [Bay et al., 2008] SURF descriptor. [ForssÃ©n & Lowe, 2007] MSER descriptor.	*Custom designed*	*[Linaker & Ishikawa, 2006]* Polar High-Order Local Auto-correlation descriptor. [Hansen et al., 2010] Proposes a spherical diffusion using stereographic projection. [Goedemé et al., 2007] Proposes a vertical edge descriptor using the global colour moments and seven intensity variants using DCT. [Murillo et al., 2007] Features from [Goedemé et al., 2007] for robot localization using omnidirectional images and pyramidal matching. [Briggs et al., 2006] Features are extracted from 1D panoramas using interest operators in scale-space. [Tapus et al., 2004] Finger print of places is introduced. It is a circular list of features extracted from panoramic images. [Guerrero, 2011] Scale-space computation using the Laplace-Beltrani operator.
Sub-space based	*[Porta & Krose, 2004]* Direct cosine transform (DCT) applied to appearance-base localization. [Gaspar et al., 2007] Principal component analysis (PCA) applied to robot localization. [Porta et al., 2005] PCA applied to appearance-based localization. [Gross et al., 2005] Fourier transforms for adaptive appearance maps. [Siggelkow, 2002] Haar integrals applied to robot localization.		

On the other hand, corner and scale-space based approaches allow extracting points of interest (Harris corners [Harris & Stephens, 1988], FAST detectors [Rosten & Drummond, 2006], SIFT/SURF/MSER descriptors [Lowe, 2004] [Bay et al., 2008] [ForssÃ©n & Lowe, 2007]) from images; in this way, one image can be represented by a set of features reducing the amount of data to represent it. An alternative approach for image dimensionality reduction is using sub-space based techniques. The aim of sub-space methods is compute compact representations using approaches as direct cosine transform (DCT), principal component analysis (PCA), Fourier transforms, or Haar integrals. Once the lower-dimensional image representation is computed, important robot tasks as: motion estimation and topological localization can be performed.

In this work an omnidirectional camera is one of the main sensors used. Therefore, it is important to review the state-of-art of the feature extraction in this kind of images. Omnidirectional vision has received special attention recently due to its long-term landmark tracking, its wide field of view, its robustness to occlusions, its ability to be fused with range data, and its reduced noise sensitivity [Gaspar et al., 2007]. However, the mirror shape introduces important radial distortions into the image plane; as a result the computer vision community has proposed novel image descriptors considering this situation. Table 2.4 shows two classifications: firstly, feature extraction techniques based on the SIFT detector applied to omnidirectional images, although it is not designed to work with them. Particularly, [Scaramuzza et al., 2009] proposes a descriptor for tracking vertical edges onto omnidirectional images which allows performing appearance-based mapping and localization; and [Lourenco et al., 2010] proposes a modified SIFT detector which considers the radial distortion in omnidirectional images, but preserving the invariance to scale and rotation.

Secondly, a novel set of feature descriptors designed to cope with the challenging distortions presented onto the omnidirectional images. To start with, [Goedemé et al., 2007] and [Tapus et al., 2004] proposes features extracted from vertical edges, which are not deformed by the mirror shape. A particularly different approach is presented in [Briggs et al., 2006], where the omnidirectional image is converted in a panoramic image, and then a 1D circular image is formed by averaging the scan-lines of the panoramic image. Basically, it is a data reduction technique to represent panoramic images and used for robot localization. The Polar High-Order Local Auto-correlation descriptor proposed in [Linaker & Ishikawa, 2006] presents a set of masks which consider the radial distortion since they are polar-based and rotation invariance. Last, [Hansen et al., 2010] and [Guerrero, 2011] propose key-point detectors which work on the sphere model representation and using spherical Gaussian techniques for feature extraction.

2.5. PLATFORM DESCRIPTION AND DATASETS

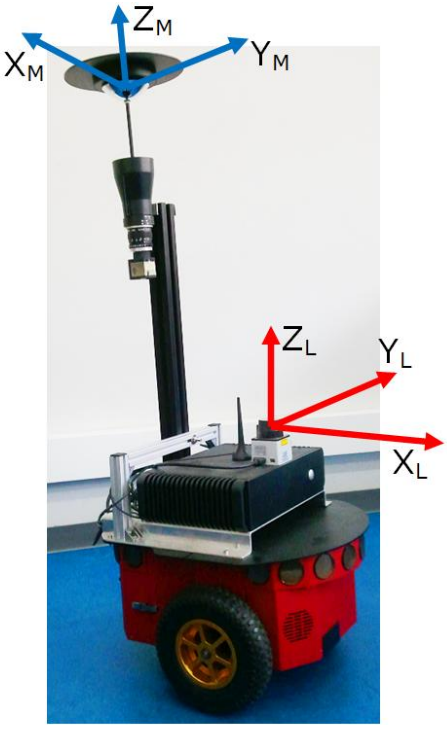
For the work reported here, the experimental platform used is composed by the Pioneer 3DX mobile robot and a perception system consisted of a URG-04LX LRF and an omnidirectional camera. Both are briefly explained in Section 2.5.1. Furthermore, Section 2.5.2 describes the collected datasets used to test part of the results reported in this work are explained in detail.

2.5.1. *The Mobile Robot and the Perception System*

Figure 2.5a presents the Pioneer 3DX mobile robot and the coordinate frames of the LRF and the omnidirectional camera. The relevant characteristics of the mobile robot platform, the omnidirectional camera and the LRF are mentioned below.

- *Pioneer 3DX mobile robot*.
    - o **Computer** – On board PC computer at 1.5Ghz running Linux Ubuntu 10.0

    - **Sensors** – 8 ultrasound sensors (4 rear and 4 front) and motors with 1/500 tick encoders.
    - **Mechanics** – Two drive wheels in differential drive arrangement and one rear caster wheel.
    - **Communications** – Wireless Ethernet 802.11
    - **Power supply** – 3 batteries of 12V providing an autonomy of 3hours approximately.
    - **Dimensions** – Width 38cm, height 21.5cm, depth 45cm and weight 9kgr.
    - **Other technical data** – Payload 17Kgr, maximum velocity 5.76 km/h.
- *Omnidirectional vision system.*
    - **Optics** – *RemoteReality*.
    - **Mirror diameter** – 74mm.
    - **Camera** - UI-2230SE-C camera with a resolution of 1024x768 pixels. Data interface USB.
    - **FOV** – Vertical 90º (15º up and 75º down), horizontal 360º.
- *Laser Range Finder (LRF)*
    - **Range** – 0.02m to 4m.
    - **Operation** – Indoors.
    - **Scan angle** – 240º.
    - **Scan time** – 100ms.
    - **Resolution** – 1mm (0.02m to 1m) and 1% between 1m and 4m.
    - **Interface** – Data RS232, power supply using USB ports.



a)

b)

Figure 2.5 Robot platform and perception system. a) Pioneer 3DX robot and coordinate frames of the LRF and the omnidirectional camera. b) Detail of the omnidirectional camera. c) Hokuyo URG-04LX LRF.

2.5.2. Datasets

The datasets available online contains one run of different places which is not useful for long-term mapping and localization. Since, it requires data recorded under various illumination conditions, over several days and presence of dynamic obstacles. Recently, the COLD database [Pronobis & Caputo, 2009] includes data recorded in the conditions described above. However, the sensor data of the COLD database was designed to be used separately, this means that sensor fusion cannot directly be used. For instance, the range data cannot be projected onto the omnidirectional image (or standard camera) since the extrinsic calibration information is missing. Therefore, a dataset was collected at the PIV building of the University of Girona which has three floors. These data sets include the LRF readings, the corresponding omnidirectional images and the robot odometry. The dataset was collected at different time of the day and between seasons. The relevant characteristics of the dataset are described below.

- **Directory structure** – The directory structure of the collected dataset is divided by year seasons, each season is divided in times of day, each time of day holds separate directories for each floor, and finally each floor contains a different sequence of data ordered by date. This directory structure is presented below:
    - *UDG-Montilivi-P4-VICOROB*
        - Autumn
            - 1-Morning
                - P0
                    - DS-2010-10-28
                        - imgData
                        - dataLaser.mat
                        - dataOdometry.mat
                        - dataOmni.mat
                        - dataSonar.mat
                        - dataTXT.zip
                    - DS-2010-11-03
                - P0+P1+P2
                - P1
                - P2
            - 2-Afternoon
            - 3-Night

    - Spring
    - Summer
    - Winter
- **Filenames and data format** – The data sequence is presented in MAT files (*dataLaser.mat, dataOdometry.mat, dataOmni.mat, dataSonar.mat*), but the original data sequence is stored in the '*dataTXT.zip*' file. In addition, the '*imgData*' stores the omnidirectional image sequence. The internal formats of each data file is presented below:
    - *Odometry data format* – Odometry data is stored in a matrix of Nx6, where N is the number of sequences. Each row is formatted as follows:

      timestamp Xr Yr ThetaR V W

      where, *timestamp* is the number of milliseconds elapsed from the acquisition started; *Xr, Yr* and *ThetaR* are the current robot position variables; *V* and *W* are the robot linear and angular velocity.
    - *Laser scan data format* – Laser scans are stored in a matrix of Nx1367, where N is the number of sequences. Each row is formatted as follows:

      timestamp scan-size range-value theta-value …

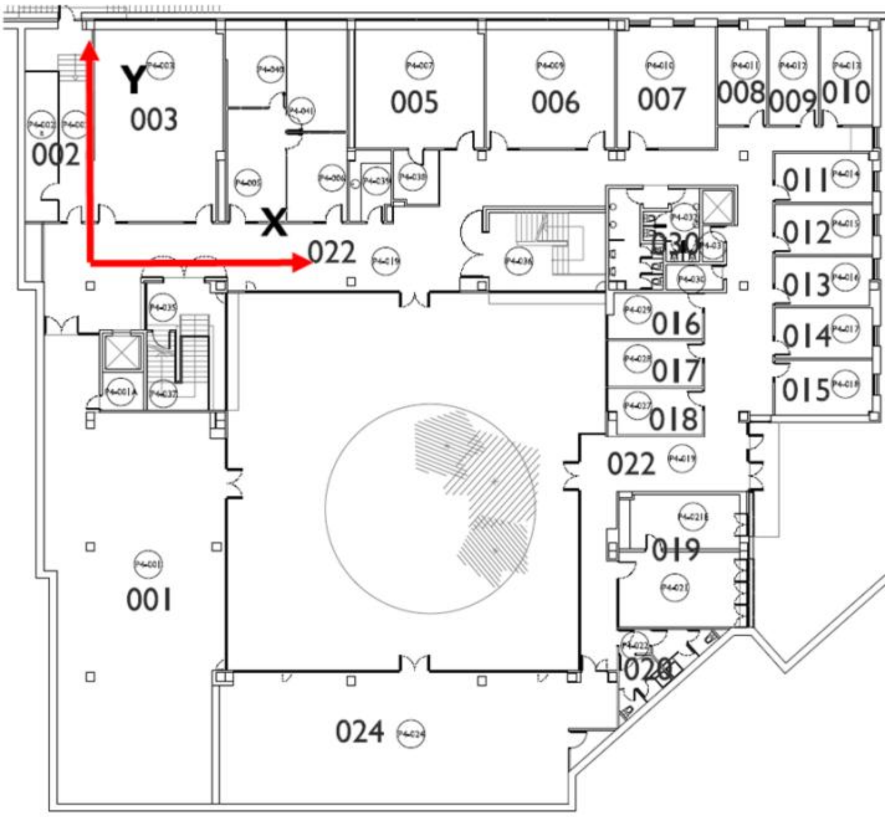
      where, *scan-size* is the number of range/theta values in the current row; *range-value* is the range value measured at the orientation given by *theta-value*. *Range-value* is measured in *m* and *theta-value* is measured in degrees.
    - *Sonar data format* – Sonar data is stored in a matrix of Nx33. Each row is formatted as follows:

      timestamp range-value theta-value …

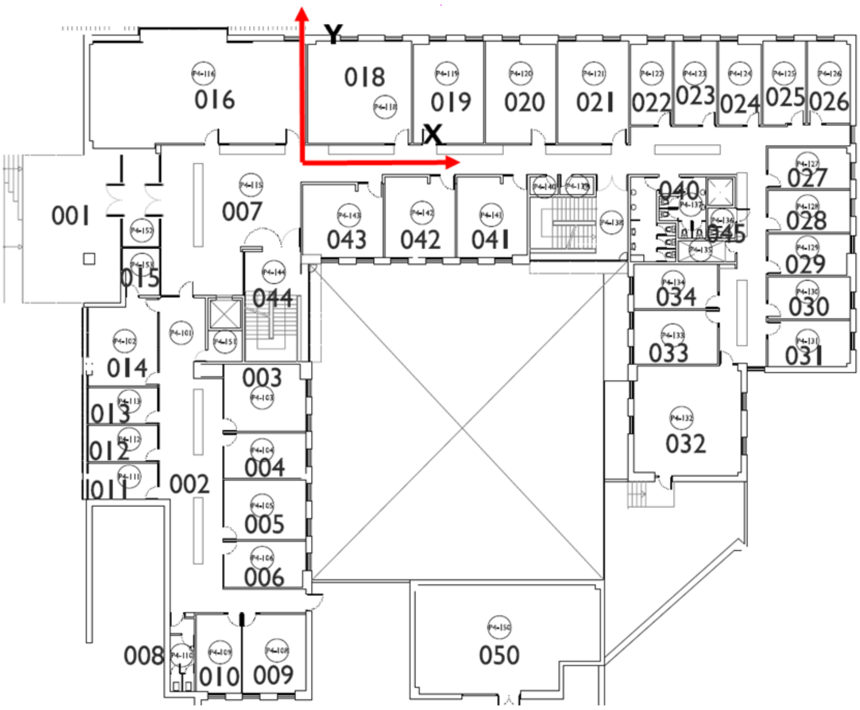
      where, *range-value* is the range value measured at the orientation given by *theta-value*. There are sixteen pairs of range/theta values corresponding to the sixteen sonar sensors installed in the Pioneer 3DX.
    - *Omnidirectional image data format* – Omnidirectional image name data is stored in a matrix of Nx2. Each row is formatted as follows:

      timestamp image-filename

      where, *image-filename* is the image filename which has the following template: *imgData/img_XXX.png*, *XXX* means the image consecutive number. All the omnidirectional images are stored in the relative sub-directory '*imgData*'.
- *Coordinate systems* – The coordinate systems assumed at each floor of the PIV building are presented in Figure 2.6.


Y
003
002
005
006
007
008
009
010
X
022
030
011
012
013
014
015
016
017
018
022
019
020
001
024

a)


Y
018
019
020
021
022
023
024
025
026
016
X
001
007
043
042
041
040
045
027
028
029
030
031
015
044
034
033
014
003
032
013
012
011
002
004
005
006
008
010
009
050

b)

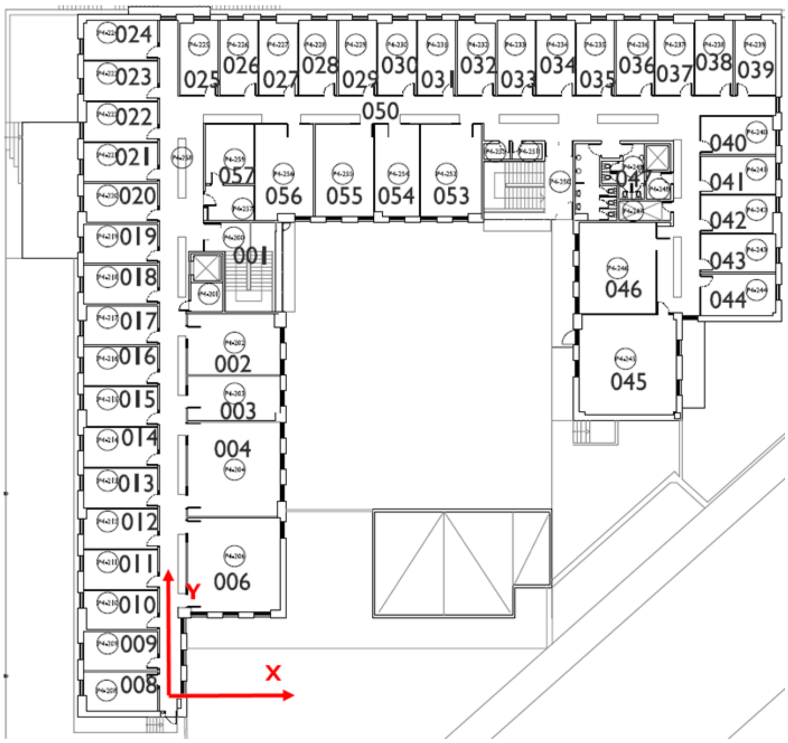


Figure 2.6 Coordinate systems of the three floors of the PIV building at the University of Girona. a) P0 level. b) P1 level. c) P2 level.

2.6. **DISCUSSION**

In the second section of this chapter the SLAM problem and the most common solutions were introduced. Afterwards, special attention was focused on lifelong mapping and localization motivated by the long-term operation involved in service robots. Furthermore, a brief review of the environment modeling techniques relying on LRF and optical images was presented.

The review of lifelong mapping and localization methods presented in Section 2.3 shown different approaches to deal with dynamic environments. Some of them relying on filtering out dynamic objects, others focused on pruning techniques, a couple of them integrate the dynamic changes of the environment into the SLAM formulation and alternative approaches are inspired on biological/psychological concepts. Since, lifelong mapping and localization is a relatively new problem in the robotics community, there is not a common option to solve it. However, it seems worthwhile designing and implementing lifelong SLAM techniques able to use the high number of SLAM solutions.

In this context, this work proposes a mapping and localization method whose main contribution is the Feature Stability Histogram (FSH). This is inspired on the human memory model [Atkinson & Shiffrin, 1968] to deal with changing environments and long-term mapping and localization. The main idea behind this is building a histogram using a weighted voting scheme. Each histogram bin is related with a map feature and it is incremented if the feature is re-observed. The weights associated to each feature are computed according to their strength, which can be the feature uncertainty the Hessian value in the SURF descriptor, or the matching distance in an epipolar geometry framework. The Atkinson and Shiffrin memory model considers basically two main components: the Short-Term Memory (STM) which retains information long enough to use it; and the Long-Term Memory, which retains information for longer periods of time or lifetime. Depending on the feature strength, it can be part of the STM or LTM. In this work, distinguishing between STM from LTM features (non-stable from stable features) is performed through *k-means* clustering or exponential decay identifying the mean lifetime. Once the LTM features are identified, they are only used for mapping and localization.

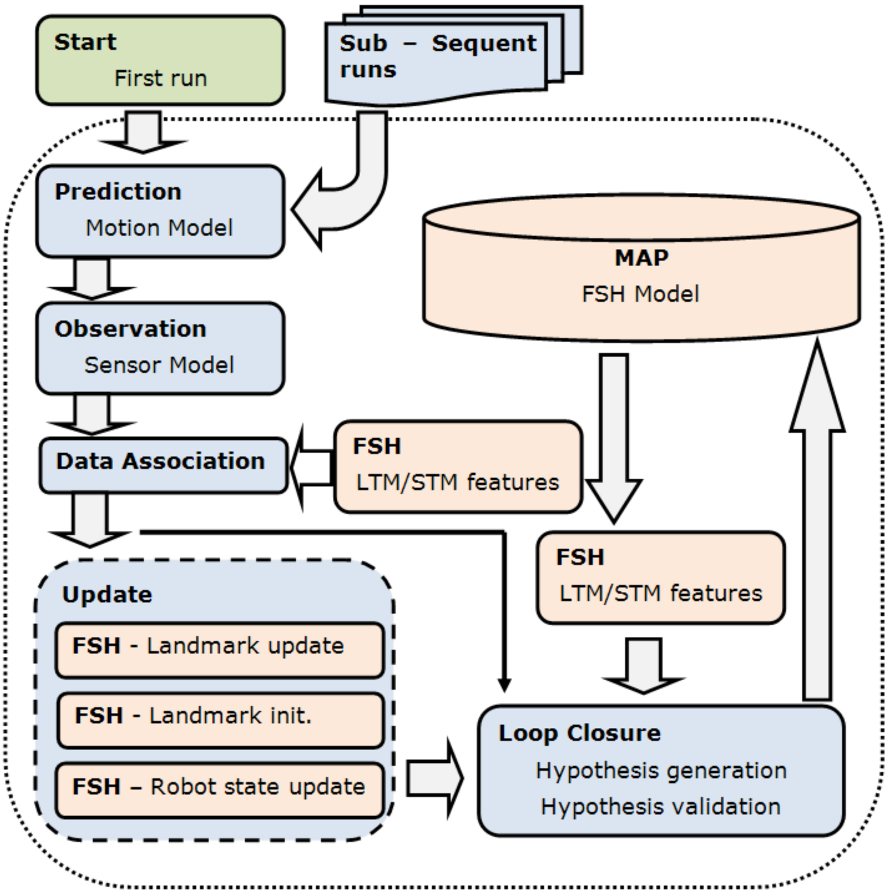


Figure 2.7 System overview emphasizing on the typical estimation loop and on affected parts when the FSH model is used.

The FSH model can be used as part of any SLAM solution; this can be observed in Figure 2.7. In this figure the typical steps of the robot pose and map estimation processes are depicted: prediction, observation, data association, update and loop closure detection. The proposed FSH model affects the data association, update and loop closure processes, as a result the map representation are particularly influenced. Once the motion and sensor model are applied, the LTM and STM features are extracted from the current map representation to obtain the re-observed landmarks after the data association process. The re-observed and new landmarks are used by the FSH approach to update the LTM and STM features, in this way the map is updated in changing environments reflecting its alterations in the LTM or STM. Furthermore, the robot pose is updated considering only the LTM features. Regarding loop closure detection, the loop closure hypothesis validation is implemented using the LTM features. Subsequent runs uses the LTM scans and LTM features to correct the predicted pose obtained from the robot motion model.

Therefore, using a memory management model to improve the mapping and localization capabilities of current SLAM solutions in long-term operation is the aim of the approach presented in this work.

CHAPTER 3

3. *FEATURE EXTRACTION AND ENVIRONMENT MODELING*

CONTENTS

The aim of the robot sensors is to model the environment, these sensors normally yield large amount of raw data which are hard to process in real time. For this reason and compactness feature-based environment models are preferred than raw data-based models. Feature-based environment models are normally used in SLAM solutions due the limited of information used and consequently the less computational time required for process it. However, the SLAM solution works properly if the feature extraction and matching process work accordingly, since the feature re-observation is crucial to correct the map information and the robot pose. Therefore, in this chapter feature extraction approaches for two different kinds of sensors are presented. Afterwards, combining them a range-augmented omnidirectional sensor is presented to extract vertical edge features from indoors scenes. The performance is analyzed through experimental validation.

3.1. INTRODUCTION

One of the most important problems in the context of map building and robot localization is to find correspondences between the observations taken at different places in the environment. These correspondences are crucial to simultaneously estimate the robot position and the map of the environment. For this reason, the environmental feature representativeness and the matching process reliability are important factors in solving the SLAM problem.

In Section 2.1 several examples were presented where depending on the robot environment and the application needs, different measurement models are used. The feature representativeness depends on the robot environment, and then the robot sensors have to take advantage of the environment structure to extract the more suitable features. For instance, office environments are full of planes (walls), lines (horizontal and vertical), doors, squares (paintings, posters), or circles (garbage cans); house environments have doors, paintings, stairs, furniture, different kind of windows, and ornaments; outdoor environments have trees, lines (vertical and horizontal), roads, polygons, buildings, civil structures, etc. The rationale behind this is given the robot sensors and the robot environment, the feature extraction process has to be focused to take out the environment landmarks with high representativeness.

Feature representativeness by itself does not ensure robustness [Lowe, 2004]. Feature robustness is normally evaluated in the context of its invariant characteristics, for instance rotation and translation invariance, color invariance, or illumination invariance. These characteristics ensure the feature can be observed again when the vehicle is revisiting a previously mapped area, which is the key to improving the vehicle localization and the map of the environment.

In this work, the LRF and the omnidirectional camera are the two main sensors used. They are combined to extract vertical features from indoor real-world images. The vertical edge features are predominant in indoor structured environments, and they are not deformed by the non-linear distortions introduced by the omnidirectional camera mirror. Section 3.2 describes the LRF features extraction process. Section 3.3 is focused on the vertical edge detection process. Combining the LRF and omnidirectional vision data is used to extrinsically calibrate these sensors, which is presented in Section 3.4. Finally, Section 3.5 explains the sensor model used, the image descriptor extracted and their matching process.

3.2. LASER RANGE FINDER FEATURES

LRF devices have been used over many years in the robotics community to obtain range/bearing measurements. The most common used LRF is a 2D scanner and suitable for planar motions. However, attaching a rotational mechanism to a 2D LRF or using more than one LRF with different orientation are common solutions to obtain 3D range/bearing measures. In this work, the LRF used is the Hokuyo URG-04LX, which has maximum range of 4m, a measurement area of 240º and an accuracy of ±1cm to ±4cm. The LRF features will be combined with vertical edges extracted from omnidirectional images, to do so breakpoint and horizontal lines detection are needed. The latter allows the sensor model to obtain an accurate estimation of the range/bearing of the vertical edge.

3.2.1. Laser Range Finder Calibration

This kind of sensors base its operation on the measurement of time-of-flight of a laser beam, which is emitted by a transmitter, reflected by a rotating mirror and then received again due its reflection on an object surface. The angular resolution of the scanner and communication speed to a host can be selected by users, minimum and maximum range values are predefined by manufacturer, but systematic errors as laser tilt/pan angle, surface reflectance properties, surface angle, and imperfect calibration, between others can affect the range measurement.

Literature offer few works measuring these properties, however [Reina & Gonzales, 1997] and [Ye & Borenstein, 2002] show a good characterization of a custom made equipment (Explorer scan laser made by Schwartz Electro-Optics Inc.), and the popular SICK LMS200 respectively. These works explore characteristics as: laser alignment, drift effect, surface reflectance properties, incident beam angle, and laser model. In our case, we focus on laser alignment, drift effect and laser model, since these properties affect directly the laser measure quality. Nowadays SICK LMS200 laser range finder sensors still are expensive, but HOKUYO made a set of variety of relatively low cost laser range finders, then in this work a URG-04LX laser range finder will be used, and the results achieved are accordingly to [Okubo, 2009].

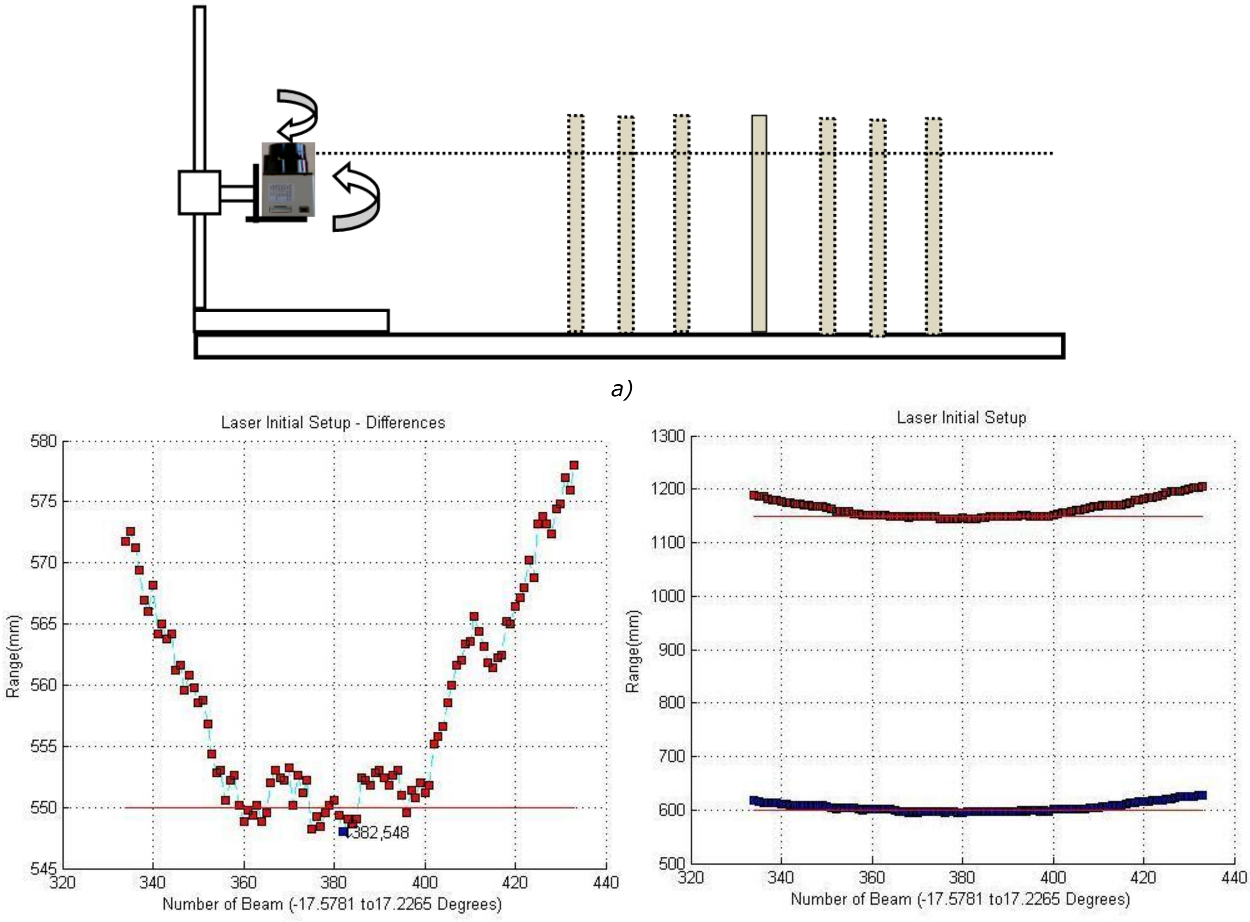
3.2.1.1. *Laser alignment*

Figure 3.1a shows the experiment setup for the laser range finder calibration. Laser alignment is an important issue, since if the LRF is no properly aligned it will be an incident angle different to a right angle between the laser beam and the calibration target, this could cause a range measure variation, and if the LRF central beam does not show the closer distance to the calibration target whole 2D laser points scanned will be rotated.

The URG-04LX has an angle range of 240º and an angle resolution of 0.36º, a total of 683 angle steps, and accordingly the manufacturer the central beam is placed at the 384 angle step. In order to achieve so, the laser alignment procedure performed is based on [Ye & Borenstein, 2002], but with a slightly difference:

- Calibration target was placed at distance D1 from the laser range finder, then using a C/Matlab software for laser continuous data acquisition the laser tilt/pan angle was modified. At this laser tilt/pan angle a total of 50 measures were taken, and its average was calculated.
- Calibration target was placed at distance D2 from the laser range finder, and then the laser tilt/pan angle was modified using the software for laser continuous data acquisition. At this laser tilt/pan angle a total of 50 measures were taken, and its average was calculated.
- Taking the difference of D2-D1 two things must be observed: first the minimum value must to be the real D2-D1 value which is known; second the laser beam at which this minimum is reached must be 384 or as close as possible to it. This manual procedure is repeated until these two conditions been satisfied.

In this work, D1 = 600mm, D2 = 1150mm, then D2-D1 = 550mm. It was found that the 382th beam produced a minimum of 548mm. Then, for further proposes the 382th laser beam was selected to work with. Figure 3.1b and 3.1c show an example of a typical data acquisition for laser alignment.



a)

b)

c)

Figure 3.1 a) Experimental setup for the laser range finder calibration. Arrows show how the laser tilt/pan angle can be modified. b) Laser alignment, it shows the difference between two distances. c) Laser alignment, It shows the average result of 50 measures for each distance.

3.2.1.2. *Drift effect*

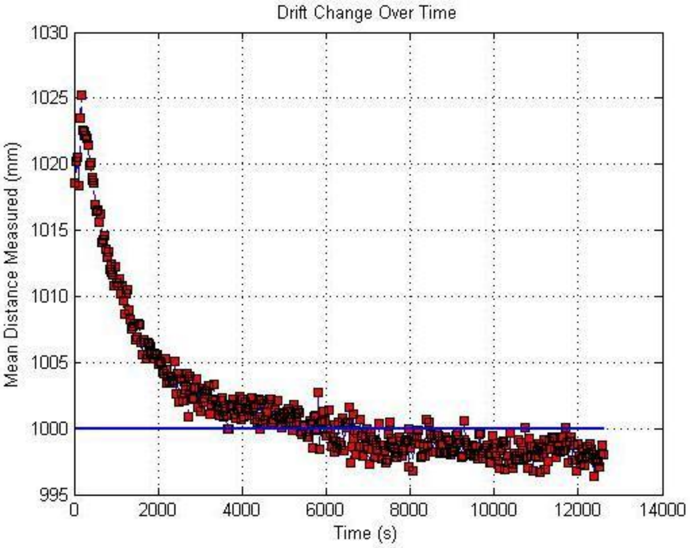


Figure 3.2 Laser range finder drift effect over time.

Range fluctuations could be present at laser range finder power-up, in practical situations these fluctuations are not keep in mind, but in the case of camera/laser calibration they are a source of error that must to be avoided. In order to reduce the laser range finder drift effect the following process was implemented [Ye & Borenstein, 2002]: the calibration target was placed at 1000mm away from the laser range finder, then at time T_i 50 samples were taken and its average distance calculated, and so on for $T_k > T_i$ until a total of time of 3 hours in permanent operation was reached.

Figure 3.2 shows after 90 minutes of continuous operation the range fluctuations decrease up reasonable levels less than 2.5mm. Therefore, before use this laser range finder for camera/laser calibration tests, it was turned on and used for 90minutes.

3.2.1.3. *LRF linear model*

In order to find the laser range finder model, experimental data were used and acquired as follows: the calibration targets were placed at 19 different positions, at each position 50 measures were taken, and its average was calculated. It was found a linear trend between the real distance and the mean at each position, which agrees with [Reina & Gonzales, 1997], [Ye & Borenstein, 2002] and [Okubo, 2009]. A linear laser range finder can be assumed, as Equation 3.1 shows.

$$\hat{y} = m\mu + b \qquad 3.1$$

Where, $\hat{y}$ is the estimate true distance, μ is the mean of measured range at each position, and m and b are the model parameters, which can be estimated using regular linear regression methods, then its values are given by Equation 3.2 and 3.3.

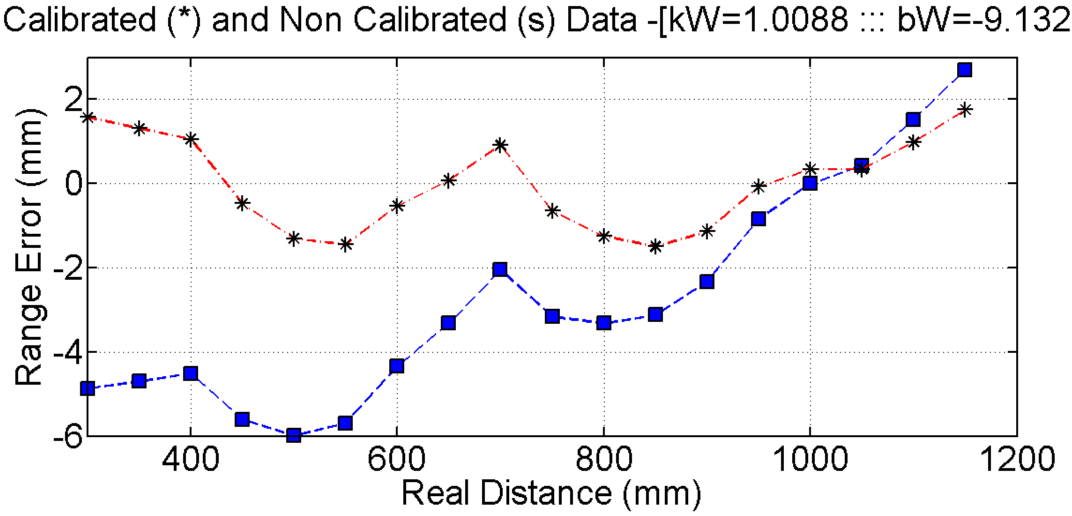


Figure 3.3 Range error Vs. real distance of calibrated (top) and non-calibrated (bottom) laser readings.

$$m = \frac{\sum_{i=1}^{n}(y_i - \bar{y})(\mu_i - \bar{\mu})}{\sum_{i=1}^{n}(\mu_i - \bar{\mu})} \qquad 3.2$$

$$b = \bar{y} - m\bar{\mu} \qquad 3.3$$

Where, $\bar{y}$ and $\bar{\mu}$ are the mean values of y_i and μ_i respectively. After run 10 different tests and consolidate its results, it was found that: $m = 1.0088$ and $b = -9.1322$. Then, for further laser range finder measurements we apply the Equation 3.4 as the laser model.

$$d = 1.0088l - 9.1322 \qquad 3.4$$

Where, l is the range given by the laser range finder, and d is the estimated range. Figure 3.3 shows the range error, or $\bar{y} - y$ the difference between the estimated distance and real distance, vs. the real distance. In this figure can be appreciated a considerable reduction of the range error.

3.2.2. *Breakpoint Detection*

The LRF returns a set of range/bearing raw data points, which can be processed to obtain features as corners or lines. This section is focused on the corner extraction problem formulated from the point of view of breakpoint detection, which is the detection of homogeneous regions represented by a state space model. On the other hand, other detection techniques based on the split-and-merge algorithm are described in [Einsele, 2001] and [Pfister et al., 2003]. They use a recursive line fitting method followed by line intersection point estimation. The problem with those methods is the line model, since they have to deal with infinite slopes. To mitigate this problem the line model has to be changed resulting in more complex estimation process.

The stochastic breakpoint detector described earlier has been formulated in [Castellanos & D., 1999] and it is briefly described here. The state space model is described by Equation 3.5.

$$\begin{aligned} \boldsymbol{x_{k+1}} &= \boldsymbol{F_k x_k} + \boldsymbol{w_k} \\ \boldsymbol{z_k} &= \boldsymbol{H_k x_k} + \boldsymbol{v_k} \end{aligned} \tag{3.5}$$

Where, $\boldsymbol{x_k} = [r_k \quad \partial r_k/\partial t]$ is the state vector composed by the range measurement and its rate of change; $\boldsymbol{w_k}$ and $\boldsymbol{v_k}$ are uncorrelated Gaussian noise with zero mean and covariance $\boldsymbol{Q_k}$ and $\boldsymbol{R_k}$ respectively; $\boldsymbol{R_k}$ is related with the sensor range error and matrices $\boldsymbol{F_k}$ and $\boldsymbol{H_k}$ are depicted in Equation 3.6.

$$\begin{aligned} \boldsymbol{F_k} &= \begin{bmatrix} 1 & \partial t \\ 0 & 1 \end{bmatrix} \\ \boldsymbol{H_k} &= [1 \quad 0] \end{aligned} \tag{3.6}$$

Where, ∂t is 1 since the time passed between *k* and *k+1* is the same for all instants. The stochastic model proposed in Equation 3.5 assumes constant acceleration motion, for which the $\boldsymbol{Q_k}$ matrix can be described as Equation 3.7 shows.

$$\boldsymbol{Q_k} = \begin{bmatrix} \frac{\partial t^4}{4} & \frac{\partial t^3}{2} \\ \frac{\partial t^3}{2} & \partial t^2 \end{bmatrix} \sigma_q^2 \tag{3.7}$$

Where, σ_q^2 in this work has the same value of the sensor range error. The formulation of [Castellanos & D., 1999] considers a Kalman filter estimation process; the Algorithm 3.1 implements the breakpoint detection using the Kalman filter framework.

Algorithm 3.1 Breakpoint detection algorithm.

```
k = 2; // Initializes the current LRF index
ki = 2; // Initializes the last breakpoint detected index.
**Rb** = [0 0 … 0]; // Initializes a Boolean array indicating if the range point is a breakpoint or not.
**While** k <= Np // Iterates over all the range points (total of points: Np)
        **If** ki == k
                $\mathbf{x_k} = \mathbf{x_0}$; // Filter reset
                $\mathbf{P_k} = \mathbf{P_0}$;
        **else**
                // Prediction
                $\mathbf{x_{k|k-1}} = \mathbf{F_k} * \mathbf{x_{k-1}}$;
```

```
            $\mathbf{P}_{k|k-1} = \mathbf{F}_k * \mathbf{P}_{k-1} * \mathbf{F}_k + \mathbf{Q}_k$;
            // Measurement
            $V_k = r_k - \mathbf{H}_k * \mathbf{x}_{k-1}$; // Innovation
            $S_k = \mathbf{H}_k * \mathbf{P}_{k|k-1} * \mathbf{H}_k^T + R_k$;
            // Breakpoint detection
            If $(V_k^2 / S_k) \geq \chi_1^2$
                // Breakpoint detected
                Rb(k) = 1;
                Rb(k-1) = 1;
                ki = k;
                k = k - 1;
            Else
                // Update the state model
                $\mathbf{K}_k = \mathbf{P}_{k|k-1} * \mathbf{H}_k^T * (1 / S_k)$;
                $\mathbf{X}_k = \mathbf{x}_{k|k-1} + \mathbf{K}_k * V_k$;
                $\mathbf{P}_k = \mathbf{P}_{k|k-1} - \mathbf{K}_k * \mathbf{H}_k * P_{k|k-1}$;
            end
        end

        k = k + 1;
    end
```

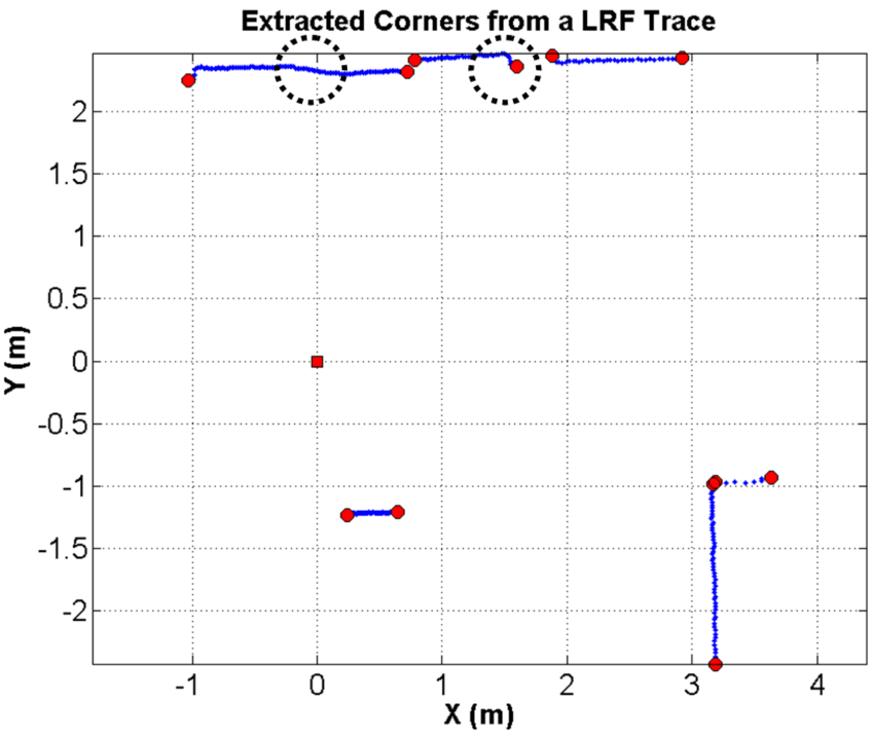


Figure 3.4 Corner extraction results.

The breakpoint detection process described above is applied each time a new observation is performed. Figure 3.4 shows the resulting breakpoints detected applying Algorithm 3.1, where the red circles are the corners detected, the dot data points represent the LRF trace and the square shaped dot is the robot position. The dotted circles in Figure 3.4 show two regions where the algorithm used has mistaken. However, in this work the detected corners are used to initializing the split-and-merge algorithm for the laser line estimation, which is proposed in the next section.

3.2.3. *Laser Lines Detection*

Line extraction algorithms from 2D laser scans has been proposed over many years. They include the widely known split-and-merge [Pavlidis & Horowitz, 1974], line-tracking approaches [Forsyth, 2003], Hough transform based methods [Pfister et al., 2003], line regression algorithms [Arras & Siegwart, 1997,], and RANSAC [Fischler & Bolles, 1981] based methods [Forsyth, 2003]. An interesting comparison of these algorithms is done in [Nguyen et al., 2005]. According to [Nguyen et al., 2005], the split-and-merge and line-tracking based approaches showed best performances taking into account speed and correctness. However, the split-and-merge methods strongly depend on the cost function and the line model assumed. In this work, the split-and-merge approach was implemented using a director vector line model and a cost function based on the geometrical constraint between the normal vector and the points of the line to be fit.

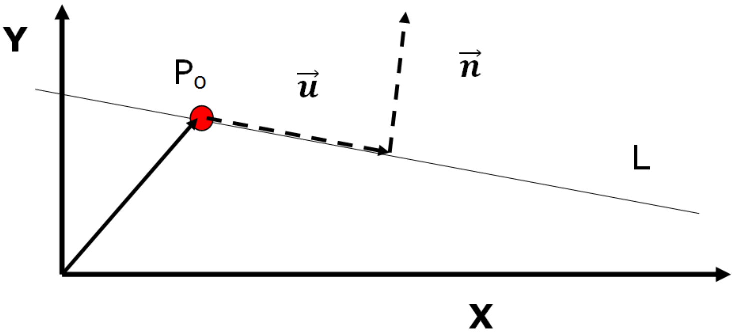


Figure 3.5 Director vector line model and the line normal vector.

Figure 3.5 shows the director vector based line model, where the line *L* can be parameterized using: $\boldsymbol{P_0}$ a point which belongs to *L* and $\vec{u}$ a unitary vector representing the direction of *L*. The laser segmentation approach followed in this work is based on two fundamental steps: firstly, transforming the laser trace to a common coordinate frame; and secondly, recursive split-and-merge taking into account a cost function which measures the data variation along the line normal.

The inputs to the laser line segmentation algorithm are the breakpoints detected and the LRF points. LRF points between two consecutive breakpoints are analyzed for line segmentation. Figure 3.6a shows the case for the first and the second breakpoint of Figure 3.4. The *{L}* local coordinate frame is defined with the X axis oriented to the next breakpoint. The base coordinate frame *{B}* is the current robot position. A transformation between *{L}* and *{B}* is defined as depicted in Equation 3.8.

$$\boldsymbol{T} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix} \left(\boldsymbol{P_L} - \begin{bmatrix} P_{bx} \\ P_{py} \end{bmatrix} \right) \qquad 3.8$$

Where, θ is the X axis orientation of the frame *{L}* with respect to the *{B}* frame, $\boldsymbol{P_L}$ is a *2xN* matrix with the LRF points (*N* is the total of LRF points considered), and $[P_{bx}\ P_{by}]^T$ are the coordinates of the first breakpoint.

Once the LRF points are transformed to the *{B}* frame, a line model is built in parametric form considering that described in Figure 3.5 and showed in Equation

3.9. Using the director vector of the line defined between the two breakpoints, a normal vector is built as denoted in Equation 3.10.

$$\boldsymbol{L_{LRF}} = \boldsymbol{P_b} + \boldsymbol{u} * t \qquad 3.9$$

$$\boldsymbol{u_N} = [-u_y \quad u_x]^T \qquad \mathit{3.10}$$

Where, $\boldsymbol{P_b}$ is the first breakpoint, *u* is the director vector and *t* is the parameter. Taking advantage of the vector definition of the laser line, the cost function to evaluate if the LRF points match the line model is defined by the dot product of the transformed LRF points and the normal vector. This is depicted in Equation 3.11.

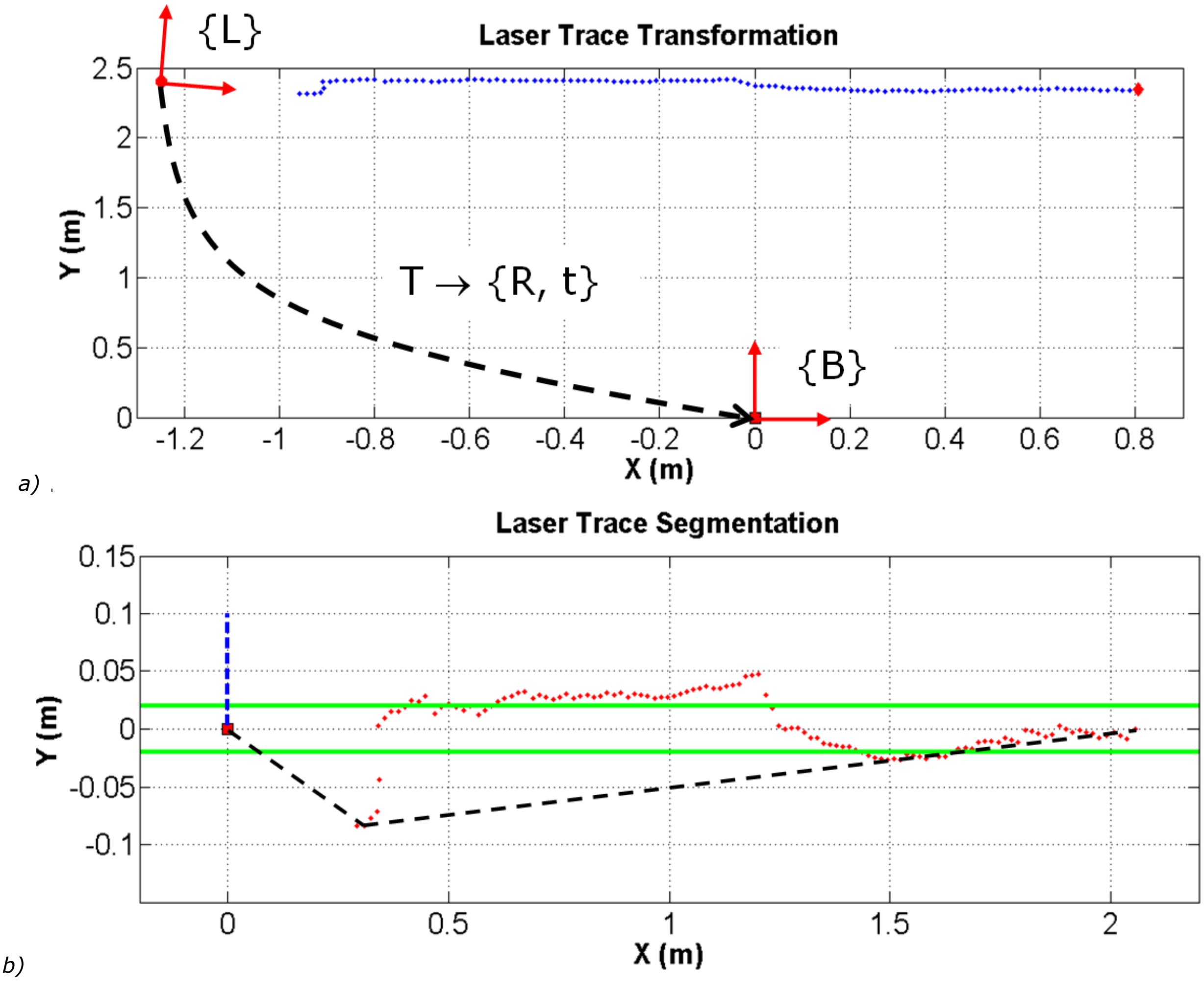


Figure 3.6 a) Laser trace transformation. b) Laser trace segmentation.

$$d(\boldsymbol{P_L^B}) = max_i(\boldsymbol{P_{L,i}^B} \cdot \boldsymbol{u_N}) \qquad 3.11$$

Where, $\boldsymbol{P_{L,i}^B}$ is the LRF point in the *{B}* frame and *d* is the maximum value of the dot product. If *d* remains below of a threshold value, the LRF points are accepted as belonging to a line, otherwise the laser trace is decomposed in two chains of laser points and the process is repeated. Figure 3.6b shows the LRF trace in dotted line; the normal vector placed in the origin and depicted in non-continuous line, the threshold values in continuous lines, and the non-continuous lines shows the laser points decomposition. The Algorithm 3.2 sums up the line segmentation process described above.

Algorithm 3.2 Laser line segmentation algorithm.

1. .	**LRFsegments** = []; // Where the point line, director vector and length is stored

```
foreach pairBP ∈ LRFbreakpoints // LRFcorners stores the breakpoints detected
        xyLRF = getPointsBetweenCorners(pairBP);
        doSegmentEstimation(xyLRF);
end

function doSegmentEstimation(xyPts)
        [Po u L] = getLinePointDirectorVectorAndLength(xyPts); // Eq. 3.9
        LineTheta = getLineOrientation(u);
        xy = getTransformedPoints(Po, LineTheta);                  // Eq. 3.8
        uN = getNormalVector(u);                                   // Eq. 3.10
        [costValue costValueID] = doEvaluateLine(xy, uN);          // Eq. 3.11
        If costValue < THRESHOLD                                   // Evaluate model or Split.
                LRFsegments ← [Po u L];
        else
                [xyPts1 xyPts2] = doSplit(xyPts, costValueID);
                doSegmentEstimation(xyPts1);
                doSegmentEstimation(xyPts2);
        end
end
```

The laser line estimation process described in the Algorithm 3.2 is applied each time the robot obtains a new observation. Figure 3.7 shows the lines extracted considering the breakpoints detected in the previous Section. Observing the original laser trace depicted in Figure 3.6b, the URG-04LX LRF used in this work has a considerably amount of noise which is significantly reduced using the laser line estimation proposed in this Section.

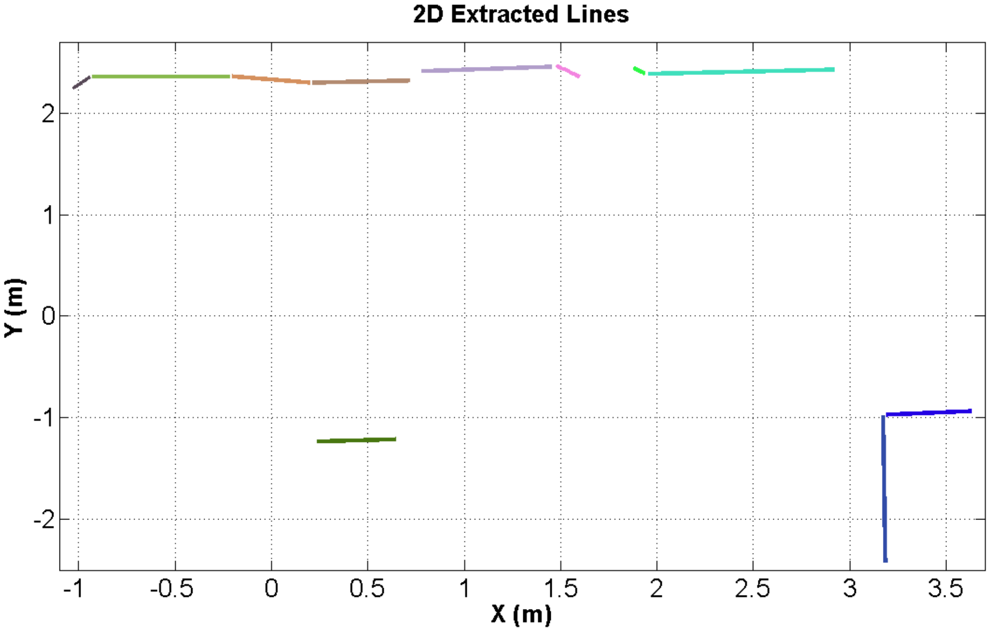


Figure 3.7 Extracted laser lines applying Algorithm 3.2.

3.3. OMNIDIRECTIONAL VISION FEATURES

The Table 2.4 from Section 2.4.2 presents a review of the feature extraction methods for omnidirectional images. Despite of the classification done in Table 2.4, a closer view of the features extracted to model the environment allows to notice that many techniques are supported in methods inherited from those used in standard vision devices as [Hansen et al., 2007], [Lourenco et al., 2010], [Hansen et al., 2010] and [Arican & Frossard, 2010]. On the other hand, other techniques use appearance-based information as the vertical edges of the environment for feature extraction [Scaramuzza et al., 2009] and [Goedemé et al., 2007]; using 1D panoramas as in [Briggs et al., 2006]; combining vertical edge positions in the panoramic image with color information as in [Tapus et al., 2004]; and, tracking planes which are detected using a LRF and an omnidirectional camera as in [Mei, 2007]. Appearance-based signatures have advantages as providing rich environmental information (color, shape, texture), however the features descriptors have high dimensions (e.g. 128 or 64) compared with geometric landmark descriptions. From SLAM point of view, appearance-based features are very useful estimating associations between places, which are valuable for loop closing or re-localization situations.

Intuitively, the appearance-based model of the environment describes the environment as it is, taking advantage of its natural structure. In this work, the omnidirectional camera is used to extract vertical edges, which afterwards will be combined with the LRF data projected onto the omnidirectional image to estimate the edge position. For this aim, this section presents the vertical edge detection algorithm implemented and the results obtained.

3.3.1. Central Catadioptric Edge Detection Algorithms

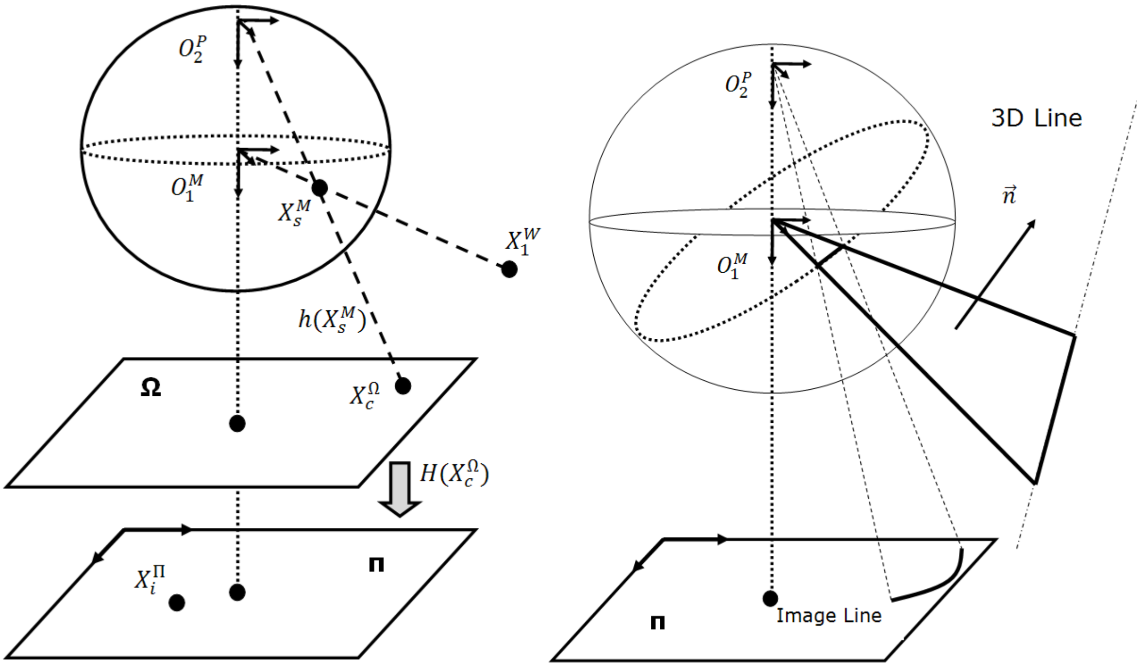
A central catadioptric camera consists of a perspective or orthographic camera, and a mirror. The latter have to be conic, hyperbolic, or parabolic in order to obtain a single view point vision system. Projective models for these cameras have been developed by [Geyer & Daniilidis, 2000], [Barreto & Araujo, 2005] and [Mei & Rives, 2007]. Catadioptric vision systems have received special attention recently due to its long-term landmark tracking, its wide field of view, its robustness to occlusions, its ability to be fused with range data, and its reduced noise sensitivity [Gaspar et al., 2007]. However, many useful classical image processing algorithms are no longer valid on omnidirectional images, because the image deformations induced by the mirror. Edge extraction is one of those algorithms, since environmental edges are projected as conics on the omnidirectional image instead of lines as in the perspective case.

Catadioptric edge detection methods can be divided in two basic categories. Firstly, methods based on conic fitting techniques as [Zhang, 1997], and more recently [J. Bermúdez, 2010]. These methods have the advantage to deal with calibrated and un-calibrated catadioptric cameras. Secondly, methods based on classical image processing techniques as the Hough transform or classical edge operators (e.g. Laplacian, Sobel masks, Canny, etc.). This is the case of [Tapus et al., 2004] and [Scaramuzza et al., 2009] which use classical edge detectors as the first step of processing, afterwards custom approaches are applied as image unwrapping and radial histograms to detect the most reliable vertical edges; in [Mei & Malis, 2006] the randomized Hough transform [Xu & Oja, 1993] is used to extract vertical edges. Related with this category, in [Bazin et al., 2007] an alternative approach is proposed. It consists in building chains of connected pixels which correspond to catadioptric edges. The proposed solution is analytic; as a result it does not depends on sampling the edge parameter space as in Hough based techniques.

The edge detection algorithm used in this work is a modified version of that presented in [Bazin et al., 2007]. This technique has the following advantages: first, it does not depend on parametric constraints as those presented in the Hough based approaches; second, it uses the standard projection model for catadioptric images proposed by [Geyer & Daniilidis, 2000] and it proposes a closed solution to define which pixels belongs or not to an edge; last, it uses well know image processing techniques, which aids in its implementation. The next two sub-sections explain the implemented algorithm in detail and show the results achieved using real-world images.

3.3.2. *Vertical Edge Detection*

The main topics of the approach proposed in [Bazin et al., 2007] will be briefly described here, as well as the modifications done in order to optimize the implementation for real-time operation. To start with, it is important to briefly describe image formation model for catadioptric cameras using the unified projection model of [Geyer & Daniilidis, 2000] and [Mei & Rives, 2007]. The basic steps are depicted in Figure 3.8a and described as follows:



a) b)

Figure 3.8 a) Image formation model in catadioptric cameras. b) Projection of a 3D line using unified model.

1. A world point given by $\boldsymbol{X_1^W} = [x_1 \quad y_1 \quad z_1]^T$ is projected onto the unitary sphere centered at $\boldsymbol{O_1^M}$; as a result, a point onto the unitary sphere can be defined as Equation 3.12 shows.

$$\mathbf{X_s^M} = \frac{\mathbf{X_1^W}}{\|\mathbf{X_1^W}\|} = \begin{bmatrix} \frac{x_1}{\|\mathbf{X_1^W}\|} & \frac{y_1}{\|\mathbf{X_1^W}\|} & \frac{z_1}{\|\mathbf{X_1^W}\|} \end{bmatrix}^T \qquad 3.12$$

2. Using the calibration parameter ζ the new projection point is changed to $\boldsymbol{O_2^P}$; doing this, the sphere point of Equation 3.12 changes to $\boldsymbol{X_s^P} = [X_{s,x}^M \quad X_{s,y}^M \quad X_{s,z}^M + \zeta]^T$.

3. The sphere points $\boldsymbol{X_s^M}$ are projected to the plane at infinity Ω using the function $\boldsymbol{h}(\boldsymbol{X_s^M})$ defined in Equation 3.13. This function is bijective, since points residing in the plane Ω can be lifted onto the unitary sphere. Equation 3.14 shows the inverse $h(\boldsymbol{X_s^M})$ function.

$$\boldsymbol{X_c^\Omega} = \boldsymbol{h}(\boldsymbol{X_s^M}) = \begin{bmatrix} \frac{X_{s,x}^M}{X_{s,z}^M+\xi} & \frac{X_{s,y}^M}{X_{s,z}^M+\xi} & 1 \end{bmatrix}^T \quad 3.13$$

$$\boldsymbol{X_s^M} = \boldsymbol{h}^{-1}(\boldsymbol{X_c^\Omega}) = \begin{bmatrix} X_{c,x}^\Omega \\ X_{c,y}^\Omega \\ 1 - \xi \frac{r+1}{\xi+\sqrt{1+(1-\xi^2)r}} \end{bmatrix}, \text{where } r = {X_{c,x}^\Omega}^2 + {X_{c,y}^\Omega}^2 \quad 3.14$$

4. Last, using the intrinsic camera parameters the $\boldsymbol{X_c^\Omega}$ are projected onto the image plane, which is depicted in Equation 3.15.

$$\boldsymbol{X_o^\Pi} = \boldsymbol{K}\boldsymbol{X_c^\Omega} = \begin{bmatrix} \gamma_1 & \gamma_1 s & u_o \\ 0 & \gamma_2 & v_o \\ 0 & 0 & 1 \end{bmatrix} \boldsymbol{X_c^\Omega} \quad 3.15$$

Where, γ_1 and γ_2 are the generalized focal lengths [Mei & Rives, 2007], *s* is the skew and *[u_o, v_o]* is the principal point.

In this work, the omnidirectional camera parameters such as ζ, the principal point, the generalized focal lengths, the skew and the distortion parameters were obtained using the C. Mei calibration toolbox [Mei, 2006]. Table 3.1 shows their values and uncertainties.

Table 3.1Omnidirectional camera calibration parameters.

Name	**Symbol**	**Value and uncertainty**
Mirror parameter	ζ	0.96651 ±0.00599
Principal point	u_0, v_0	510.61716 ±1.43172, 420.61415 ±1.68635
Generalized focal lengths	γ_1, γ_2	402.97909 ±0.67177, 403.55496 ±0.67395
Skew	*s*	0
Distortion parameters	K_1, k_2, k_3, k_4, k_5	-0.01934 ±0.00623, 0.00229 ±0.00362, -0.00016 ±0.00183, -0.00017 ±0.00046, 0

Using the projection model described above, a 3D line can be projected onto the image plane. To do so, the 3D line and the center of the unitary sphere $\boldsymbol{O_1^M}$ define a plane. This plane intersects the unitary sphere defining a great circle onto it. The great circle points are projected from $\boldsymbol{O_2^P}$describing a cone, which intersects the image plane describing a conic line. Having described the projection model process, and particularly the projection of 3D lines onto catadioptric images, the vertical edge detector is described below.

The Algorithm 3.3 outlines the vertical edge detector implemented in this work. The edge detection algorithm receives three control parameters: the minimum edge length (*minEdgeLength*), the minimum angle between edges to split (merge) them (*minAng2Split*) and the maximum error to define what pixels belong to an edge (*maxError2Split*). The main steps of the algorithm are as follows: Firstly, the algorithm works with edge images, which can be extracted using various methods, in this work the Canny detector was used. In [Bazin et al., 2007] the authors used the Matlab implementation of the Canny detector, but to improve the computing time the edge detection was implemented using OpenCV [Bradski, 2000] and Matlab™ MEX files (line 4). Secondly, using the binary image and the minimum edge length chains of connected pixels are extracted (line 5). The computational time of this chaining process is very high to be used in real-time. For this reason, in this work the chaining process was implemented entirely in C++ and accessible through Matlab using MEX files.

Thirdly, the list of edges is evaluated to find which edges are catadioptric line projections (lines 6-12). The criterion to consider a point part of a catadioptric line is based on geometrical constraints. The plane defined by the center of the unitary sphere $\boldsymbol{O_1^M}$ and the edge projection onto this sphere can be parameterized by its normal vector $\vec{\boldsymbol{n}}$, which is shown Figure 3.8b. Any point which belongs to this plane has to satisfy the geometric relationship give by Equation 3.16, which is called the great circle constraint.

$$\boldsymbol{X_{s,i}^{M}}^{\boldsymbol{T}} \boldsymbol{n_s} = 0 \tag{3.16}$$

Where, $\boldsymbol{n_s}$ is the plane normal vector; and $\boldsymbol{X_{s,i}^{M}}^{\boldsymbol{T}}$ is the *i-th* point on the sphere with respect to the mirror frame, which are obtained through a lifting process (e.g. take an image point and find its corresponding sphere point) using a look-up-table (LUT) improving the computational time for real-time operation (line 10). The normal vector $\boldsymbol{n_s}$ can be extracted using the extreme points of the current line on the sphere, which is the method done in [Bazin et al., 2007] (line 11). However, it is not accurate enough since it does not take into account the other edge points. In this work, the normal vector is computed using singular-value-decomposition (SVD) as depicted in Equation 3.17.

$$\begin{bmatrix} x_{s1} & y_{s1} & z_{s1} \\ \cdots & \cdots & \cdots \\ x_{sN} & y_{sN} & z_{sN} \end{bmatrix} \boldsymbol{n_S} = \boldsymbol{A} * \boldsymbol{n_S} = 0 \tag{3.17}$$

Where, $\boldsymbol{X_{s,i}^{M}}^{\boldsymbol{T}} = [x_{si} \quad y_{si} \quad z_{si}]$ is a point on the sphere, *N* is the total number of points belonging to a chain, and $\boldsymbol{n_s}$ is the normal vector. Using SVD, $\boldsymbol{A} = \boldsymbol{USV}^T$, ordering the eigenvalues of $\boldsymbol{S}$ in decreasing order, the third column of $\boldsymbol{V}$ contains the least-squared solution of Equation 3.17.

Last, the method presented so far detects any edge; hence, in this work the pipeline process was modified in order to detect only the vertical edges (lines 30-38). One additional constraint given by the cross-product between the sphere projection of the edge (parameterized by the normal vector $\boldsymbol{n_s}$) and the normal ground plane ($\boldsymbol{n_G}$) was included (line 31). The process to obtain the normal vector $\boldsymbol{n_G}$ is as follows: the reference frame of the laser points is changed to the camera frame using the extrinsic calibration described in Section 3.4; using Equation 3.17 applied to these points, the current normal vector n_G is found.

Algorithm 3.3 Vertical edge detection algorithm implemented.

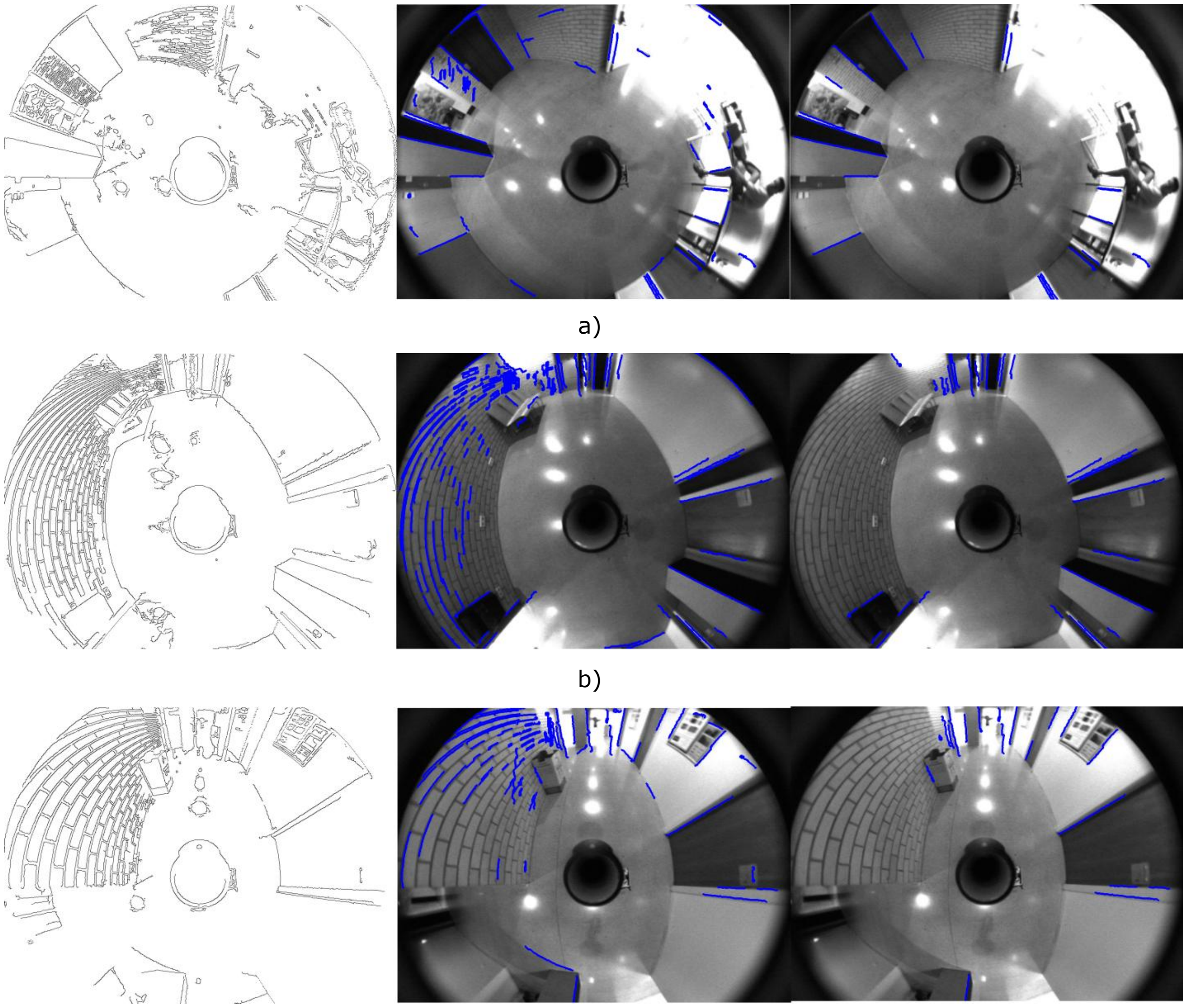
```
1.  // CATADIOPTRIC EDGE DETECTION PROCEDURE.
2.  function [eNormals, eSphEdges, eImgEdges] = getCatadioptricEdges(I, minEdgeLength,
3.  minAng2Split, maxError2Split)
4.  // Getting image edges and pixel chains.
5.  Ibin = getBinaryEdgeImage(I);
6.  allPixelChanins = getPixelChains(Ibin, minEdgeLength);
7.  // Edge detection on omnidirectional image through recursive split.
8.  foreach pixelChain ∈ allPixelChanins
9.          edgeImgPixels = getImgPixels(pixelChain);
10.         edgeSphPoints = getShperePoints(pixelChain);
11.         [eNormals, eSphEdges, eImgEdges] = doEdgeDetection_Split(edgeImgPixels,
            edgeSphPoints, minEdgeLength, minAng2Split, maxError2Split);
```

```
End
// Merging edges
[eNormals, eSphEdges, eImgEdges] = doEdgeDetection_Merge(eNormals, eSphEdges, eImgEdges, minAng2Split);
End

// MAIN PROCEDURE.
// Initialization parameters.
I ← getGrayScaleImage();
minEdgeLength ← 0.3*(OUTTER_RADI – INNER_RADI);
minAng2Split ← 1º:
maxError2Split ← 0.01;
// Getting all the normals, sphere points and image points of the edges detected.
[eNormals, eSphEdges, eImgEdges] = getCatadioptricEdges(I, minEdgeLength, minAng2Split, maxError2Split);
// Extracting vertical edges.
veNormals = [];
veSphEdges = [];
veImgEdges = [];
foreach edgeNormal ∊ eNormals
        costValue = doEvaluateVerticalEdge(edgeNormal);
        if costValue < VERTICAL_EDGE_THRESHOLD
                veNormals ← edgeNormal;
                veSphEdges ← getRespectiveShpereEdge(edgeNormal);
                veImgEdges ← getRespectiveImageEdge(edgeNormal);
        end
end
```

3.3.3. *Results*

The Algorithm 3.3 described in the section above was tested using the dataset described in Section 2.5. It is worth noting that the images used correspond to real-world images captured in a mobile robot on motion. The vertical edge detection algorithm has three basic parameters namely the minimum edge length (*minEdgeLength*), the minimum angle between edges to split (merge) them (*minAng2Split*), the maximum error to define what pixels belong to an edge (*maxError2Split*) and the vertical edge threshold. In this work, they were fixed as follows: the *minEdgeLength* is the third part of the effective omnidirectional area given by the difference between the outer (taken from the omnidirectional camera calibration files) and inner radii; the *minAng2Split* was set to 1º; the *maxError2Split* and the *VERTICAL_EDGE_THRESHOLD* were set to 0.01.



a)

b)

c)

Figure 3.9 Vertical edge detection onto omnidirectional images. Each row shows a different scenario of the dataset.

Figure 3.9 shows three different scenarios of the same dataset (level zero-building PIV-summer-morning). The leftmost image is the binary edge image obtained using the Canny edge detector, the central image shows all the pixel chains projected onto the omnidirectional image (blue lines), and the rightmost image shows all the vertical edges found (blue lines).

Taking into account the calibration parameters of the omnidirectional image, the *minEdgeLength* has a value of 112 pixels, which means that long and prominent chains of pixels are only extracted. As a result, observing the edge images not all the obvious edges are shown in the central images. Despite the fact that the omnidirectional camera mechanical support was carefully ensured, the robot motion causes a blur in the images. For this reason, and considering the pixel chaining algorithm analyzes the image using an 8-neighbour area, various edges are discarded since either they do not satisfy the Equation 3.16, or after successive splitting the resulting chain is not long enough.

The environment modeled so far includes: a set of corner points and line segment extracted from the LRF data, and a set of vertical edges detected onto the omnidirectional image. The close relationship between the perceptual system used and the effectiveness of the SLAM solution was previously mentioned. In this work, the LRF and the omnidirectional camera are not taken as separate sensors generating separate feature sets; instead they are combined to obtain a more reliable feature set. To do so, the extrinsic calibration between these sensors is needed, which is described in the next section.

3.4. RANGE-AUGMENTED OMNIDIRECTIONAL VISION SENSOR

The aim of this section is to explain how to embed range information in omnidirectional images. To do this the extrinsic calibration of a 2D LRF and a catadioptric camera is computed. A state of art about the extrinsic calibration of these sensors is shown in Table 3.2. Works such as [Mei & Rives, 2006] report interesting methods for visible laser traces, but this approach clearly cannot be applied to a 2D LRF with an invisible trace as mostly used in robotics (SICK or HOKUYO). Other works like [Sooyeong Yi, 2007] and [Orghidan et al., 2003] build a minimization function based on the re-projection error, or obtain a closed form solution based on conics, since they use visible laser traces but their results show they have a limited range of operation. 2D LRFs are commonly used to build 3D range scans by adding a pan/tilt mechanism. This is the case with [Scaramuzza et al., 2007], where a 3D range scan and a catadioptric camera are used in 3D indoor/outdoor environmental reconstruction adopting a camera/laser calibration based on [Weingarten, 2006], which does not reveal enough details of the calibration process. In terms of perspective cameras, [Zhang & Pless, 2004] has a general non-linear approach unlike [Wasielewski & Strauss, 1995]. The method proposed in [Zhang & Pless, 2004] can be extended to omnidirectional cameras, and the methods of this paper are based on that approach. A Matlab toolbox is included in [Unnikrishnan & Hebert, 2005], but does not give enough details of its model.

Table 3.2 State of art summary for LRF and camera calibration

Ref.	Sensors	Observation
[Wasielewski & Strauss, 1995]	Perspective camera + 2D LRF	Calibration using linear models.
[Zhang & Pless, 2004]	Perspective camera + 2D LRF	Adaptable to central catadioptric cameras
[Antone & Friedman, 2007]	Perspective camera + 2D LRF	Calibration based on specific target
[Mei & Rives, 2006]	Omnidirectional camera + 2D LRF (visible, invisible)	Several methods for visible laser, but not clear fo invisible laser
[Unnikrishnan & Hebert, 2005]	Perspective camera + 3D LRF	Fast calibration using a Matlab toolbox
[Scaramuzza et al., 2007]	Omnidirectional camera + 3D LRF	One shot calibration
[Sooyeong Yi, 2007] [Orghidan et al., 2003]	Omnidirectional camera + 2D LRF (visible)	Closed solution based on conics.
[Weingarten, 2006]	Perspective camera + 3D LRF	Not enough details about calibration

The methods described in Table 3.2 use perspective or catadioptric cameras and a 2D laser with approaches applicable to visible laser traces. The method presented in this document embeds range information in omnidirectional images using the extrinsic calibration of a 2D LRF with an invisible laser trace and an omnidirectional camera using checkerboard patterns, and laser data points on these patterns. Unlike [Zhang & Pless, 2004] or [Mei & Rives, 2006], two calibration methods and three initial guess options are proposed. The calibration methods proposed are evaluated at pixel error level using ground truth data from the calibration patterns projected onto the omnidirectional image. In this work, the omnidirectional camera was calibrated with the C. Mei toolbox [Mei, 2006], and the 2D LRF with invisible trace was calibrated following the procedure specified in Section 3.2.1.

3.4.1. *Problem Formulation*

The LRF readings are distance measurements, belonging to a 2D plane parallel to the floor. The coordinate system used has the *Z* axis points upwards, the *Y* axis points forward, and is centered on the laser projection point. This is the same axis configuration as for the catadioptric camera, but centered on the mirror's central projection point.

Figure 3.10 shows the experimental setup. There are three coordinate systems, those of the laser, the calibration pattern and the camera. The problem focuses on finding ***R*** and ***T*** so that laser points $\boldsymbol{P_L}$ can be represented in the camera coordinate system, and then projected onto the omnidirectional image. The authors in [Zhang & Pless, 2004] base their model finding a rotation matrix ***R*** and a translation vector ***T*** in a simultaneous manner for perspective cameras, the approach described here finds ***R*** and ***T*** through a simultaneous and non-simultaneous way, and three different methods to get the initial guess for the non-linear minimization were explored. A point $\boldsymbol{P_C}$ in the camera coordinate frame can be described by Equation 3.18.

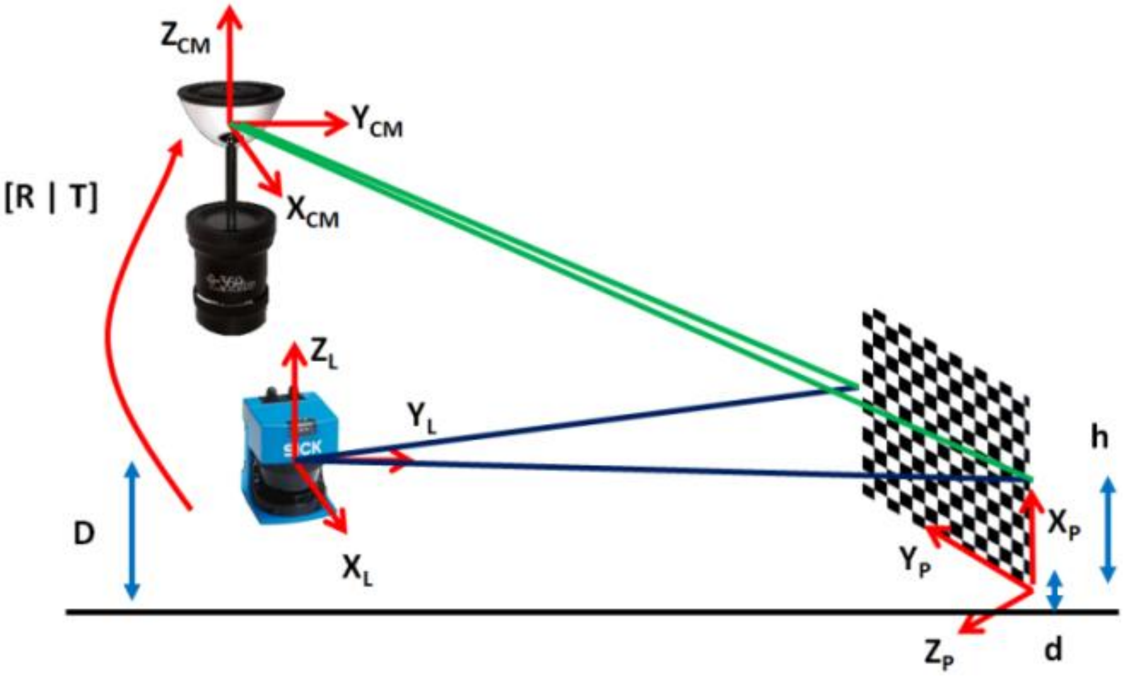


Figure 3.10 Problem description and experimental setup.

$$\boldsymbol{P_C} = \boldsymbol{R}\boldsymbol{P_L} + \boldsymbol{T} \qquad 3.18$$

Where, $\boldsymbol{P_L}$ is a point in the laser coordinate frame, ***R*** is a 3x3 rotation matrix, and ***T*** is a 3D translation vector of the camera center of projection with respect to the laser frame. The aim is to find ***R*** and ***T*** using the checkerboard planes, which are visible from the laser and camera point of view. This work shows two methods to find ***R*** and ***T***. The first, which tries to find them simultaneously performing two non-linear minimization processes, it has the advantage of arriving at a solution regardless of any alignment constraint between the LRF and the omnidirectional camera, but it involves 6 degrees of freedom (DOF) estimation. The second tries to find the translation vector ***T***, and then the rotation matrix ***R***. This decoupled parameter estimation has the advantage to reduce the problem estimation to 3DOF + 3DOF, which requires less training data in comparison with a simultaneous parameter estimation. In both cases a Levenberg-Marquardt (LM) optimization algorithm was used because it is easy to implement, and it generally has a fast convergence rate. However, this algorithm is sensitive to the initial guess [Madsen et al., 2004]. Then, three different ways to calculate this initial guess were implemented unlike [Zhang & Pless, 2004]. The initial guess methods are described as follows: first, one of the calibration poses available was selected and a set of linear least squares (LLS) equations were defined using these laser points. Second, two calibration poses were selected and the LLS equations were defined using these points. And third, a sub-set of the central laser points of each calibration pose was used to build the LLS equations.

3.4.2. *Simultaneous Parameter Estimation*

Data acquisition for this method includes images with checkerboard calibration planes and 2D laser points which belong to the calibration plane. Using these images camera calibration was performed and the extrinsic parameters were then used to define a vector parallel to the normal calibration plane as described by Equation 3.19 [Zhang & Pless, 2004].

$$\boldsymbol{N_C} = \boldsymbol{R_{3W}}\left(\boldsymbol{R_{3W}^T} \cdot \boldsymbol{T_W}\right) \qquad 3.19$$

Where, ***R*$_{3W}$** and ***T*$_{W}$** are the third column of the rotational matrix and the translation vector of the calibration plane pose with respect to the omnidirectional camera. Since, the laser points belong to the calibration plane there is a geometric constraint based on the distance between the camera and the calibration plane. This constraint can be expressed by Equation 3.20 [Zhang & Pless, 2004].

$$\boldsymbol{N_C} \cdot (\boldsymbol{R}\boldsymbol{P_L} + \boldsymbol{T}) = \|\boldsymbol{N_C}\|^2 \qquad 3.20$$

Where, ***R*** and ***T*** are the parameters to estimate, and $\boldsymbol{P_L}$ is a laser point. This expression differs from that proposed in [Zhang & Pless, 2004] since in this work the camera frame was selected as the base frame rather than the laser frame. Using Equation 3.20 a non-linear minimization function can be expressed as in Equations 3.21 and 3.22.

$$f(\boldsymbol{R}, \boldsymbol{T}, \boldsymbol{P_L}) = \boldsymbol{N_C} \cdot (\boldsymbol{R}\boldsymbol{P_L} + \boldsymbol{T}) - \|\boldsymbol{N_C}\|^2 \qquad 3.21$$

$$\nabla_{Q,T}\boldsymbol{f}(\boldsymbol{R}, \boldsymbol{T}, \boldsymbol{P_L}) = \boldsymbol{N_C} \cdot \left[\boldsymbol{\nabla R_{Q,P_L}} \boldsymbol{\nabla n_Q},\ \boldsymbol{I}_{3x3}\right] \qquad 3.22$$

Where, $\nabla_{Q,T}\boldsymbol{f}(\boldsymbol{R}, \boldsymbol{T}, \boldsymbol{P_L})\boldsymbol{P}_L$ is the gradient of the minimization function using quaternions instead rotational matrix, $\boldsymbol{\nabla R_{Q,P_L}}$ is the quaternion's gradient evaluated at point $\boldsymbol{P_L}$, and $\boldsymbol{\nabla n_Q}$ is the gradient of the quaternion's normalization factor. A LM algorithm was used to minimize 3.21, and we described three different ways to obtain the initial guess. By organizing the laser points in an ***AX***=***B*** system, and using Equation 3.21, the LLS equations for the initial guess are given by Equations 3.23, 3.24 and 3.25.

$$[N_{CX}P_{LX}, N_{CX}P_{LY}, N_{CX}P_{LZ}, N_{CY}P_{LX}, N_{CY}P_{LY}, N_{CY}P_{LZ}, N_{CZ}P_{LX} \quad N_{CZ}P_{LY}, N_{CZ}P_{LZ}, N_{CX}, N_{CY}, N_{CZ}] = \boldsymbol{A_i} \qquad 3.23$$

$$\left[r_{11}, r_{12}, r_{13}, r_{21}, r_{22}, r_{23}, r_{31}, r_{32}, r_{33}, t_x, t_y, t_z\right]^T = \boldsymbol{X} \qquad 3.24$$

$$\|\boldsymbol{N_C}\|^2 = B_i \qquad 3.25$$

Where, $\boldsymbol{A_i}$ is a row of the A matrix, B_i is a row of the B vector, X is the vector parameters to estimate, N_{CX}, N_{CY} and N_{CZ} are the $\boldsymbol{N_C}$ vector components of Equation 3.19, r_{11} to r_{33} are the 9 terms of the rotation matrix, t_x, t_y and t_z are the translation vector, and P_{LX}, P_{LY} and P_{LZ} are the *i-th* laser point components. The solution to these simultaneous equations gives a rank-2 rotation matrix for all three methods to obtain the initial guess. This happens when the training points are planar [Eggert et al., 1997] like the laser points, which are constrained to a 2D plane parallel to the floor. Due to this, the resulting matrix is not a proper rotation matrix, since it does not satisfy $\boldsymbol{RR}^T = \boldsymbol{I}$. In order to resolve this problem, [Eggert et al., 1997] propose finding the nearest rotation matrix which satisfies $\boldsymbol{RR}^T = \boldsymbol{I}$, by calculating the eigen-values and eigen-vectors of the resulting matrix, the nearest proper rotation matrix can be expressed by Equation 3.26.

$$\widehat{\boldsymbol{R}} = \boldsymbol{M}S^{+} \pm \frac{\boldsymbol{X}}{\sqrt{|Trace\,(\boldsymbol{X})|}} \qquad 3.26$$

$$S^{+} = \left(\frac{\boldsymbol{u_1 u_1^T}}{\sqrt{\delta_1}} + \frac{\boldsymbol{u_2 u_2^T}}{\sqrt{\delta_2}}\right) \qquad 3.27$$

$$X = [(\boldsymbol{M}S^{+})(\boldsymbol{M}S^{+})^T - \boldsymbol{I}]\boldsymbol{u_3 u_3}^{\boldsymbol{T}} \qquad 3.28$$

Where, u_1 and u_2 are the eigen-vectors corresponding to the non-zero eigen-values δ_1 and δ_2, and u_3 is the eigen-vector associated with the zero eigen-value. The sign in Equation 3.26 is chosen in line with the determinant of $\widehat{\boldsymbol{R}}$ being +1; and $\boldsymbol{M}$ is the LLS resulting matrix. After the LLS and the first non-linear minimization are solved, a second non-linear minimization is performed using and Euclidean constraint rather than a geometrical one. This is shown in Equation 3.30.

$$f(\boldsymbol{R},\boldsymbol{T},\boldsymbol{P_L}) = \sum_{i=1}^{K}\sum_{j=1}^{M}\frac{1}{2}\left(\frac{\boldsymbol{N_{Ci}}}{\|\boldsymbol{N_{Ci}}\|}(\boldsymbol{RP_{Lij}} + \boldsymbol{T}) - \|\boldsymbol{N_{Ci}}\|\right) \qquad 3.30$$

Where, K is the total number of calibration poses, M is the total number of laser points in the *i-th* checkerboard pattern, and $\boldsymbol{N_{Ci}}$ is the vector defined by Equation 3.19 for the *i-th* calibration plane. The minimization algorithm used was LM, but with an initial guess given by the first minimization stage.

3.4.3. *Non-simultaneous Parameter Estimation*

The second approach consists of performing decoupled parameter estimation. First, the translation vector is found, and then the rotation matrix is estimated using this information. This decoupled problem statement is common in mobile robotics [Unnikrishnan & Hebert, 2005] and [Mei, 2006]. In general, this method uses the same minimization and geometrical constraints as above. In the translation vector estimation Equations 3.19 to 3.21 are still valid. But, the Jacobian is now described by Equation 3.31. The initial guess for the LM algorithm can be re-written as shown in Equations 3.32 to 3.34.

$$\nabla_{Q,T}\boldsymbol{f}(\boldsymbol{R},\boldsymbol{T},\boldsymbol{P_L}) = \boldsymbol{N_C}.[\boldsymbol{I}_{3x3}] \qquad 3.31$$

$$[N_{CX}\ N_{CY}\ N_{CZ}] = \boldsymbol{A_i} \qquad 3.32$$

$$\left[t_x\ t_y\ t_z\right]^T = \boldsymbol{X} \qquad 3.33$$

$$\|\boldsymbol{N_C}\|^2 - (N_{CX}P_{LX} + N_{CY}P_{LY} + N_{CZ}P_{LZ}) = B_i \qquad 3.34$$

Where, $\boldsymbol{A_i}$ is a row of the A matrix, B_i is a row of the $\boldsymbol{B}$ vector, $\boldsymbol{X}$ is the parameters to estimate, N_{CX}, N_{CY} and N_{CZ} are the $\boldsymbol{N_C}$ vector components of Equation 3.19, t_x, t_y and t_z are the translation vector, and P_{LX}, P_{LY} and P_{LZ} are the *i-th* laser point components. The second parameter estimation calculates the rotation matrix using the translation vector found above. Equations 3.19 to 3.21 are still valid, but their Jacobian is described by Equation 3.35. The initial guess of the rotation matrix changes too, and the LLS equations can then be expressed as shown in Equations 3.36 to 3.38.

$$\nabla_{Q,T}\boldsymbol{f}(\boldsymbol{R},\boldsymbol{T},\boldsymbol{P_L}) = \boldsymbol{N_C}\cdot\left[\boldsymbol{\nabla R_{Q,P_L}\nabla n_Q}\right] \qquad 3.35$$

$$[N_{CX}P_{LX}, N_{CX}P_{LY}, N_{CX}P_{LZ}\ , N_{CY}P_{LX}, N_{CY}P_{LY}, N_{CY}P_{LZ}, N_{CZ}P_{LX}, \\ N_{CZ}P_{LY}, N_{CZ}P_{LZ}, N_{CX}, N_{CY}, N_{CZ}] = \boldsymbol{A_i} \qquad 3.36$$

$$[r_{11}\ , r_{12}, r_{13}, r_{21}, r_{22}, r_{23}\ , r_{31}, r_{32}, r_{33}]^T = \boldsymbol{X} \qquad 3.37$$

$$\|\boldsymbol{N_C}\|^2 - (N_{CX}T_X + N_{CY}T_Y + N_{CZ}T_Z) = B_i \qquad 3.38$$

Where, $\boldsymbol{A_i}$ is a row of the ***A*** matrix, B_i is a row of the ***B*** vector, ***X*** is the vector parameters to estimate, N_{CX}, N_{CY} and N_{CZ} are the $\boldsymbol{N_C}$ vector components of Equation 3.19, r_{11} to r_{33} are the 9 terms of the rotation matrix, P_{LX}, P_{LY} and P_{LZ} are the *i-th* laser point components, and t_x, t_y and t_z are the translation vector components of the first minimization stage. The resulting rotation matrix has the same rank-2 problem as described previously, so the nearest proper rotation matrix is found following the same procedure as shown above. In similar way as the simultaneous parameter estimation, a second minimization is performed using Equations 3.26 to 3.28.

3.4.4. Results

The calibration methods proposed were tested on a dataset composed by seventeen different images and LRF data points. Each image contains the calibration plane at different positions and orientations. The data acquisition process had 6 main steps: first, laser alignment was performed; second, there was a warm-up period of 90minutes; third, the checkerboard calibration pattern was placed; fourth, 15 laser readings were taken and corrected according to the laser model; fifth, an omnidirectional image was captured; sixth, steps 3 to 5 were repeated until acquisition was completed.

The omnidirectional camera was calibrated, and its intrinsic and extrinsic parameters were used to get the "ground truth" of the laser points on the calibration planes only. It is difficult to get a reliable ground truth data. However, taking advantage of the experimental setup (Figure 3.10), and defining: *h* = *D* – *d*, where *D* is the distance from the laser to the floor, *d* is the distance from the bottom right corner on the calibration pattern to the floor, and *h* is the relative distance from the laser trace to the calibration pattern origin. *D* and *d* were measured with a LLB-60D laser (±1.5mm). Then, using the intrinsic and extrinsic camera parameters a very close "ground truth" data can be obtained. In this way, unlike with [Zhang & Pless, 2004] and [Mei & Rives, 2006] the methods proposed are able to get the pixel error associated to the laser point's projection onto the omnidirectional image. Using the ground truth and the experimental laser points projected onto the omnidirectional image, the results for the three initial guess methods were tested.

Figure 3.11 graphs the results for the simultaneous parameter estimation method. Figure 3.11a shows the mean-squared-error (MSE) when each pose was taken to perform the extrinsic calibration, and tested over all other images. In the same way, Figure 3.11b shows the MSE for the central points of each calibration pose. On the other hand, Figure 3.11c shows the MSE when each pose is combined with other to obtain the extrinsic calibration, and tested over all other images. Observing Figure 3.11, the shape of the error curves is similar which suggest the calibration procedure is stable. The standard deviation at each point is plotted as a bar. In general, the two first guess methods offer less uncertainty compared with the third one. However, the total of successful calibration results for the first and second guess method was eight over seventeen calibration poses (giving a successful rate of 47%); nevertheless, the total of successful calibration results for the third guess method was 111 over 136 (total combinations of 2 calibration poses over 17) calibration poses (giving a successful rate of 82%). Therefore, despite the fact that the calibration results for the third guess method appear noisier considering their uncertainty bands, it is more likely to get a successful calibration result combining the LRF points of two calibration positions.

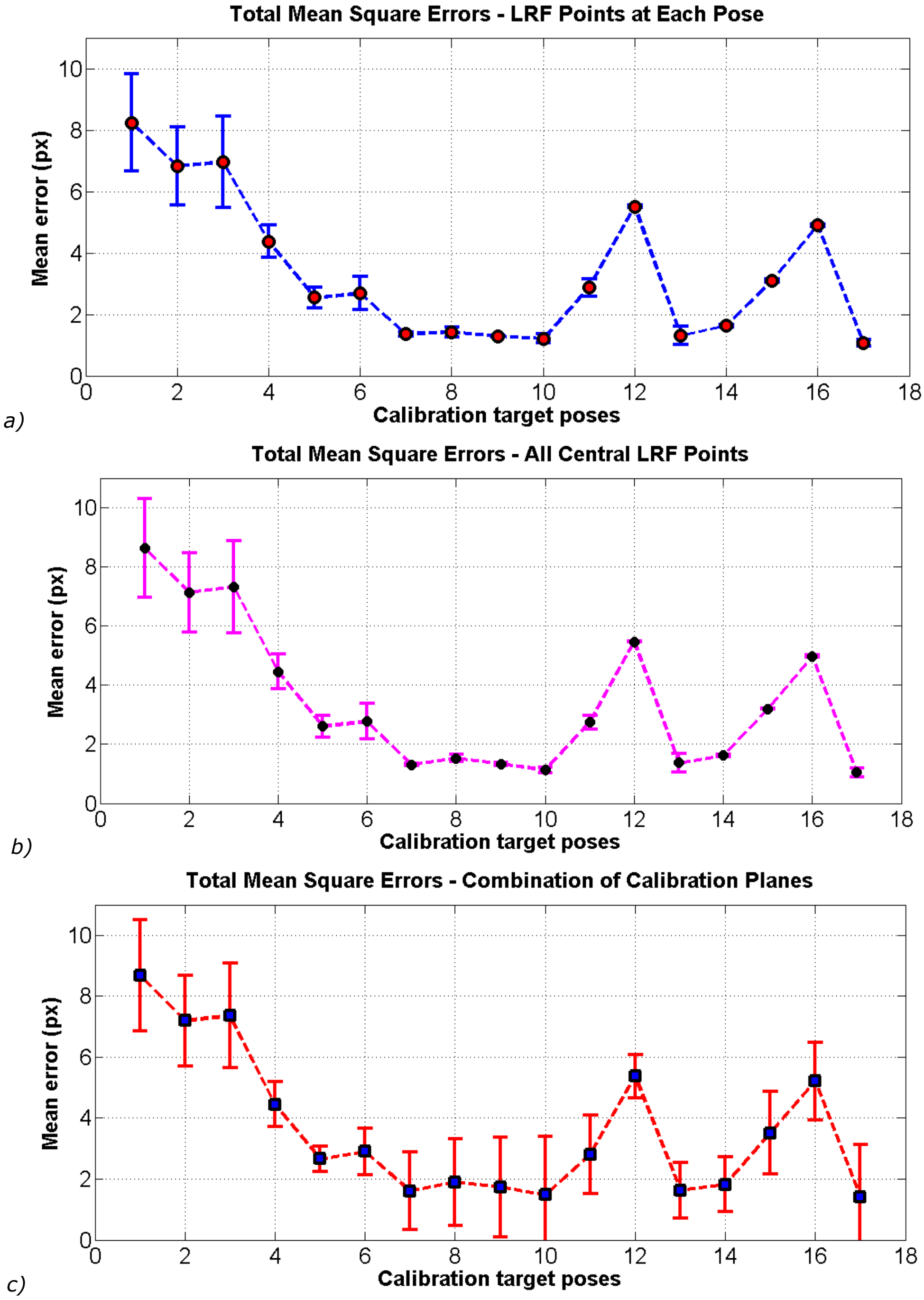
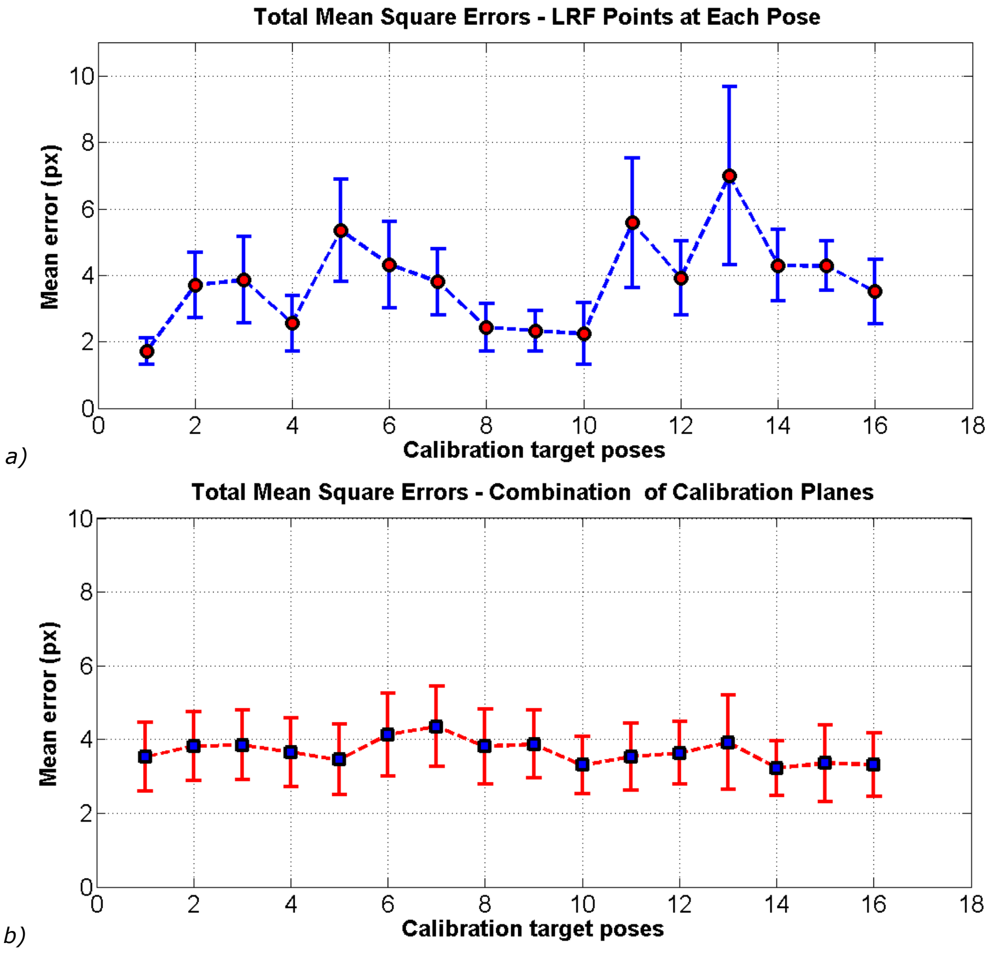


Figure 3.11 Results for simultaneous estimation with a total mean-squared-error less than 5px over all poses. a) Using a single calibration pose. b) Using the central points of all calibration poses. c) Using a combination of two calibration poses.

Figure 3.12 shows the results for the non-simultaneous parameter estimation method. It should be noted that the initial guess method which takes the central points of all the calibration planes is not present. In this case, even with a total MSE of less than 20px (extremely high and with no practical use) it was not possible to obtain any calibration result with this initial guess method. Observing the shape of the error graphs in Figure 3.12, it is worth noting less noisy results. However, the successful rate is less compared with the results of the simultaneous parameter estimation. In the case of the first guess method, it yields seven successful calibration results over seventeen poses (giving a rate of 41%); and in the case of the third guess method, it yields forty successful results over 136 (giving a rate of 29%).


Total Mean Square Errors - LRF Points at Each Pose

Total Mean Square Errors - Combination of Calibration Planes

Figure 3.12 Results for non-simultaneous estimation with a total mean-squared-error less than 5px over all poses. a) Using a single calibration pose. b) Using a combination of two calibration poses.

It can be observed that in case of the simultaneous parameter estimation is more likely to get a successful calibration result. In addition, it does not assume any alignment constraint between the LRF and the omnidirectional camera. Another important advantage of this method is considering the intrinsic coupling between rotation and translation. However, the simultaneous parameter estimation is nosier compared with the non-simultaneous parameter estimation due the difference in the degrees of freedom (DOF). The former performs a calibration process considering 6DOF, while the latter considers two successive calibrations of 3DOF. In the both methods proposed the calibration poses between the 6th and the 10th show low MSE. This may be due to the relative inclination of the checkerboard pattern with respect to the LRF frame, particularly in these calibration poses.

It is worth noting that the URG-04LX LRF is a low power device with a limited range, and noisy laser readings. It makes difficult to get a good enough initial guess due its geometrical constraint. For instance, there are missing laser readings on the black squares of the checkerboard pattern, with which the lesser LRF points the lesser accurate calibration results.

Using the best result found for the simultaneous parameter estimation method, the LRF points can be projected onto the omnidirectional image. Figure 3.13 shows two cases using real-world images of the dataset collected. In both images, the blue trace shows the LRF points which a range value less than the maximum measured by the URG-04LX laser scanner; the red trace shows those LRF points bigger than the maximum range. The latter is useful to observe the limited measurement capability of the URG-04LX.

In Section 3.2 the LRF data points were segmented, and their corners found. In Section 3.3 the vertical edges of the environment were detected from the omnidirectional image and its projection model. In this Section the extrinsic calibration of the LRF and the omnidirectional camera was computed. The next Section is focused on using the methods described and proposed up now to define the sensor model used in this work, which uses metric data as well as appearance-based information to extract reliable environmental features.

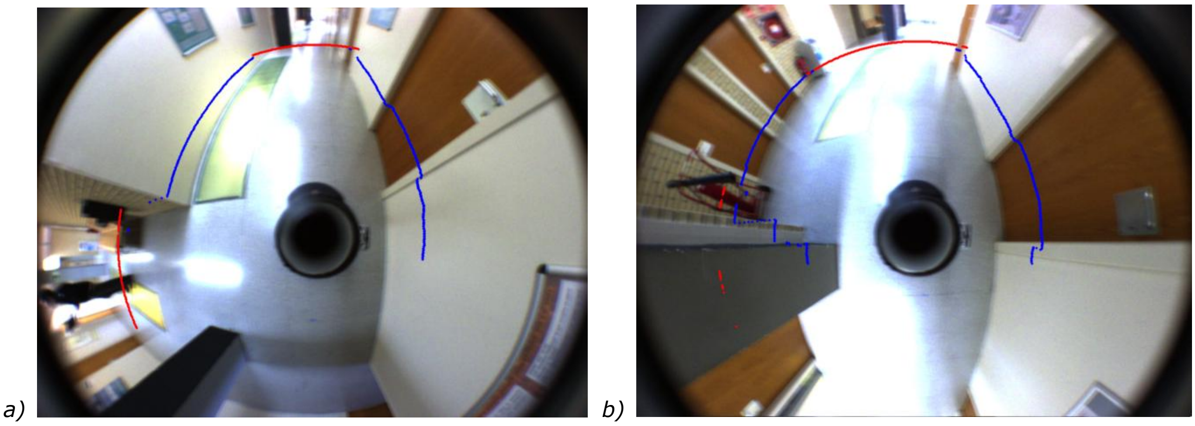


Figure 3.13 LRF points projected onto the omnidirectional image. a) Level 1 of the PIV building. b) Level 2 of the PIV building.

3.5. TEXTURED VERTICAL EDGE FEATURES

The most relevant solutions to SLAM are focused on the feature-based approach, where feature descriptors are extracted from laser scans or images to solve the problem of matching observations to landmarks. Other alternative approaches are: in [Nieto et al., 2007] 2D raw range data is used to extract saliency using the Iterative Closest Point (ICP) [Besl & McKay, 1992] algorithm regardless the geometric representation of the environment; [Silveira et al., 2008] computes the camera pose and the scene structure considering illumination changes using the entire image intensities to extract saliency regions, and then perform a non-linear minimization.

In recent years, appearance-based mapping and localization has gained special attention since these methods use a richer description of the environment giving more cues to improve robot mapping and localization as [Cummins & Newman, 2009] and [Ranganathan & Dellaert, 2011]. These approaches are featured-based (e.g. SIFT [Lowe, 2004], SURF [Bay et al., 2008], Harris corners [Harris & Stephens, 1988], etc.) and they present a probabilistic framework to build appearance-based topological maps, assuming environments with enough texture information to extract the image features.

Appearance-based methods also exploit the environment structure given mainly by vertical and horizontal edges, such as doors, planes, etc. to obtain a better representation of the robot surroundings. In this sense, [Wongphati et al., 2008] proposed a solution using omnidirectional images and bearing information of vertical edges to solve SLAM, but this approach needs two consecutive frames to extract the position of the vertical edges, delaying the robot pose estimation. The authors in [Scaramuzza, 2008] proposed a vertical edge descriptor which robustly matches catadioptric vertical edges. This descriptor is valid locally, requiring two or more views to extract the scene vertical edges position.

Besides, common available laser range finders work in a plane parallel to the ground, and then it limits the environment representation to 2D. Combining vision sensors with laser range finders increase the perceptual information, but monocular or stereo cameras have limited field of view affecting their perception due to occlusions and feature lifetime observation. However, omnidirectional cameras have important advantages as long-term landmark tracking, wide field of view, and robustness to occlusions. Therefore, combining an omnidirectional camera with a LRF for feature extraction has many advantages: all the laser trace can be used to extract environment features on the image plane, depth information can be embedded into the omnidirectional image, 3D feature information can be recovered, and once the calibration between these sensors is performed it can be used in real time. In [Gallegos et al., 2010] a system based on a LRF and an omnidirectional camera is described to obtain a map of the environment using scan matching and vertical edges; however the authors neither solve the SLAM problem nor define a data association method. In [Mei, 2007] a hybrid sensor is presented and composed of an omnidirectional camera and a LRF, where the laser trace is projected onto the omnidirectional image and it is used to extract salient features on the image plane using 1D intensity signals around the detected vertical edge as a local data association method; however, pixel intensity based methods need short base-line movements to be used with custom similarity metrics and do not produce discriminative enough features for data association.

In this work, a sensor model based on the extrinsic calibration between a LRF and an omnidirectional camera [Bacca et al., 2010] in order to extract the 3D position of vertical edges in indoor environments is presented. This extrinsic calibration was obtained using the Simultaneous Parameter Estimation method, which was described in Section 3.4.2. Data association is very important for probabilistic frameworks which deal with mapping and localization. The proposed approach uses a two-step algorithm to solve data association: first, a Joint Compatibility Branch and Bound test (JCBB) [Neira & Tardos, 2001] is performed; second, a geometric constraint based on the catadioptric projection of the scene vertical edges and a rigid transformation is considered in order to resolve the ambiguous associations obtained from the JCBB test. The experimental validation was performed integrating our sensor model into the FastSLAM [Montemerlo & Thrun, 2003] framework and using a dataset collected as describe in Section 2.5.

3.5.1. Sensor Model

The Algorithm 3.4 depicts the basic steps to build the sensor model proposed in this work. It consists of six stages as follows: firstly, using Algorithm 3.1 to 3.3 the LRF corners, the LRF segments and the vertical edges are detected; secondly, using the line projection model and its conic representation [Barreto & Araujo, 2005] the LRF segments are projected on the sphere and the image plane, being able to compute their normal vectors and conic matrices respectively; thirdly, the LRF corners are projected on the image plane and their uncertainties found; fourthly, the vertical edge model on the image plane is computed and their intersections found with the conic corresponding to the LRF segment; fifthly, using the corner uncertainties in the image plane and the intersects computed above, the data association vector is found relating each corner with the vertical edge intersect; last, the vertical edge observation is performed.

Algorithm 3.4 Sensor model algorithm.

```
lrfSegments = getLRFsegments();
lrfCorners = getCornersFromSegments();
veList = getCatadioptricVerticalEdges();
veListFOV = getVerticalEdgesInFOV(veListSmooth);
veListSmooth = doVerticalEdgeSmooth(veList);
// Computing the segment normals in the sphere and their conic matrix on the image plane.
lrfSegNormals ← [];
lrfSegConic ← [];
foreach S ∈ lrfSegments
        lrfSeg = doTransformLRFsegment(S);
        lrfSegSph = doProjectSegment2Sphere(lrfSeg);
        lrfSegNormals ← getNormalVector(lrfSegSph);
        lrfSegConic ← getConicMatrix(lrfSegSph);
end
// Computing the corners uncertainty on the image plane.
Jp = getPolarCartesianJacobian(lrfCorners);
lrfCornersCamera = getTransform2CameraFrame(lrfCorners);
Jr = getTransformationJacobian(lrfCornersCamera);
Js = getInfinitePlaneJacobian(lrfCornersCamera);
lrfCornersUnd = getPointsOnInfinitePlane(lrfCornersCamera);
Jd = getDistortionJacobian(lrfCornersUnd);
lrfCornersDist = getDistortedPoints(lrfCornersUnd);
Ji = getCameraProjectionJacobian();
lrfCornersImg = getPointsOnImage(lrfCornersDist);
sensorSigma = getSensorSigma(lrfCorners);
lrfCornersSigma = getUncertaintyOnImagePlane(Jp, Jr, Js, Jd, Ji, sensorSigma);
// Vertical edge intersection with the LRF segment conics on the image plane.
veMeasurement ← [];
foreach ve ∈ veListFOV
        // Computing edge model on image plane.
        xyImg = getVEimagePoints(ve);
        veImgModel = doComputingVEmodel(xyImg);
        // Computing segment conics and their intersection with the vertical edge.
        foreach Ns ∈ lrfSegNormals; Cs ∈ lrfSegConic
                // Computing intersections.
                [conicCenter, conicRadii] = getConicCenterRadii(Cs);
                [conicI1 conicI2] = getConicIntersects(coniCenter, conicRadii, veImgModel) ;
                // Validating intersections with the corner uncertainties.
```

```
39.                 H = getCornerIntersectsDataAssociation(conicI1, conicI2, lrfCornersImg, lrfCornersSigma);
40.                 If empty(H)
41.                         Continue;
42.                 else
43.                         veMeasurement ← doMeasurement(H, lrfCorners);
44.                 end
45.         end
46. end
47.
```

The first stage corresponds to lines 1 to 5 in the Algorithm 3.4. Line 1 considers the LRF data process considered in Section 3.1. However, after segmenting the LRF data additional possible corners are introduced (Line 2), since the LRF data is branched at the point specified by the cost function defined in Equation 3.11. Afterwards, the vertical edges are detected using the process described in Section 3.2 (Line 3). The vertical edges detected are spread in the 360º field of view of the omnidirectional image. However, the LRF has a limited field of view of 240º. Then those vertical edges on the image out of the field of view of the LRF are filter out (Line 4). Furthermore, the vertical edges are smoothed computing their line model on the image plane. The line model used is the standard slope-intersect, which is computed in the least squares sense using singular value decomposition (SVD) considering all the vertical edge points (Line 5).

The second stage is performed between lines 7 and 14. In line 10, the LRF segment is transformed with respect to the camera frame using the extrinsic calibration described in Section 3.4. Afterwards, using the projection model explained in Section 3.3 the world points are projected onto the sphere (Line 11). There, using SVD as explained in Section 3.3.2 the normal vector corresponding to the plane (PS) defined by the LRF segment and the mirror projection center is computed (Line 12). And then, using the conic projection model proposed by [Barreto & Araujo, 2005] and the intrinsic camera parameters the catadioptric projection of the LRF segment is defined by the conic Equation 3.39.

$$\boldsymbol{C_I} = \begin{bmatrix} \gamma_1 & \gamma_1 s & u_0 \\ 0 & \gamma_2 & v_0 \\ 0 & 0 & 1 \end{bmatrix}^{-T} \begin{bmatrix} n_x^2(1-\xi^2) - n_z^2\xi^2 & n_x n_y(1-\xi^2) & n_x n_z \\ n_x n_y(1-\xi^2) & n_y^2(1-\xi^2) - n_z^2\xi^2 & n_y n_z \\ n_x n_z & n_y n_z & n_z^2 \end{bmatrix} \begin{bmatrix} \gamma_1 & \gamma_1 s & u_0 \\ 0 & \gamma_2 & v_0 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \qquad 3.39$$

Where, the first and third matrix correspond to the intrinsic camera parameters, and the second one correspond to the projection of the conic on the image plane at infinity; *$[n_x\ n_y\ n_z]^T$* is the normal vector which parameterizes the plane ***P_S*,** and ζ is the mirror parameter.

The third stage corresponds to lines 16-26. Here, the projection model described in Section 3.3.1 is used to extract the Jacobians in order to estimate the LRF corners uncertainties on the image plane. Also, it is worth recalling that given two random variables a and b related by a non-linear function ***a* = *f(b)***, then the variance Σa may be approximated by the variance of $\boldsymbol{\Sigma_b}$ using Equation 3.40.

$$\boldsymbol{\Sigma_a = \nabla f_b \Sigma_b \nabla f_b{}^T} \qquad 3.40$$

Where $\boldsymbol{\nabla f_b}$ is the Jacobian of ***f(.)*** taken with respect to ***b***. Applying Equation 3.40 successively over all the projection process the corners uncertainties on the image plane can be estimated. To do so, the Jacobians corresponding to the following functions have to be computed: the transformation from polar to Cartesian coordinates, the transformation to the camera frame, the projection to the infinite image plane, the distortion model and the camera projection model. The Jacobians of these functions are evaluated in the corresponding mean value obtained evaluating each non-linear function. The corners uncertainties on the image plane can be defined as depicted in Equation 3.41.

$$\boldsymbol{\Sigma}_{\boldsymbol{Ci}}^{\boldsymbol{I}} = \boldsymbol{J_K J_D J_S J_R J_P} \begin{bmatrix} \sigma_r^2 & 0 & 0 \\ 0 & r\sigma_{\theta c}^2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \boldsymbol{J_P^T J_R^T J_S^T J_D^T J_K^T} \qquad 3.41$$

Where, σ_r^2 and $\sigma_{\theta c}^2$ are the variances of the range and orientation of the LRF, *r* the range of the *i-th* corner in the LRF frame, $\boldsymbol{J_P}$ is the Jacobian of the polar to Cartesian coordinate transformation, $\boldsymbol{J_R}$ is the Jacobian of the transformation to the camera frame, $\boldsymbol{J_S}$ is the Jacobian of the projection function to the image plane at infinity, $\boldsymbol{J_D}$ is the Jacobian of the distortion function which was taken from the calibration toolbox presented in [Mei, 2006] and $\boldsymbol{J_K}$ is the Jacobian of the camera projection function. These Jacobians can be found in the Appendix A. Figure 3.14 shows how the uncertainty projection works. In this figure, the dominant vertical edges are displayed in magenta, the LRF trace is projected using the extrinsic calibration discussed in Section 3.4 (dotted blue trace), the LRF corners are displayed using red crosses and their uncertainties correspond to the ellipses painted in green.

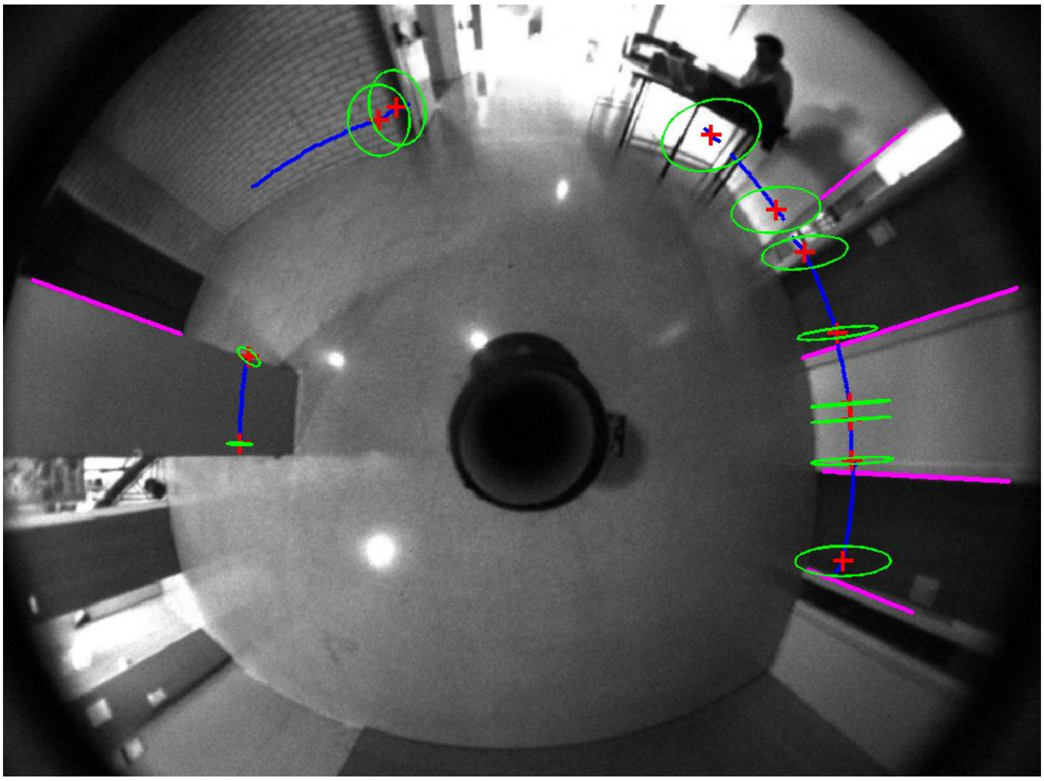


Figure 3.14 Catadioptric projections. In magenta the dominant vertical edges are displayed, the LRF trace is projected in blue, the LRF corners are drawn using red crosses and their uncertainties are shown in green.

The fourth stage corresponds to lines 36 and 37. The SVD is used to compute the slope and intersect of each vertical edge, and using the conic matrix computed in second stage the conic center and radii are extracted. The intersection between the vertical edge line model and the conic corresponding to the catadioptric projection of the LRF segments is computed as follows:

1. A point $\boldsymbol{P_L}$ which belongs to the conic has to satisfy the equality $\boldsymbol{P_L^T C_I P_L} = 0$, where $\boldsymbol{C_I}$ is defined by Equation 3.38. As a result of operating over the conic equality, and adding the vertical edge line model, the system to solve is depicted in Equation 3.42.

$$[x_L \quad y_L \quad 1]^T \begin{bmatrix} A & B & D \\ B & C & E \\ D & E & F \end{bmatrix} \begin{bmatrix} x_L \\ y_L \\ 1 \end{bmatrix} = Ax_L^2 + Bx_L y_L + Cy_L^2 + Dx_L + Ey_L + F = 0$$

$$y_L = Mx_L + k \qquad 3.42$$

Where, $\boldsymbol{P_L}$ = *[x$_L$ y$_L$ 1]^T* is the intersection point, *M* and *k* are the slop and intersect of the vertical edge.

2. Simplifying Equation 3.42, it can be expressed in form of a quadratic equation of x_L as shown in Equation 3.43, which can be solved in closed form.

$$x_L^2(A + MB + CM^2) + x_L(Bk + 2CM + D + EM) + (Ck^2 + kE + F) = 0 \qquad 3.43$$

3. Taking into account the limitation of the slope-intersect model for the vertical edges, if the vertical edge slope is high the vertical edge model is switched to $x_L = (y_l - k)/M$. By doing this, the quadratic equation has the form of Equation 3.44.

$$y_L^2\left(\frac{A}{M^2} + \frac{B}{M} + C\right) + y_L\left(\frac{D}{M} - \frac{Bk}{M} - \frac{2Ak}{M^2} + E\right) + \left(\frac{Ak^2}{M^2} - \frac{Dk}{M} + F\right) = 0 \qquad 3.44$$

4. As a result, there are two intersects $\boldsymbol{P_{L1}}$ = *[x$_{L1}$ y$_{L1}$]^T* and $\boldsymbol{P_{L2}}$ = *[x$_{L2}$ y$_{L2}$]^T* for each catadioptric projection of the LRF segment.

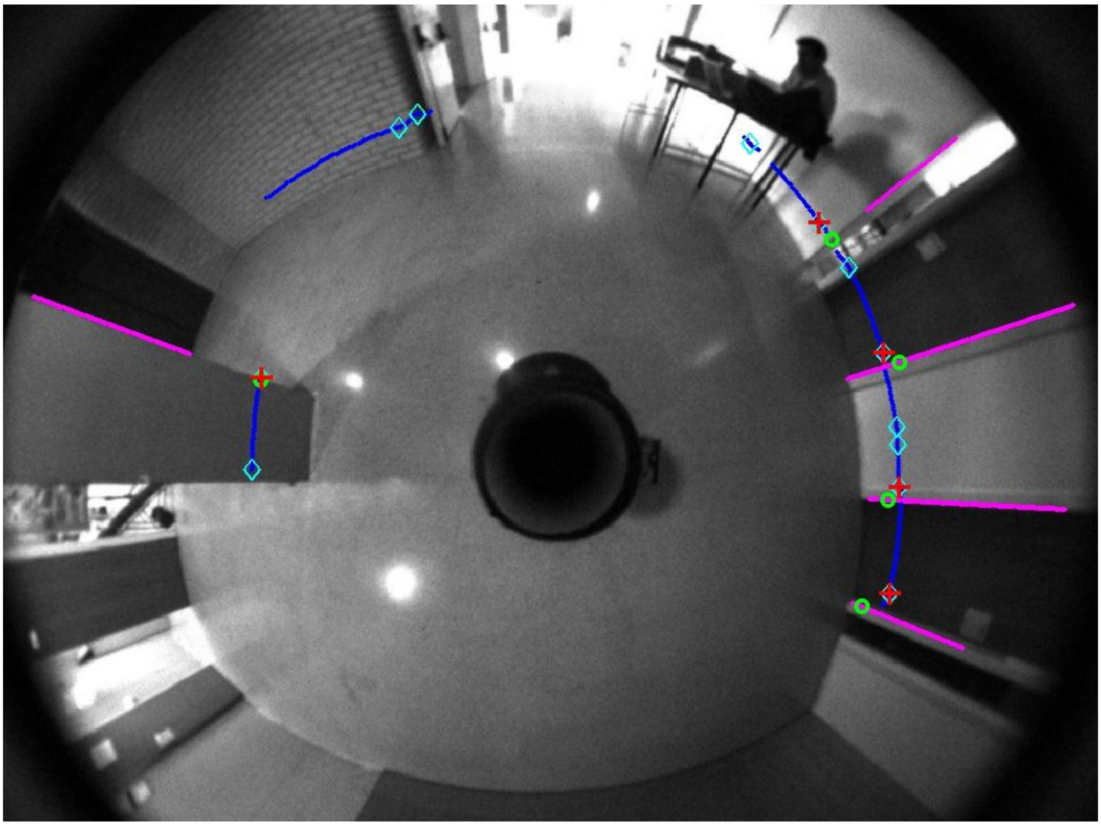


Figure 3.15 Vertical edges and their LRF corner associations. Diamond-shaped points represent the LRF corners; circle-shaped points represent the conic intersections; and the cross-shaped points the associated LRF corners.

The fifth stage is depicted in line 39 of the Algorithm 3.4. At this point of the process there are a set of LRF corners mapped on the image plane, and a set of intersects corresponding to the solution of Equation 3.43 or 3.44. The data association is done using the Joint Compatibility Branch and Bound (JCBB) test [Neira & Tardos, 2001]. By doing this, the data association vector has the length of the observations and in each field the associated corner ID. To do so, it is necessary define the uncertainty of the vertical edge intersects in the image plane. This is not an easy problem, however in [Meltzer & Soatto, 2008] a wide baseline correspondence method using edges is proposed, and the initial guess of the edge region of interest was based on the size mask used to detect the edges on the image plane. In this way, the uncertainty of the intersect points between the vertical edge and the catadioptric projection of the LRF segments is defined as a diagonal matrix of σ_W = *ceil(WINDOW_CANNY / 2)*. In this work, the edge Canny detector works with a image window of 7. Figure 3.15 shows an example of the resulting data association described above. In this figure, the conic intersects are shown using circle-shaped points and the associated LRF corners are shown using cross-shaped points. The remaining LRF corners are also shown using diamond-shaped points.

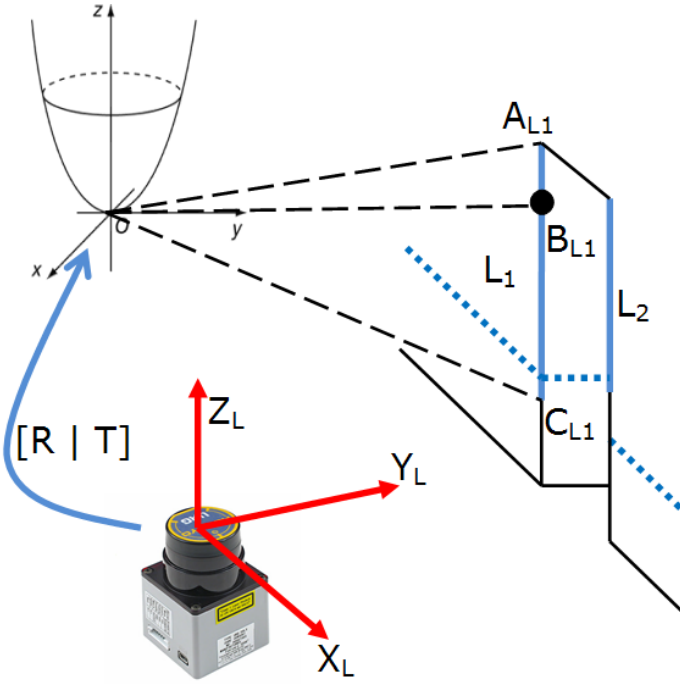


Figure 3.16 Vertical edge position measurement.

Once the data association of the vertical edge / conic intersects and the projected corner points are done, the measurement process starts. The aim of this process is to find the 3D initial and ending points of the vertical edge, the vector director and the vertical edge length. This information is joined to the range and bearing measurements performed above. Figure 3.16 shows how the vertical edge properties described above can be computed. In this figure two typical vertical edges are presented. The edge L_1 crosses the LRF trace (dotted line) and the edge L_2 does not. $\boldsymbol{A_{L1}}$ and $\boldsymbol{C_{L1}}$ are the initial and ending points of L_1. $\boldsymbol{B_{L1}}$ is the range of L_1 and taken from the LRF corner associated to L_1.

The process described here is based on the following angle definitions: $\theta_{A1} = \angle \boldsymbol{zOA_{L1}}$ and $\theta_{C1} = \angle \boldsymbol{zOC_{L1}}$. The initial and ending points of the vertical edge on the image plane are lifted and projected on the sphere. This process is done using the same LUT mentioned in Section 3.3.2 to speed-up the computation time. These sphere points are used to compute the angles θ_{A1} and θ_{C1}, from which the scene points $\boldsymbol{A_{L1}}$ and $\boldsymbol{C_{L1}}$ can be expressed as depicted in Equations 3.45 and 3.46.

$$\boldsymbol{A_{L1}} = \begin{bmatrix} x_A \\ y_A \\ z_A \end{bmatrix} = \begin{bmatrix} \rho_B \cos(\theta_{L1}) \\ \rho_B \sin(\theta_{L1}) \\ \frac{\rho_B}{\tan(\pi - \theta_{A1})} \end{bmatrix} \qquad 3.45$$

$$\boldsymbol{C_{L1}} = \begin{bmatrix} x_C \\ y_C \\ z_C \end{bmatrix} = \begin{bmatrix} \rho_B \cos(\theta_{L1}) \\ \rho_B \sin(\theta_{L1}) \\ \frac{\rho_B}{\tan(\pi - \theta_{C1})} \end{bmatrix} \qquad 3.46$$

Where, ρ_B is the range of the vertical edge and θ_{L1} is the bearing information of the vertical edge with respect to the camera frame. It is worth noting that the SLAM algorithm will receive the typical range and bearing measurement model, however the environmental representation is performed using Equations 3.45 and 3.46. As a result, the LRF data, the range/bearing data of the vertical edges and their 3D initial and ending points give an enriched representation of the environment, in other words an appearance-based representation.

The observation model used by the SLAM algorithm is shown in Equation 3.47. This is a non-linear function in terms of the vertical edge position $\boldsymbol{m_{L1}} = [m_x, m_y]$ and the state vector of the mobile robot $\boldsymbol{X_t} = [x_R, y_R, \theta_R]$. Most of the SLAM techniques estimate robot position and the features position through the linearization of the measurement model. For self-containment, the Equations 3.48 and 3.49 show the Jacobians with respect to the vertical edge position and the robot state vector respectively.

$$\boldsymbol{Z_n} = \begin{bmatrix} r_n \\ \phi_n \end{bmatrix} = \begin{bmatrix} \sqrt{(m_x - x_R)^2 + (m_y - y_R)^2 + {m_z}^2} \\ \operatorname{atan2}\left(\frac{m_y - y_R}{m_x - x_R}\right) - \theta_R \end{bmatrix} \qquad 3.47$$

$$\mathbf{J_m} = \begin{bmatrix} \frac{dx}{r_n} & \frac{dy}{r_n} \\ \frac{-dy}{d} & \frac{dx}{d} \end{bmatrix} \qquad 3.48$$

$$\boldsymbol{J_x} = \begin{bmatrix} \frac{-dx}{r_n} & \frac{-dy}{r_n} & 0 \\ \frac{dy}{d} & \frac{-dx}{d} & -1 \end{bmatrix} \qquad 3.49$$

Where, $d_x = m_x - x_R$, $d_y = m_y - y_R$ and $d = (m_x - x_R)^2 + (m_y - y_R)^2$.

3.5.2. Data association

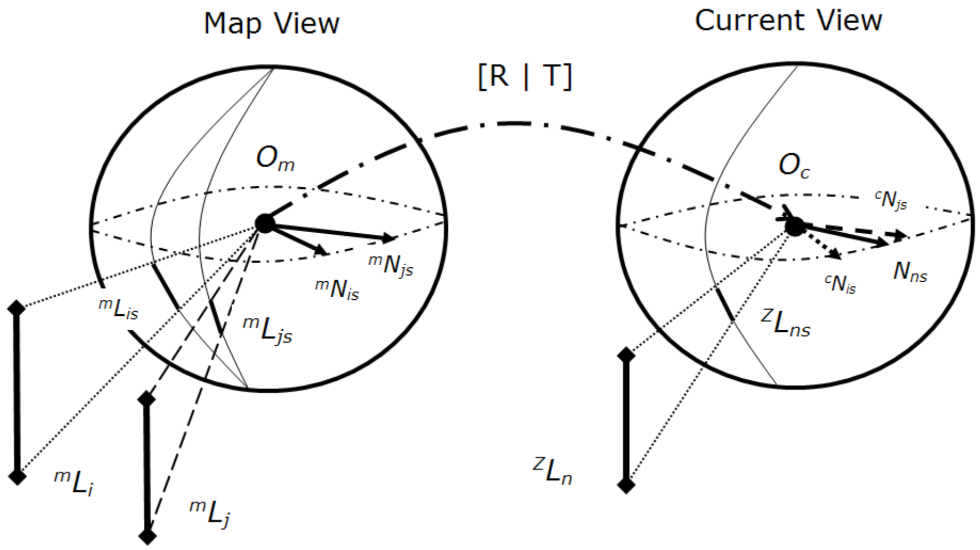


Figure 3.17 Resolving ambiguous data association.

In this work, the JCBB [Neira & Tardos, 2001] was considered to solve the data association problem, such that the correlations between innovations are explicitly taken into account to determine the joint compatibility of a set of pairings. The sensor model proposed in this Section is focused on finding the range and bearing of the vertical edges. However, JCBB associate features depending on the innovations, their covariance and the innovation gate distance metric used (Mahalanobis distance), which are no longer valid when the uncertainty becomes important. Therefore, in this work a method for distinguishing vertical edge features which are close to each other when the measure of uncertainty is not sufficient is proposed in order to obtain a better data association.

In first place, the JCBB test is used. If a current vertical edge feature is associated with two or more landmarks in the map a second step is performed. Figure 3.17 shows a typical situation, where two views are involved: the current view with one observation associated with two map landmarks. This view is placed at the predicted robot pose $\boldsymbol{O_C}$, where a scene vertical edge $^{\boldsymbol{Z}}\boldsymbol{L_n}$ is observed and projected onto the sphere as $^{\boldsymbol{Z}}\boldsymbol{L_{ns}}$ using the unified projection model; a plane is defined between $^{\boldsymbol{Z}}\boldsymbol{L_{ns}}$ and $\boldsymbol{O_C}$, which can be parameterized by the normal vector $\boldsymbol{N_{ns}}$, since for all points $^{\boldsymbol{Z}}\boldsymbol{L_k}$ belonging to $^{\boldsymbol{Z}}\boldsymbol{L_{ns}}$ the dot product property $^{\boldsymbol{Z}}\boldsymbol{L_k}^{\boldsymbol{T}} * \boldsymbol{N_{ns}} = 0$ is satisfied. On the other hand, two landmarks candidates (i.e. $^{\boldsymbol{m}}\boldsymbol{L_i}$ and $^{\boldsymbol{m}}\boldsymbol{L_j}$) corresponding to the map view placed at $\boldsymbol{O_m}$ have their analogous spherical projections (i.e. $^{\boldsymbol{m}}\boldsymbol{L_{is}}$ and $^{\boldsymbol{m}}\boldsymbol{L_{js}}$) and planes parameterized by the vector normal $^{\boldsymbol{m}}\boldsymbol{N_{is}}$ and $^{\boldsymbol{m}}\boldsymbol{N_{js}}$ respectively. In this work, the robot pose in which a feature was seen for the first time is saved; this allows us to relate $\boldsymbol{O_C}$ and $\boldsymbol{O_m}$ through a transformation defined by $\boldsymbol{R}$ and $\boldsymbol{T}$.

The intuitive idea behind our method is: given the current observation parameterized by the vector normal $\boldsymbol{N_{ns}}$ and the map landmarks candidates, in this case parameterized by $^{\boldsymbol{m}}\boldsymbol{N_{is}}$ and $^{\boldsymbol{m}}\boldsymbol{N_{js}}$ in the map view, associate the observation $^{\boldsymbol{Z}}\boldsymbol{L_{ns}}$ with the closer landmark map parameterized by the transformed vector normal $^{\boldsymbol{C}}\boldsymbol{N_{is}}$ and $^{\boldsymbol{C}}\boldsymbol{N_{js}}$ in the current view. This is formally described by Equation 3.50.

$$c_i = \min_i \sum_{k=1}^{F} \left\| \; ^{Z}\boldsymbol{L}_{\boldsymbol{k}}^{\boldsymbol{T}}.\boldsymbol{N}_{\boldsymbol{ns}} - {}^{C}\boldsymbol{L}_{\boldsymbol{i,k}}^{\boldsymbol{T}}.{}^{C}\boldsymbol{N}_{\boldsymbol{is}} \right\| \qquad 3.50$$

Where, F is the total number of points of the observed feature, $^{\boldsymbol{Z}}\boldsymbol{L_k}$ is the *k-th* observed point, $^{\boldsymbol{C}}\boldsymbol{L_{i,k}}$ is the *k-th* point of the *i-th* landmark candidate with respect to $\boldsymbol{O_C}$ and c_i is the resulting associated map feature index. Equation 3.50 selects the landmark candidate with the minimum difference between the projection of the observed points $^{\boldsymbol{Z}}\boldsymbol{L_k}$ on the plane parameterized by $\boldsymbol{N_{ns}}$, and the projection of the *i-th* map feature on the plane parameterized by $^{\boldsymbol{C}}\boldsymbol{N_{is}}$ as the final feature association.

3.5.3. *Results*

The dataset collected for this work and described in Section 2.5.2 was used to obtain the results shown in this Section. It is worth noting that the dataset images were collected in diverse environmental conditions, as those shown in Figure 3.18. In this figure, it can be observed that there are illumination changes and occlusions caused by pedestrians assuring in this way a real-world experiment. The data set corresponding to the first floor has a path length of ~100m and that on the third floor a length of ~65m. In addition, the robot mobile platform and the perception system were described in Section 2.5.1.

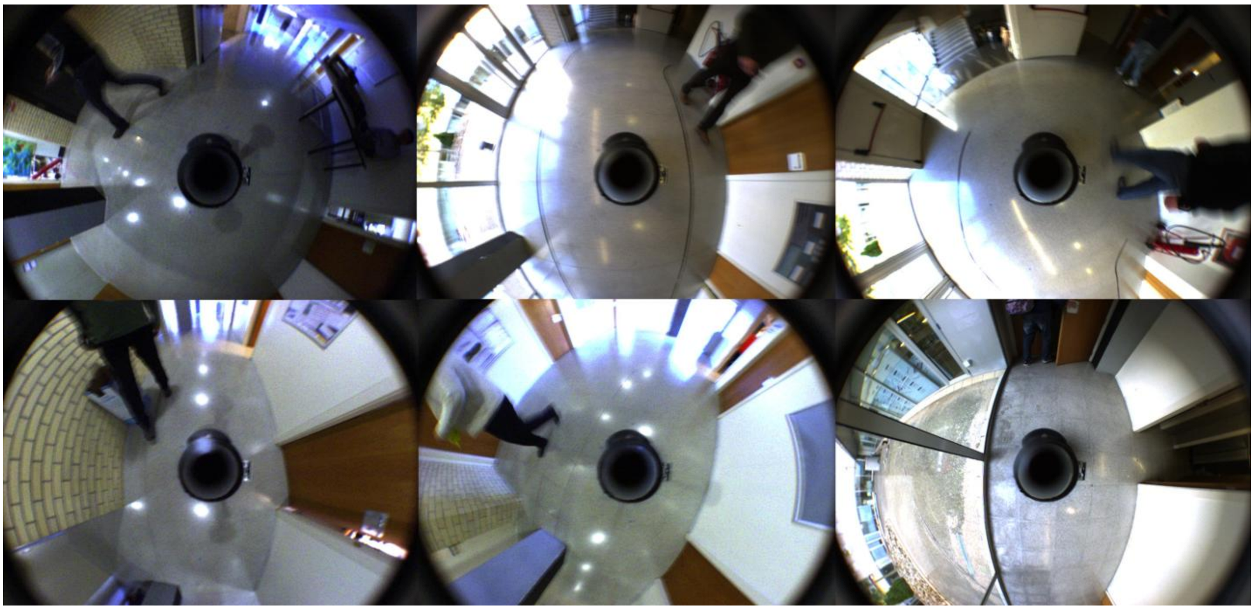
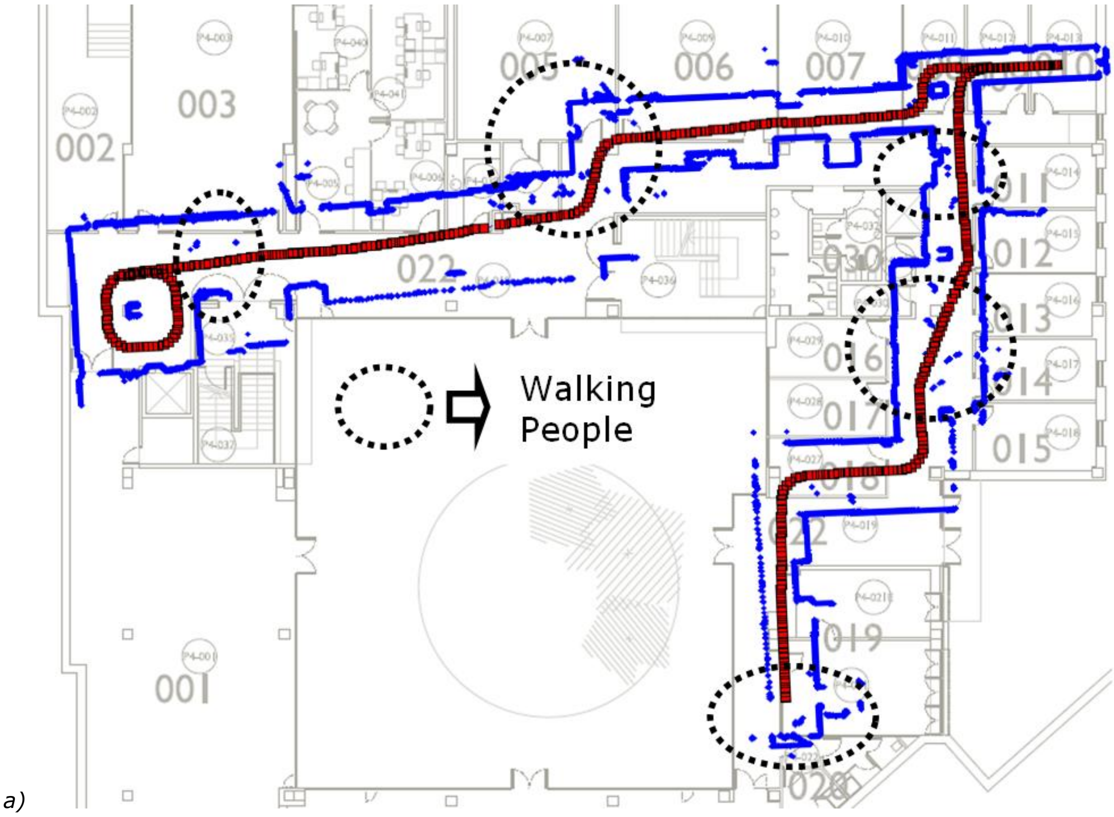


Figure 3.18 Omnidirectional images samples taken from the collected data set. Each row corresponds to the first and third floor of the PIV building, respectively.



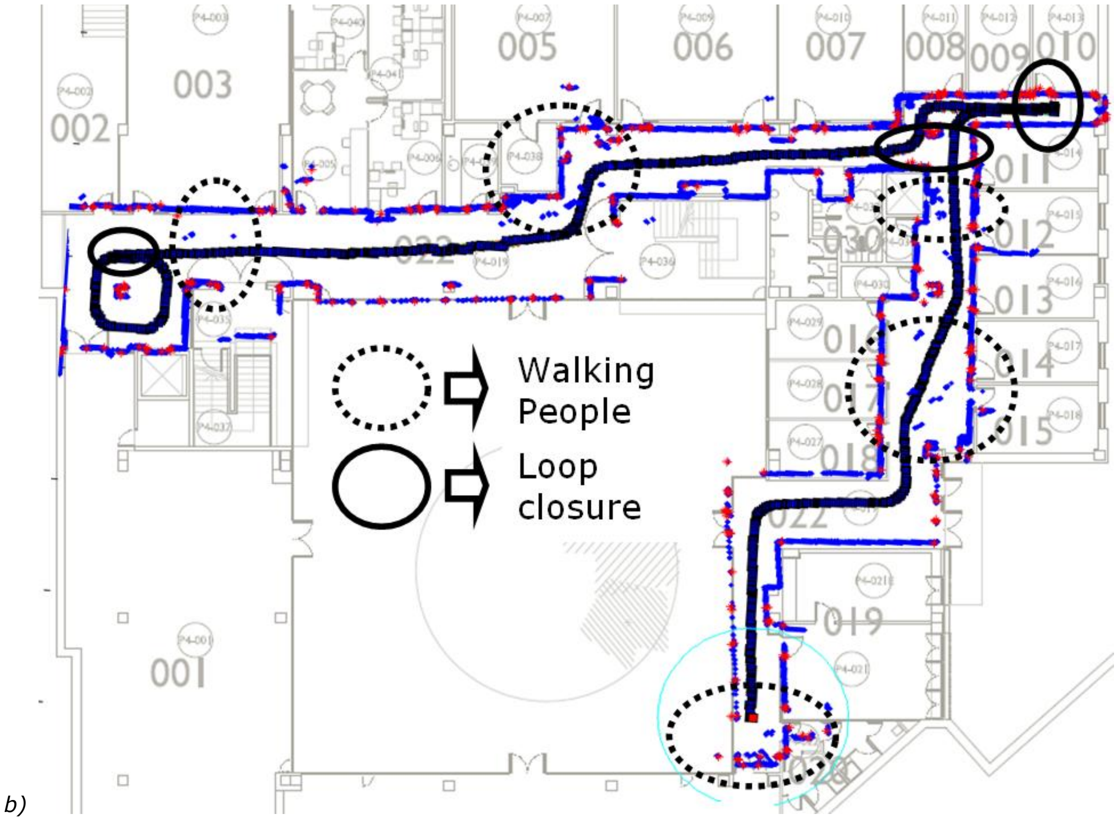


Figure 3.19 First experiment: SLAM at the first floor of PIV building. a) Scan matching map. b) Estimated map using our approach.

Simultaneous mapping and localization for mobile vehicles is a challenging task which depends on the filter (EKF, UKF or particle filter), but in which the sort of perception system used and the method to extract the salient features are crucial. Generally speaking, the SLAM problem can be solved using iterative methods (Kalman filters, particle filters and occupancy grids), or using global minimization methods (Bundle adjustment and expected maximization) [Durrant-Whyte & Bailey, 2006]. The family of Kalman filters assumes a high dimensional Gaussian distribution of the error source, the robot pose and the map features, however the mapping and localization problem involves non-linear functions and unknown error sources, the latter cannot be modeled using an exact mathematical representation. In this context, particle filters can handle this kind of problems sampling from an estimate probabilistic distribution and improving it recursively. These reasons motivate the integration of the sensor model proposed in this Section within a popular SLAM algorithm as FastSLAM [Montemerlo & Thrun, 2003], which uses particle filtering and it has been adapted to support the sensor model described above.

The results shown in this section were divided in two groups. Firstly, a qualitative comparison between the map obtained using a standard scan-matching technique [Besl & McKay, 1992] and the map obtained using the FastSLAM algorithm with the sensor model described above. Secondly, a quantitative comparison between this work and the G^{2}O framework [Kummerle et al., 2011] in terms of the robot pose error.

Figure 3.19 shows two maps, the first one (Figure 3.19a) was built using scan matching and the second one (Figure 3.19b) depicts the estimated map using the approach exposed above, both on the CAD map of the first floor of the PIV building in order to see their correspondence. The estimated map in Figure 3.19b shows the robot path and the landmarks estimated positions of the best particle. Comparing these both maps is clear that the approach proposed above obtains a better match with respect to the CAD drawing. The dataset was taken in presence of pedestrians, who are shown by the arrows in both maps. Despite of this fact, the FastSLAM algorithm integrated with the sensor model proposed obtains a coherent representation of the environment.

Figure 3.20 shows a zoomed region of Figure 3.19b. This figure shows the consistency between the laser scan matching and the corresponding vertical edges in the scene. The vertical edges depicted in this figure correspond to the most predominant ones. In this figure the vertical edge position uncertainties are also shown, as well as the measured length of the vertical edges.

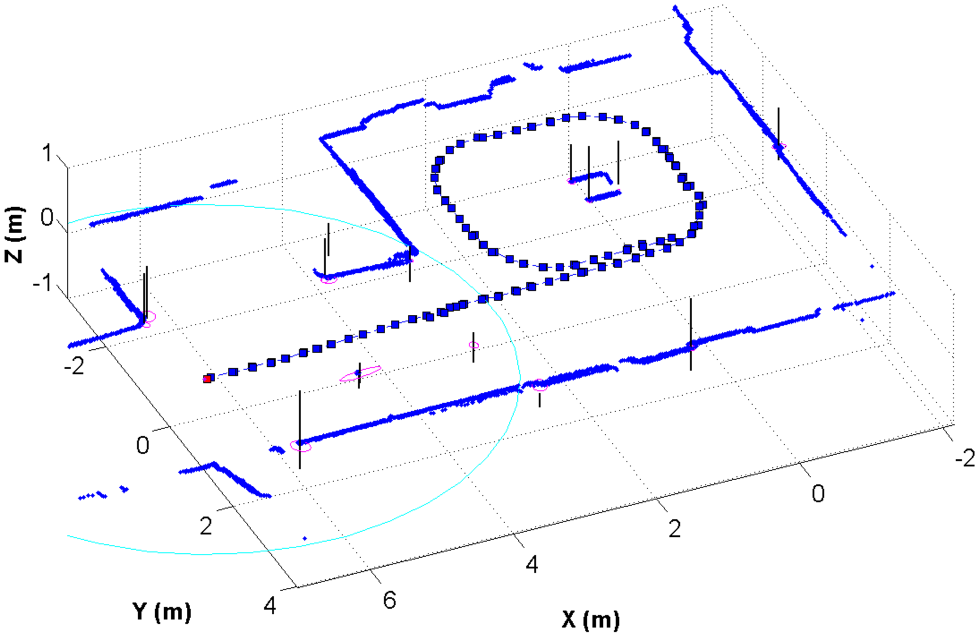
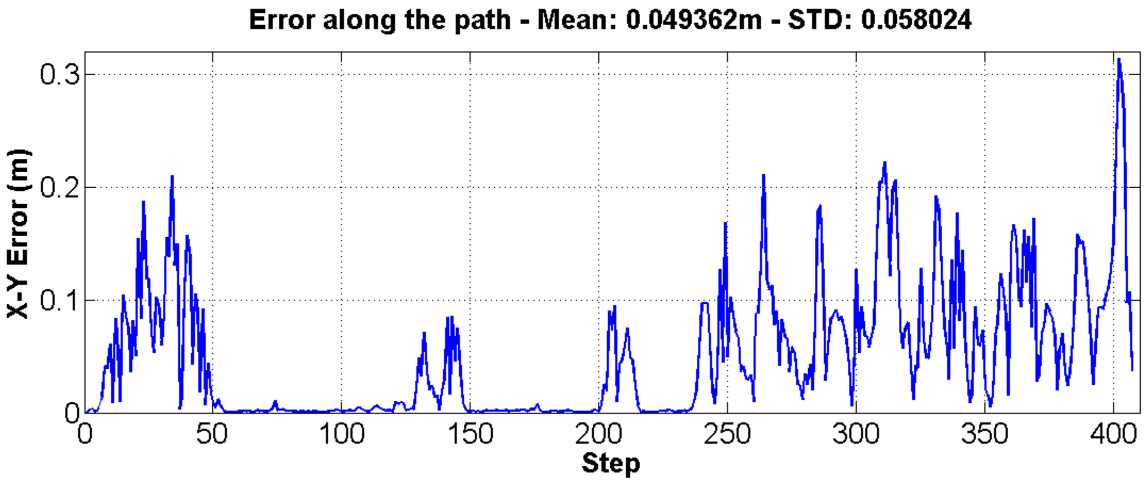
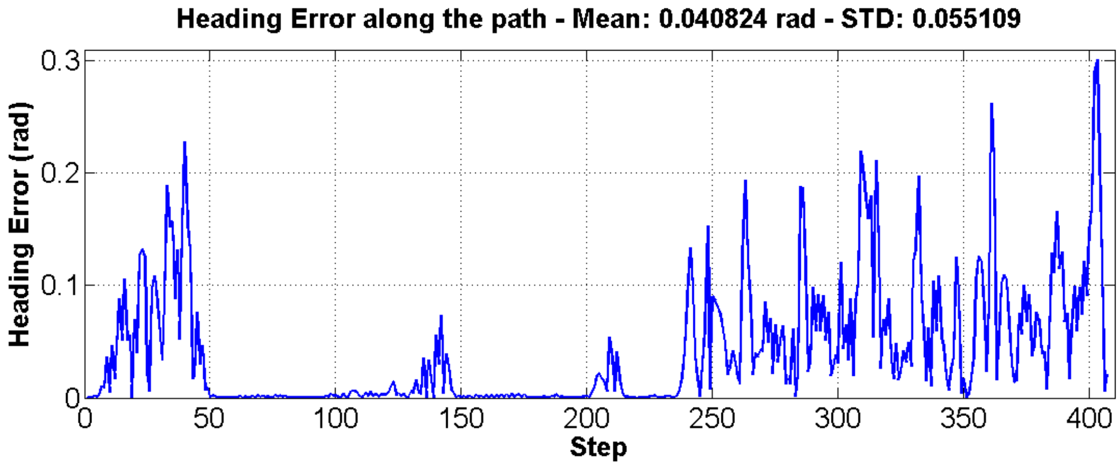


Figure 3.20 Detailed view of the predominant vertical edges of the map depicted in Figure 3.19.



a)



b)

Figure 3.21 a) X-Y error in meters along the robot trajectory. b) Heading error in radians along the robot trajectory.

In order to obtain the ground truth data the G^{2}O framework was used. This algorithm provides a solution for batch optimization of graph-based nonlinear error functions. The robot trajectory of the first experiment as a graph was introduced, including nodes, edges and constraints. The output of the algorithm is an estimate of the robot trajectory obtained after a nonlinear minimization using Levenberg-Marquardt. Using this output the X-Y error and the heading error of the robot along its path can be extracted.

Observing the X-Y and heading error shown in Figure 3.21a and Figure 3.21b, it is worth noting the error decreasing at steps 50, 150 and 225 approximately due the loop closure as depicted in Figure 3.19 within the circles with solid line. After step 240 approximately, the particle filter depends of the sensor model for accurately detect the vertical edges and associate them with map features. This can be observed in the shape of the error graphs, since periodically there is a rise and fall of the X-Y and heading error due to vertical edge features coming into and leaving from the field of view.

3.6. **DISCUSSION**

In this chapter, the feature extraction approach to model the environment was presented. In this work, the environment modeling is composed by a set of vertical edges detected combining two different kind of sensors: an omnidirectional camera and a LRF. The sensor model presented is based on the extrinsic calibration of a LRF and an omnidirectional camera. Using this calibration, 3D vertical edges were extracted and considered as observations in the implementation of the FastSLAM algorithm. The sensor model described provides metric information and appearance-based environmental description using one omnidirectional image and the corresponding LRF trace.

The feature extraction method proposed included several important steps: firstly, the LRF data was segmented in corners and lines, the latter using a modified implementation of the popular split-and-merge algorithm. Secondly, the vertical edges were detected on the image plane using an algorithm based on that proposed by [Bazin et al., 2007]. Thirdly, using the extrinsic calibration proposed in [Bacca et al., 2010] the LRF points can be projected onto the omnidirectional image. Last, using the LRF corners uncertainties on the image plane, the catadioptric projections of the LRF line segments and a robust data association method as the JCBB, the range and bearing measurements of each vertical edge were found. Also, an extra data association method was proposed using a geometrical constraint and the lifted vertical edges on the sphere.

Partial analysis of results were presented, particularly those regarding the extrinsic calibration between the LRF and the omnidirectional camera, the LRF corners uncertainties, and the vertical edges / LRF corners data association. However, a complete test was performed in the context of the robot mapping and localization, due its importance within the robotics community and the big impact the sensor model has over the performance of the SLAM solutions. The sensor model proposed was shown to perform well in a very challenging environment, where illumination changes and occlusions by pedestrians were part of the data set used.

CHAPTER 4

4. *FEATURE STABILITY HISTOGRAM MODEL*

CONTENTS

Typical SLAM techniques assume static environments, and they build a map without taking into account the real-world conditions as pedestrians, moving obstacles, perceptual aliasing, weather changes, occlusions and robot-human interaction. Then, how can a mobile robot update its internal representation of the environment and its location on it when the appearance of the environment is changing?. The Feature Stability Histogram (FSH) is a solution proposed in this work to deal with changing environments and long-term mapping and localization. The main idea behind this is to classify the features of the environment as stable and non-stable. To do so, the FSH is inspired on the human memory model proposed by [Atkinson & Shiffrin, 1968] *in order to sequentially build a histogram of the feature strengths which is updated once the feature is re-observed. Stable features, belonging to the Long-Term Memory (LTM) are used for localization and mapping. On the other hand, non-stable features belonging to the Short-Term memory (STM), can be part of the LTM depending on their strength. The use of the FSH method indicates that it can adapt the internal map representation over time to localize the robot. Two real-world experiments were carried out to validate the FSH approach: firstly, a static LRF-based experiment to show that the proposed method is able to update the environmental representation; last, an appearance-based topological mapping experiment whose results include a comparison with the approach proposed in*

[Dayoub et al., 2011] *and the popular Bag-of-Words approach* [Sivic & Zisserman, 2003]*.*

4.1. **INTRODUCTION**

The motivation of this work is to improve mapping and localization in long-term operation and in dynamic environments. The Feature Stability Histogram (FSH) [Bacca et al., 2011] is the main contribution of this work in order to be used in conjunction of typical SLAM solutions. This innovative feature management approach for mapping and localization is able to cope with changing environments, long-term operation, and also contributes to the semantic environment representation.

Considering the long-term techniques reviewed in Section 2.3, the human memory model proposed by [Atkinson & Shiffrin, 1968] has been an inspiration for many works in the robotics community. In this work, the FSH model modifies the [Atkinson & Shiffrin, 1968] model in order to consider the criticism drawn to it by psychologists and neuroscientists [Baddeley, 2003] [Llinas, 2002]. Furthermore, the rehearsal process which defines the method to promote STM features to LTM features was adapted to its use in the SLAM context. Last, using the rehearsal method proposed the STM feature pruning can be carried out considering the oldest feature timestamp.

This chapter is organized as follows: Section 4.2 presents the human memory model basis; the FSH model inspired on the modified human memory model is described in Section 4.3; Section 4.4 describes the probabilistic foundations in order to integrate the FHS model to typical SLAM solutions.; Section 4.5 shows the experimental results which test the main operating principle of the FSH model; last, the final remarks are presented in Section 4.6.

4.2. **HUMAN MEMORY MODEL**

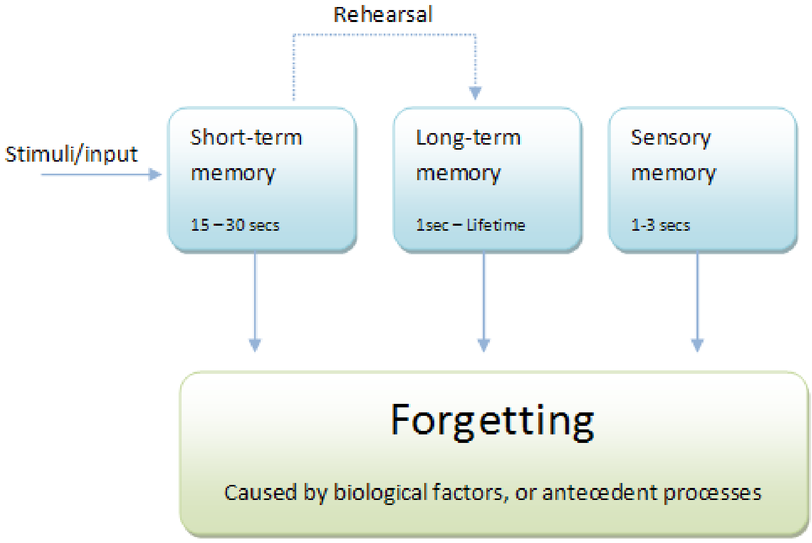


Figure 4.1 Atkinson and Shiffrin model of human memory [Atkinson & Shiffrin, 1968].

For years the scientific community has been finding inspiration in nature, even though probabilistic localization models have their origins in how the "place cells" in the hippocampus works. In this work, the Atkinson and Shiffrin memory model [Atkinson & Shiffrin, 1968] was used to distinguish stable features from unstable ones, and then use the stable features for robot mapping and localization. Figure

4.1 shows this model, where four main components are shown: the Short-Term Memory (STM) which retains information long enough to use it; the Long-Term Memory, which retains information for longer periods of time or lifetime; the Sensory Model, which was added afterwards and was experimentally demonstrated to have the capability of the sensing organs to discriminate information for subsequent processing; and the forgetting module, which affects all other components since it was experimentally demonstrated that memories can be forgotten through trace decay. This model proposes entering stimuli inputs in the STM. If these inputs are continuously rehearsed, they become part of the LTM. Information retained in LTM is recalled continuously in lifetime, but it does not reside permanently; if it is not rehearsed it can be forgotten. This memory model has been applied in robot mapping [Dayoub & Duckett, 2008], and in robot control architectures [Barber Castaño, 2000].

4.3. METHOD OVERVIEW

The memory model proposed in [Atkinson & Shiffrin, 1968] has drawn criticism from psychologists and neuroscientists due to its extremely linear representation of the memory process [Baddeley, 2003] [Llinas, 2002]. They argue that the Atkinson and Shiffrin model does not take into account the ability of many people to recall information despite the fact this information has not been rehearsed. This phenomenon is more accentuated in autistic savants. In other words, apparently stimuli inputs can bypass STM to achieve LTM. In addition, this memory model does not consider different levels of memory [Baddeley, 2003] [Llinas, 2002]. From the robotics point of view, it would be useful to take into account levels of memory represented in the strength of the feature information.

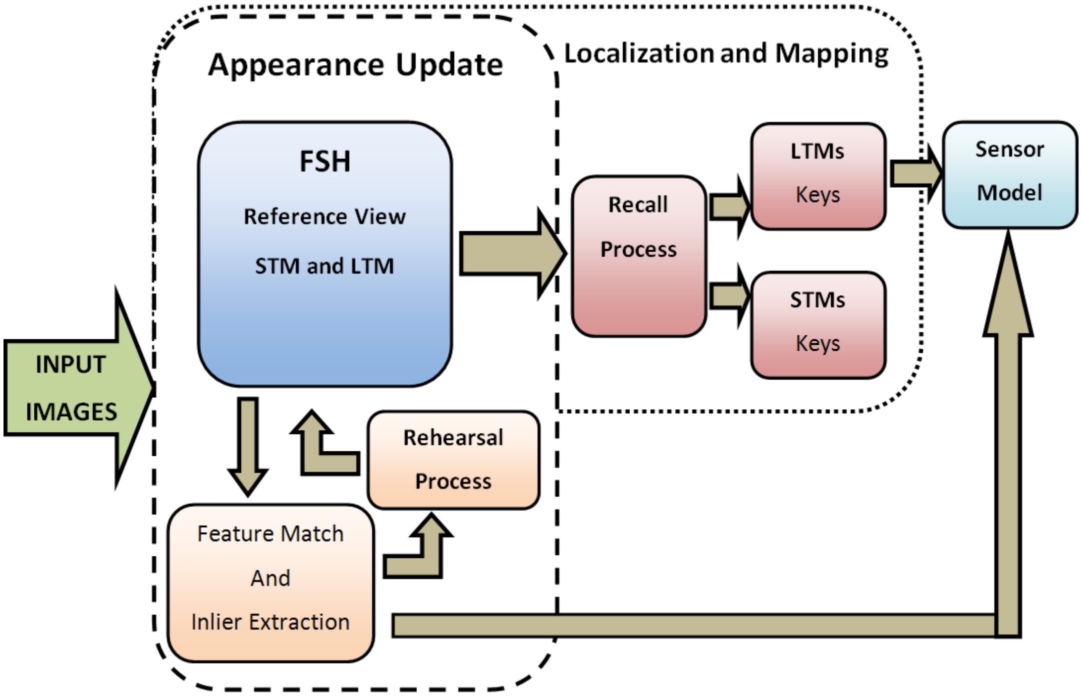


Figure 4.2 The modified human memory model used in this work.

In this work, a map update approach for robot mapping and localization inspired by the Atkinson and Shiffrin memory model is proposed (see Figure 4.2). The reference view is composed of both memories, the STM and LTM. It has two main advantages: first, an input feature can bypass the STM and become a LTM, keeping in mind the feature strength, e.g., the feature uncertainty, the Hessian value in the

SURF descriptor, or the matching distance; second, using the FSH as the reference view, the feature classification (STM or LTM) is not linear since the rehearsal process can take into account the feature strength. The rehearsal process implemented in this work is based on the number of times a feature has been observed, but weighted by a function of the feature strength. In this way, the appearance of the environment represented for the FSH is updated according to the presence or absence of pre-observed features, or the inclusion of new features. When the robot starts the mapping and localization a method to distinguish STM and LTM features is needed. The approach proposed in this work considers this situation because the voting scheme is weighted by a feature importance value.

Once the environment representation is updated, the FSH can be used for mapping and localization. To do so, the recall process distinguishes between STM and LTM features, i.e. differentiating the most stable features (LTM) from the STM features. A feature is defined as an LTM if it has a high value in the FSH; otherwise it is considered an STM feature. This classification has two main advantages: first, it is a straightforward method to deal with temporal occlusions because, when using the voting scheme of the rehearsal process the FSH value of the corresponding feature suffers a relative decrease, or an increase if the feature is re-observed; and second, it is a suitable method to deal with changing environments where illumination changes and pedestrians cause feature appearance or disappearance. In the end, the more stable features will belong to the LTM and will only be used for mapping and localization. The recall process implemented in this work is based on *k-means* clustering and exponential decay, such that the feature mean lifetime is identified.

Once the LTM features are found, they are used crucial SLAM situations as loop-closing. In loop-closing situations, the observed LTM features are used to impose constraints in the loop. As a result, considering the LTM features into the typical SLAM tasks, the map of the environment can be updated accordingly and the robot can deal with long-term operations due the reduced computational cost.

4.4. LOCALIZATION AND MAPPING USING THE FEATURE STABILITY HISTOGRAM

The FSH model can be viewed as a transversal method to the current SLAM solutions, providing them long-term operation and map updating skills. To do so, the SLAM problem formulation and the feature stability degree (provided by the FSH) are considered to re-formulate the SLAM problem as shown in Section 4.4.1. The environmental features are classified as LTM or STM regarding their strength, a detailed description of this process is done in Section 4.4.2. Also, Section 4.4.2 discusses the ever increasing number of features in long-term operations problem, which is common in service robotics. The integration of the FSH model in a SLAM method involves the modification of the data association, update and loop closure processes, which is described in Sections 4.4.3.

4.4.1. Probabilistic Foundations

Figure 2.7 is shown again in Figure 4.3 in order to take into account the modules affected by the integration of the FSH model in a typical SLAM method. Basically, the data association, map update and loop closing processes are involved. The FSH model depends of the data association process to update the histogram of weighted votes. Afterwards, all the re-observed features are estimated, new features are initialized as LTM and only the current re-observed LTM features are used to estimate the robot position. Furthermore, loop closing situations require a reliable landmark shared between the current robot position and the re-visited one to close the loop; LTM features surrounding the re-visited position are used to impose reliable constraints and start the preferred non-linear minimization method.

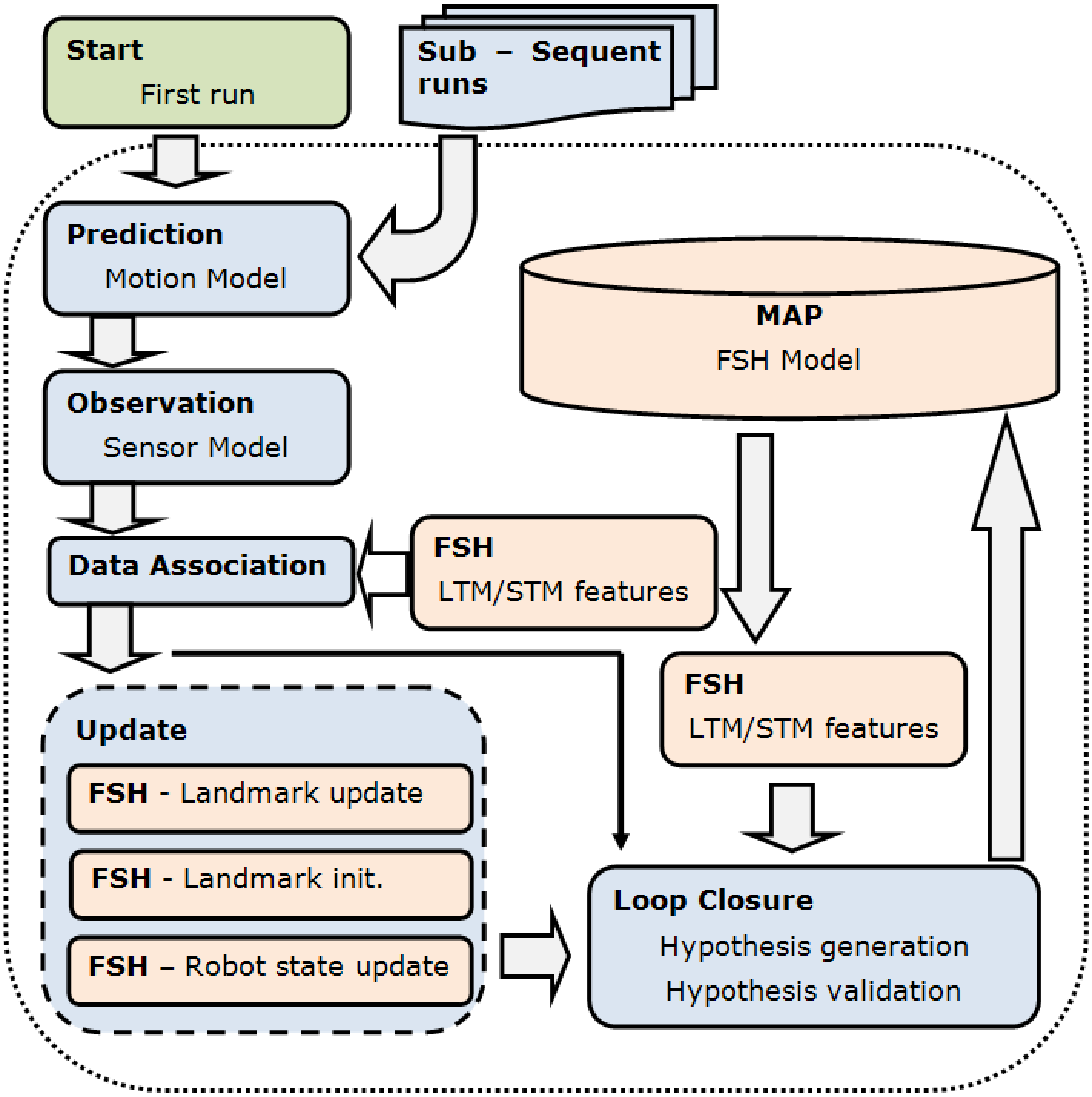


Figure 4.3 Typical estimation loop and the affected processes when the FSH model is integrated in a SLAM method.

To begin with, the probabilistic derivation is based on the SLAM problem formulation, which was modified in order to introduce a hidden variable c_t as depicted in Equation 4.1. This variable indicates if the measurement $\mathbf{z_t}$ belongs to the LTM or not.

$$p(\boldsymbol{x}_t,\boldsymbol{M}_t|\boldsymbol{Z}_t,\boldsymbol{U}_t,\boldsymbol{C}_t) \propto p(\boldsymbol{z}_t,\boldsymbol{c}_t|\boldsymbol{x}_t,\boldsymbol{M}_t,\boldsymbol{Z}_{t-1},\boldsymbol{U}_t,\boldsymbol{C}_{t-1})p(\boldsymbol{x}_t,\boldsymbol{M}_t|\boldsymbol{Z}_{t-1},\boldsymbol{C}_{t-1},\boldsymbol{U}_t) \qquad 4.1$$

Where, $\boldsymbol{x_t}$ is the robot state at time t, $\boldsymbol{M_t}$ is the map built so far, $\boldsymbol{Z_t}$ are the measurements taken up time t, $\boldsymbol{U_t}$ are the control inputs up time t, $\boldsymbol{C_t}$ are the correspondence variable with respect to $\boldsymbol{Z_t}$ indicating the map landmarks that belong to the LTM, and $\boldsymbol{z_t}$ is the last measurement performed. The measurement can be factorized out as Equation 4.2 shows.

$$p(\boldsymbol{z}_t,c_t|\boldsymbol{x}_t,\boldsymbol{M}_t,\boldsymbol{Z}_{t-1},\boldsymbol{U}_t,\boldsymbol{C}_{t-1}) = p(\boldsymbol{z}_t|c_t,\boldsymbol{x}_t,\boldsymbol{M}_t,\boldsymbol{Z}_{t-1},\boldsymbol{U}_t,\boldsymbol{C}_{t-1})p(c_t|\boldsymbol{z}_t,\boldsymbol{x}_t,\boldsymbol{M}_t,\boldsymbol{Z}_{t-1},\boldsymbol{U}_t,\boldsymbol{C}_{t-1})$$

$$p(\boldsymbol{z}_t,c_t|\boldsymbol{x}_t,\boldsymbol{M}_t,\boldsymbol{Z}_{t-1},\boldsymbol{U}_t,\boldsymbol{C}_{t-1}) = p(\boldsymbol{z}_t|c_t,\boldsymbol{x}_t,\boldsymbol{M}_t)p(c_t)$$

$$p(\boldsymbol{z}_t,c_t|\boldsymbol{x}_t,\boldsymbol{M}_t,\boldsymbol{Z}_{t-1},\boldsymbol{U}_t,\boldsymbol{C}_{t-1}) = p(\boldsymbol{z}_t|c_t = LTM,\boldsymbol{x}_t,\boldsymbol{M}_t)p(c_t = LTM) \qquad 4.2$$

Where, the standard Markov assumption was considered, as well as, $\boldsymbol{z_t}$ and c_t are independent given $\boldsymbol{x_t}$ and $\boldsymbol{M_t}$; also, in the last line, the LTM features are only considered for mapping and localization. Furthermore, considering the LTM features only the rightmost term of Equation 4.1 can be further developed as depicted in Equation 4.3.

$$p(\boldsymbol{x_t}, \boldsymbol{M_t} | \boldsymbol{Z_{t-1}} \boldsymbol{C_{t-1}^{LTM}}, \boldsymbol{U_t}) = p(\boldsymbol{x_t} | \boldsymbol{M_t}, \boldsymbol{Z_{t-1}} \boldsymbol{C_{t-1}^{LTM}}, \boldsymbol{U_t}) p(\boldsymbol{M}_t | \boldsymbol{Z_{t-1}} \boldsymbol{C_{t-1}^{LTM}}, \boldsymbol{U_t})$$

$$p(\boldsymbol{x_t}, \boldsymbol{M_t} | \boldsymbol{Z_{t-1}} \boldsymbol{C_{t-1}^{LTM}}, \boldsymbol{U_t}) = \int p(\boldsymbol{x_t} | \boldsymbol{x_{t-1}}, \boldsymbol{M_t}, \boldsymbol{Z_{t-1}} \boldsymbol{C_{t-1}^{LTM}}, \boldsymbol{U_t}) p(\boldsymbol{x_{t-1}} | \boldsymbol{M_t}, \boldsymbol{Z_{t-1}} \boldsymbol{C_{t-1}^{LTM}}, \boldsymbol{U_t}) p(\boldsymbol{M_t} | \boldsymbol{Z_{t-1}} \boldsymbol{C_{t-1}^{LTM}}, \boldsymbol{U_t}) d\boldsymbol{x_{t-1}}$$

$$p(\boldsymbol{x_t}, \boldsymbol{M}_t | \boldsymbol{Z_{t-1}} \boldsymbol{C_{t-1}^{LTM}}, \boldsymbol{U_t}) = \int p(\boldsymbol{x_t} | \boldsymbol{x_{t-1}}, \boldsymbol{u_t}) p(\boldsymbol{x_{t-1}}, \boldsymbol{M_t} | \boldsymbol{Z_{t-1}} \boldsymbol{C_{t-1}^{LTM}}, \boldsymbol{U_{t-1}}) d\boldsymbol{x_{t-1}} \qquad 4.3$$

Last line of Equation 4.3 depicts the well known SLAM prediction step, considering only the LTM correspondence variables. The final filter equation is obtained putting Equations 4.2 and 4.3 into Equation 4.1. This is shown in Equation 4.4.

$$p(\boldsymbol{x_t}, \boldsymbol{M_t} | \boldsymbol{Z_t}, \boldsymbol{U_t}, \boldsymbol{C_t^{LTM}}) \propto p(c_t = LTM) p(\boldsymbol{z_t} | c_t = LTM, \boldsymbol{x_t}, \boldsymbol{M_t}) \int p(\boldsymbol{x_t} | \boldsymbol{x_{t-1}}, \boldsymbol{u_t}) p(\boldsymbol{x_{t-1}}, \boldsymbol{M_t} | \boldsymbol{Z_{t-1}} \boldsymbol{C_{t-1}^{LTM}}, \boldsymbol{U_{t-1}}) d\boldsymbol{x_{t-1}} \qquad 4.4$$

Equation 4.4 shows the SLAM posterior weighted by the term *p(*c_t *= LTM)*, which is the likelihood that a measurement $\boldsymbol{z_t}$ corresponds to a LTM feature. Exploiting the fact that c_t is conditional independent of $\boldsymbol{z_t}$ given $\boldsymbol{x_t}$ and $\boldsymbol{M_t}$, *p(*c_t *= LTM)* is extracted from the normalized FSH and regarded as a probability distribution. The FSH values are related with the information content of a landmark, which can be computed by the sum of the reciprocals of the main diagonal elements of the covariance matrix as suggested by [Dissanayake et al., 2000], and depicted in Equation 4.5.

$$fsh(\boldsymbol{z_{t,i}}, \boldsymbol{\Sigma_{z_t}^i}) = \sum_{n=1}^{R} \frac{1}{\sigma_{nn}^2} \qquad 4.5$$

Where, $\boldsymbol{z_{t,i}}$ is the *i-th* feature of the measurement performed at time *t*, *R* is the rank of $\boldsymbol{z_{t,i}}$ and σ_{nn}^2 is the *n-th* value of the covariance matrix $\boldsymbol{\Sigma_{z_t}^i}$. However, the feature strength not always can be expressed in terms of the covariance matrix. In this case, alternative options may be the Hessian value in the SURF descriptor, or the matching distance when epipolar geometry is used. By doing this, the FSH values can be updated using Equation 4.6 [Bacca et al., 2011].

$$fsh(\boldsymbol{z_{t,i}}, m) = e^{-\frac{m^2}{\sigma_m^2}} \qquad 4.6$$

Where, *m* is the matching distance between each corresponding feature and σ^2_m is the variance of the matching distances. Equations 4.5 and 4.6 are part of the rehearsal process depicted in Figure 4.2, which is in charge of rating the map landmarks when updating the FSH values.

The aim of the probabilistic derivation done in Equations 4.1 to 4.5 is to show the FSH model proposed can be implemented on different SLAM solutions. The rightmost term of Equation 4.4 is basically the SLAM problem formulation considering the LTM features as observations in the filtering process. This means that parametric (EKF-based) and non-parametric filters (e.g. particle filters) are able to implement the FSH model.

4.4.2. *LTM / STM Feature Classification and STM Features Removal*

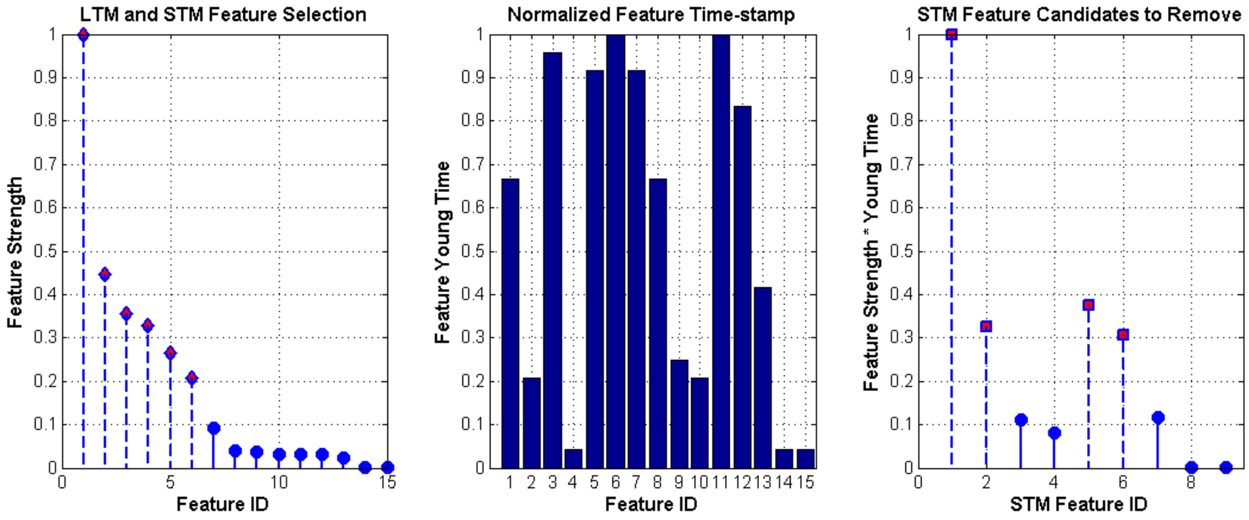
The FSH model discussed up to now considers two types of features, namely LTM or STM. Taking into account the FSH model depicted in Figure 4.2 and the human memory model proposed by [Atkinson & Shiffrin, 1968], it is important to define a discrimination method to classify the environmental features as LTM or STM. As a result, useless STM features can be removed in order to increase the SLAM algorithm scalability.

According to how the FSH is built, a good way to differentiate LTM from STM features is using a threshold such that FSH values greater than the threshold are considered LTM features, and those less than the threshold are STM features. But, the FSH values change continuously, so an arbitrary threshold (e.g. 0.5) could cause some problems: firstly, very few LTM features could be obtained which is not good for the mapping and localization process, because the robot position uncertainty increases and causes lack of representativeness of the environment; last, a high number of LTM features could increase the data association complexity and decrease scalability. Within the FSH model, the map features have an additional parameter given by the feature strength, which can be computed by Equations 4.5 or 4.6. Taking advantage of this parameter and how it is used to build the FSH, the LTM/STM feature classification problem can be solved using *k-means* [Lloyd, 1982]. The clustering problem has low dimensionality (rank-2), and the number of data points is low. The latter can be achieved using many well known SLAM techniques as: topological mapping [Thrun et al., 2005] or sub-mapping methods [Estrada et al., 2005] [Aulinas et al., 2010].

Then, given a set of observations (***sf***$_1$, ***sf***$_2$, …, ***sf***$_N$) where each observation is the feature strength, the aim of the clustering problem is to classify the *N* features into LTM or STM sets, ***S*** = {S$_{\text{LTM}}$, S$_{\text{STM}}$}. The within-cluster sum of squares cost function yields as depicted in Equation 4.7.

$$min_S \sum_{j=1}^{K} \sum_{i=1}^{N} \left\| \boldsymbol{sf}_i - \boldsymbol{\mu}_j \right\|^2 \qquad 4.7$$

Where, *K* is the number of sets which in this case is 2; and μ_j is the mean of the data points belonging to the set S_j (*j* stands for LTM or STM). Once the minimization of the Equation 4.7 is done, all those features which belong to the highest mean are considered as LTM.



a) *b)* *c)*

Figure 4.4 a) LTM and STM features selection using k-means. b) Normalized feature time stamp with respect to the current viewing step. c) STM feature candidates to be removed (circle-shaped).

Figure 4.4a shows a typical normalized FSH where the data points depicted using diamonds correspond to the LTM features, and those points drawn with circles correspond to the STM features. This classification was done applying the Equation 4.7 to the FSH values, which is directly related with the feature strengths. Algorithm 4.1 summarizes the basic steps to classify the environmental features as STM or LTM. A well known drawback of *k-means* is that the number of clusters has to be known in advance; however as a rule of thumb the number of sets can be estimated as Equation 4.8 shows [Mardia et al., 1980] (line 2).

$$k \approx \sqrt{\frac{N}{2}} \tag{4.8}$$

Where N is the number of data points. In this way, if there are not enough data points, the LTM/STM classification is performed using exponential fitting in order to estimate the mean lifetime as threshold value to distinguish LTM from STM features (lines 4 to 6). Otherwise, the threshold value is obtained from the set centroid locations computed by the *k-means* algorithm (line 9). This threshold value is calculated using the maximum value of the centroid locations and its corresponding data points (line 10).

Algorithm 4.1 LTM / STM features classification.

```
1.   FSHvalues = getFSHvalues();
2.   numClusters = getNumClustersEstimation(FSHvalues);
3.   if numClusters == 1
4.           % LTM/STM classification using exponential decay.
5.           [N0 λ] = getExponentialDecayParameters(FSHvalues);
6.           LTM_STM_threshold = getMeanLifeTime(N0, λ);
7.   else
8.           % LTM/STM classification using k-means.
9.           [setCentroid setIndexes] = doKMeans(FSHvalues, numClusters);
10.          LTM_STM_threshold = getThresholdFromMaxCentroid(setCentroid, setIndexes,
11.          FSHvalues);
12.  end
13.
14.  LTM_FeaturesIndex = getLTMfeatures(LTM_STM_threshold, FSHvalues);
15.  STM_FeaturesIndex = getSTMfeatures(LTM_STM_threshold, FSHvalues);
```

Observing Figure 4.4a, particularly the STM features, their strengths are comparatively low with respect to the LTM features strength. Basically, this is due to two main reasons: the features are recently incorporated to the map and their strength is not high; or, the features are old and they have not been re-observed inducing a relative decreasing of their strength. Taking advantage of this, a STM feature removal algorithm can be proposed considering the following requirements:

1. Limiting the ever growing number of STM features.
2. Preserving the newest features despite the fact that they have low strengths.
3. Removing the oldest and weak features.

To do so, the FSH model also considers the time stamp at which the feature was first measured, and it is updated each time the feature is re-observed. Figure 4.4b shows the normalized time stamp corresponding to those features in Figure 4.4a. The higher the values, the younger the feature is. Observing the normalized time stamp values for features 7 to 15, the weakness of features 11 to 14 is not enough justification to be removed since they are new features. On the other hand, features 14 or 15 could be candidates to be removed, since they have low strengths and they are the oldest. Formally, assuming that the normalized feature time stamp and the feature strength stored in the FSH are independent given the robot position, the likelihood model for the STM feature removal is given in Equation 4.9.

$$p(sf_i^{STM}, tf_i^{STM}|\boldsymbol{x_t}) = p(sf_i^{STM}|\boldsymbol{x_t})p(tf_i^{STM}|\boldsymbol{x_t}) \tag{4.9}$$

Where, $\boldsymbol{x_t}$ is the robot position, sf_i^{STM} is the *i-th* STM feature strength and tf_i^{STM} is the *i-th* STM feature time stamp. Figure 4.4c shows the resulting likelihood model given by Equation 4.9 for the typical FSH model and time stamps of Figure 4.4a and Figure 4.4b. Here, the classification problem is to group the STM features into two sets, those STM features to be removed and those not. To this end, the *k-means* algorithm can be used as depicted in Equation 4.7 doing the appropriate changes in the variables. As a result, the circle shaped points in Figure 4.4.c show the STM features to be removed.

Algorithm 4.2 STM Feature removal.

```
FSHvalues = getFSHvalues();
STMfeatureStrength = getSTMstrengths(FSHvalues, STM_FeatureIndex);
FeatureTimeStamps = getFSHtimeStamps();
STMfeatureTimeStamps = getSTMtimeStamps(FeatureTimeStamps, STM_FeatureIndex);
STMlikelihood = doComputeSTMLikelihood(FeatureTimeStamps , STMfeatureTimeStamps);
numClusters = getNumClustersEstimation(STMlikelihood);
if numClusters > 1
        % STM feature using k-means.
        [setCentroids setIndexes] = doKMeans(STMlikelihood, numClusters);
        STMremovalThreshold = getThresholdFromMaxCentroid(setCentroids, setIndexes, STMlikelihood);
        % Delete features.
        removeSTMfeatures(STM_FeatureIndex, STMremovalThreshold, STMlikelihood);
end
```

Algorithm 4.2 summarizes the STM feature removal. In this case, the exponential fitting is not performed when the number of data points is not enough, since feature deleting has to be carried out once enough evidence is collected.

4.4.3. *Map Building*

The previous sections have described the essential characteristics of the FSH model such as: its structure based on the human memory model, the probabilistic foundation to be adapted to the current SLAM solutions, the rehearsal procedure to promote features from the STM to the LTM, the STM / LTM feature classification method and the STM feature removal. However, considering Figure 4.3 additional insights have to be described in order to carry out the FSH model implementation into any SLAM solution.

To start with, the following assumptions have to be considered:

1. The features can be decomposed into LTM ($\boldsymbol{z^{LTM}}$) and STM ($\boldsymbol{z^{STM}}$), such that they can be conditionally independent given the robot position ($\boldsymbol{x_t}$). This is important since Equation 4.4 was derived base on this assumption.

2. The posterior over the map and robot poses are estimated regarding the LTM features ($\boldsymbol{z^{LTM}}$), however in order to update the FSH model the STM features ($\boldsymbol{z^{STM}}$) are considered in the data association process.
3. The FSH model is computed once the SLAM solution has finished the update stage.
4. The LTM/STM feature classification is performed at each time step, however the STM feature removal is performed each time the corresponding area is re-visited. The latter includes: loop closure situations or further SLAM runs.

According with Figure 4.3, the FSH model is involved in the data association, map and state update and loop closure detection processes. However, before explaining the influence of the FSH model in these stages, the FSH model computation pipeline is described first. Following with the notation used in Table 2.1, Table 4.1 lists the main variables that will be used and their meaning.

Table 4.1 Notation.

Variable	Description
x_{t-1}, $\hat{x}_t$, x_{t+1}	Previous, estimated and corrected robot pose.
Px_{t-1}, $\widehat{Px}_t$, Px_{t+1}	Previous, estimated and corrected robot pose covariance matrix.
u_t	Current motion command.
R_t, Q_t	Process and observation uncertainty matrix.
M_t, M_{t+1}	Current and updated map
FSH_t, FSH_{t+1}	Current and updated FSH values.
$Zind_t^{LTM}$, $Zind_t^{STM}$	Current LTM and STM feature indexes.
$Zind_{t+1}^{LTM}$, $Zind_{t+1}^{STM}$	Updated LTM and STM feature indexes.
H_t	Current data association vector.
Z_t, Z_{t+1}	Estimated and corrected re-observed set of features.
Z_t^{LTM}, Z_t^{STM}	Current LTM and STM landmarks.
$z_{G,t+1}^i$	Corrected *i-th* feature, *G* stands for LTM or STM.
$Pz_{G,t+1}^i$	Corrected *i-th* feature covariance matrix, *G* stands for LTM or STM.
$Sz_{G,t+1}^i$	*i-th* feature strength, *G* stands for LTM or STM.
PS_{t-1}, PS_t	Previous and computed FSH model for the laser scans.
PS_{t-1}, PS_t	Previous and computed FSH model for the laser scans.
$Z_{0:t}^{LTM}$, $PS_{0:t}^{LTM}$, $A_{0:t}^{LTM}$	LTM landmarks, FSH model for the laser scans and appearance-based image descriptors from *t=0* to *t=t*.

Algorithm 4.3 shows how the FSH model can be integrated as part of a SLAM solution, and it summarizes the description of the FSH model done so far. The SLAM prediction and measurement stages are not affected by the FSH model; however, the data association process keeps in mind the current LTM and STM feature indexes in order to perform its work hierarchically, i.e. the LTM features are associated first, and then the STM features. Data association of the STM features is important because depending on it the FSH values corresponding to the STM features can be updated.

Other SLAM stage affected is the filter update process. Here, the LTM features are only considered to correct the robot pose and covariance. On the other hand, the re-observed STM features are updated in order to estimate their covariance matrix and in this way compute their strength in further time steps. Once the SLAM filter finishes, the FSH model is computed using the data association vector, the current FSH values and the updated covariance matrices of the re-observed features. Afterwards, the LTM/STM feature classification take place as well as the STM features removal process.

Algorithm 4.3 SLAM and FSH model computation.

```
while operating
    % SLAM prediction.
    [$\hat{x}_t$   $\widehat{Px}_t$] = doSLAMprediction($x_{t-1}$, $u_t$, $Px_{t-1}$, $R_t$);

    % SLAM measurement.
    $Z_t$  = doMeasurement($\hat{x}_t$, $\widehat{Px}_t$, $M_t$, $Q_t$);
    $s_t$  = getCurrentScan();

    % Current FSH model.
    $FSH_t$  = getFSHvalues();

    % SLAM data association.
    $H_t$  = getDataAssociationVector($Z_t$, $M_t$, $Zind_t^{STM}$, $Zind_t^{LTM}$);

    % SLAM update.
    [$x_{t+1}$, $Px_{t+1}$, $Z_{t+1}$, $M_{t+1}$] = doSLAMupdate($\hat{x}_t$, $\widehat{Px}_t$, $M_t$, $Z_t$, $H_t$, $FSH_t$, $Zind_t^{STM}$, $Zind_t^{LTM}$);

    % FSH model computation – Rehearsal process.
    for $z_{G,t+1}^i$ = $Z_{t+1}$ ( $H_t$ )
        $Sz_{G,t+1}^i$  = getFeatureStrength($Pz_{G,t+1}^i$);
        $FSH_{t+1}$  = updateFSHvalues($Sz_{G,t+1}^i$, $FSH_t$);
    end
    % FSH model computation – LTM/STM feature classification.
    [$Zind_{t+1}^{LTM}$, $Zind_{t+1}^{STM}$] = doLTM_STM_Classification($FSH_{t+1}$);
    % FSH model computation – STM pruning
    doSTMfeaturePruning($FSH_{t+1}$, $Zind_{t+1}^{STM}$);
    % FSH model computation – LRF readings.
    $PS_t$ = doFSHoverLRFreadings($PS_{t-1}$, $s_t$);

    % Loop-closure detection.
    LC_Alert = doLoopClosureDetection($x_{t+1}$, $Px_{t+1}$, $M_{t+1}$, $H_t$, $Zind_{t+1}^{STM}$, $Zind_{t+1}^{LTM}$, $PS_t$);
    if (LC_Alert)
        doConstraintLoop($x_{t+1}$, $Px_{t+1}$, $M_{t+1}$, $H_t$, $Zind_{t+1}^{LTM}$);
```

```
34.         end
35.     end
```

LRF sensors are popular in the robotics community. The FSH model described so far can be applied to any kind of observation. In this work, the laser scan is also processed by the FSH model (line 28). By doing this, the more stable local environmental structure can be obtained, filtering out pedestrians and moving objects. Formally, given a sequence of LRF readings $\boldsymbol{S_t} = \{\boldsymbol{s_1} \quad \cdots \quad s_N\}$, where *N* is the total number of LRF scans and $\boldsymbol{s_j}$ corresponds to a *m x N* matrix being *m* the rank of the data points, a set of votes vs_j^i can be computed for each *i-th* point in the *j-th* laser scan.

Using the previous filtered robot positions the $\boldsymbol{s_{t-1}}$ LRF readings are sequentially registered, yielding a local map patch $\boldsymbol{PS_{t-1}}$. At each step, the $\boldsymbol{PS_{t-1}}$ is aligned with the current LRF reading $\boldsymbol{s_t}$ using the filtered robot position $\boldsymbol{x_{t+1}}$ and the set of votes for the FSH model is computed using the nearest-neighbor (NN) approach. The vote of a data point can is defined as depicted in Equation 4.10.

$$vs_j^i = \begin{cases} 1 & if \left\| \boldsymbol{s_t^i} - \boldsymbol{PS_{t-1}^k} \right\|^2 < LRFresThreshold \\ 0 & Otherwise \end{cases} \qquad 4.10$$

Where, $\boldsymbol{s_t^i}$ is the *i-th* point in the current laser scan, $\boldsymbol{PS_{t-1}^k}$ is the *k-th* point in the previous map patch, and *LRFresThreshold* depends on the LRF resolution. In this work, the URG-04LX is used and its range resolution is 0.04m [Hokuyo, Automatic Co., 2009], then the *LRFresThreshold* value was set to 0.04m.

Detecting loop closure situations is a challenging task. Generally, loop closing techniques take advantage of the topological representation of the environment [Thrun et al., 2005], in this way the computation time, the uncertainty estimation and scalability are improved. In this work, a global stochastic map representation is stored using the relative locations between nodes. A new node is started when the number of features in the current node reaches a maximum, or no matchings were found by the data association stage.

There are two ways in which the FSH model is used for loop closure detection: firstly, the map patches $\boldsymbol{PS_j}$ and the current scan $\boldsymbol{s_t}$ are used to get a similarity measure using the Hausdorff fraction [Rucklidge, 1996], which is a metric used to measure the distance between two sets of points; secondly, an observed LTM feature is used to close the loop, yielding a set of constraints which include the robot positions, the LTM feature and the base position of the node which closes the loop. The similarity between the map patches and the current scan, and the amount of overlapping features are considered to issue a loop closing alert (line 31). In this work, an overlapping of 60% [Estrada et al., 2005] [Aulinas et al., 2010] (both map patches and landmarks) is used to issue a loop closing alert.

Finally, in line 33 the detected loop closure is processed using the graph representation of the robot poses, their uncertainties, and the additional constraint over the observed LTM feature. The non-linear optimization is done using TORO [Grisetti et al., 2008], which has been adapted to work in incremental mode and through Matlab MEX files.

4.5. EXPERIMENTAL RESULTS

This section presents a set of experimental results with the aim to test the operating principle of the FSH model. The experiments conducted to test the FSH model were classified in two groups: firstly, a static LRF-based experiment in order to show how the FSH model works with range data, and how the appearance representation of the environment is updated in presence of dynamic objects. In addition, a static vision-based experiment is performed in order to observe the image similarity behavior with and without the environment appearance representation update. The aim of the static vision-based experiment is to show how dynamic environments affect similarity measures, and how the FSH model properly updates the appearance representation of the environment.

Secondly, the FSH model is used to build a vision-based topological map [Bacca et al., 2011]. Here, the FSH model is compared with the method proposed by [Dayoub et al., 2011] and the Bag-of-Words technique [Sivic & Zisserman, 2003] that nowadays is becoming popular for appearance-based mapping and localization [Bazeille & Filliat, 2010]. The localization and topological map building method used was a Bayesian-based simulation framework, where the motion model and sensor model are defined by Equations 4.11 and 4.12. To define the motion model the temporary coherence of the node position estimation was enforced, and it was assumed that transitions between closer places are more likely than transitions between more distant node locations.

$$p(x_t|x_{t-1}) = \gamma e^{-\frac{\|x_t - x_{t-1}\|}{\sigma_x{}^2}} \qquad 4.11$$

$$p(z_t|x_t) = \delta e^{-\frac{sim\,(z_t, Z_{LTM})}{\sigma_z{}^2}} \qquad 4.12$$

Where, x_t is the current node, γ is a normalization constant, $||x_t - x_{t-1}||$ is the distance between two nodes in the topological map, σ^2_x is the variance of the distances on the map, z_t is the current set of image descriptors, δ is a normalization constant, $sim(z_t, Z_{LTM})$ is the similarity measure defined by matching candidates and feature correspondences ratio, Z_{LTM} is a set of LTM features stored in the topological map, and σ^2_z is the variance of this measure.

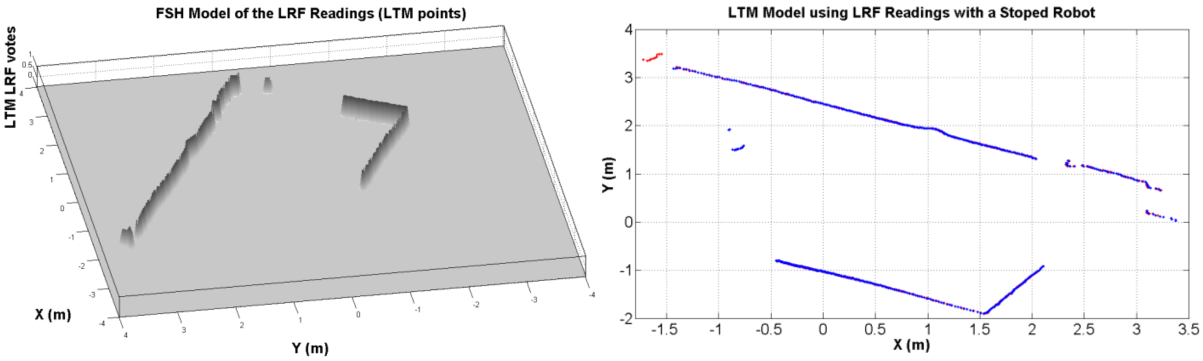
In the results reported here, the method proposed by [Dayoub et al., 2011] was implemented with 4 and 5 stages in STM and LTM finite-state-machine respectively. The feature extraction was done using the SURF algorithm over the original omnidirectional image; these features were not computed over the unwrapped panoramic image as in [Dayoub et al., 2011]. Bag-of-Words methods based their environmental representation on a set of unordered features (the visual words) taken from a dictionary. The dictionary is built using a clustering technique, commonly *k-means*, and then image classification is based on the occurrence of the visual words in an image to infer its class. The dictionary is built beforehand in an offline process. The matching process is based on a Nearest-Neighbor (NN) search among the distance separating the corresponding visual words. The Bag-of-Words toolbox used was the Caltech large scale image [Aly et al., 2011]. The FSH model was implemented to use SURF features computed on the original omnidirectional image. The clustering process involves 80 different classes corresponding to each node in the topological map. Each time the map was updated, the dictionary and clustering process were generated. The motion model assumed was defined by Equation 4.11, and the sensor likelihood model was the *term-frequency-inverse document frequency (tf – idf)* weighting depicted in Equation 4.13.

$$tf - idf = \frac{n_{wi}}{n_i} log \frac{N}{n_w} \qquad 4.13$$

where, n_{wi} is the number of occurrences of word w in an image I_i, n_i is the total number of words in I_i, n_w is the number of images containing word w, and N is the total number of images seen so far.

4.5.1. *Static LRF-based and Vision-based Experiment*

The static LRF-based experiment took place in a crowded corridor in the building PIV of the University of Girona. In this experiment the Pioneer 3DX mobile robot as depicted in Figure 2.5a was used. Basically, the robot was acquiring data over 30 minutes, the robot surroundings were composed by static objects (walls, door and one window); during the data acquisition the appearance of the environment was changed along with many pedestrians were passing by. Figure 4.5 shows the initial appearance of the environment, where Figure 4.5a shows the FSH model of the range data (the darker a point is, the more stable it is) and Figure 4.5b shows the LTM laser readings. In the following figures the complete laser readings, as well as the LTM laser readings are shown in order to observe how the approach discussed in this chapter filter out spurious readings.

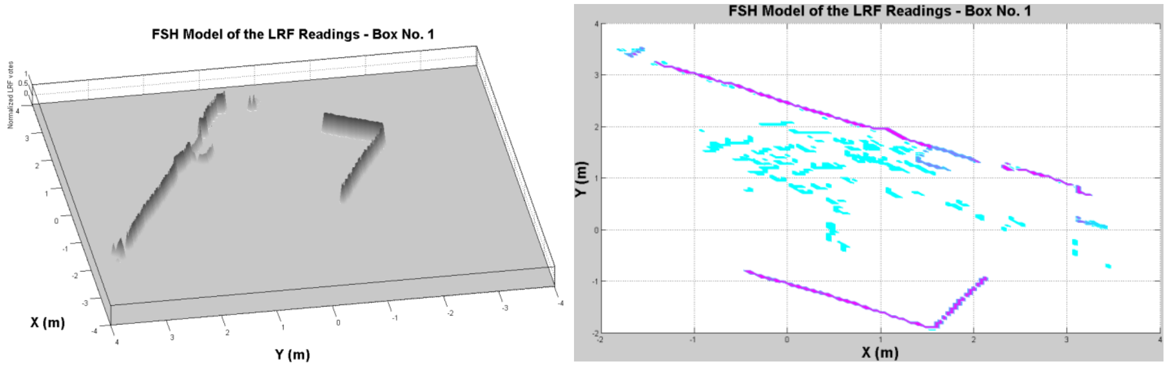


a)

b)

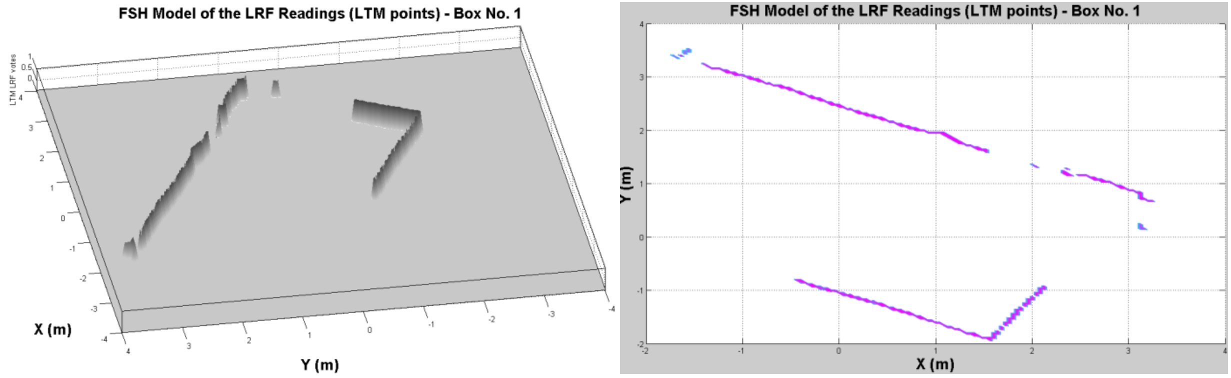
Figure 4.5 Initial appearance of the environment. a) FSH model of the range data. b) LTM laser readings on the plane XY.

Using Algorithm 4.1 and Equation 4.10, the laser readings are continuously classified as LTM or STM. Figure 4.6a and 4.6b show the time in which the box No. 1 is placed modifying the appearance of the environment. The FSH model starts assigning votes to those laser readings belonging to the box No. 1, however they are not immediately classified as LTM. This can be observed in Figure 4.6c and 4.6d where the appearance of the environment slightly changes. From Figure 4.6b and 4.6d it can be observed that the range data corresponding to pedestrians (brighter blobs in Figure 4.6b) are filter out in the LTM laser readings.



a)

b)

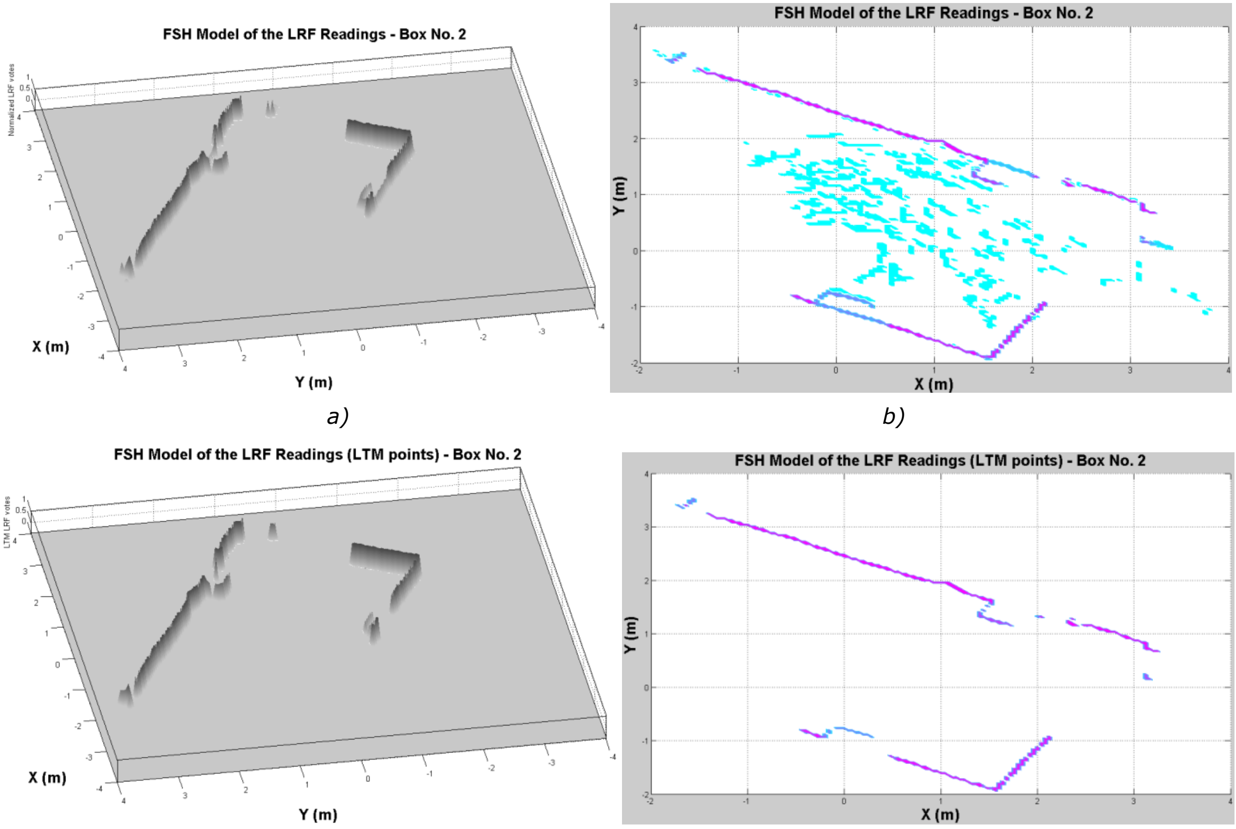


c) d)

Figure 4.6 Modifying the environment when the box No. 1 is placed. a) FSH model at the moment of placing the box No. 1. b) FSH model projected on the XY plane. c) LTM laser readings. d) LTM laser readings on the XY plane.

Figure 4.7 shows the time in which the box No. 2 is placed. The scene structure is changed again, and after some time the LTM laser readings reveal this change once the laser readings have gained enough votes to be classified as LTM. Comparing Figure 4.7b and 4.7d, despite the fact that the corridor selected is very crowded, those spurious laser readings do not appear in the LTM laser points.

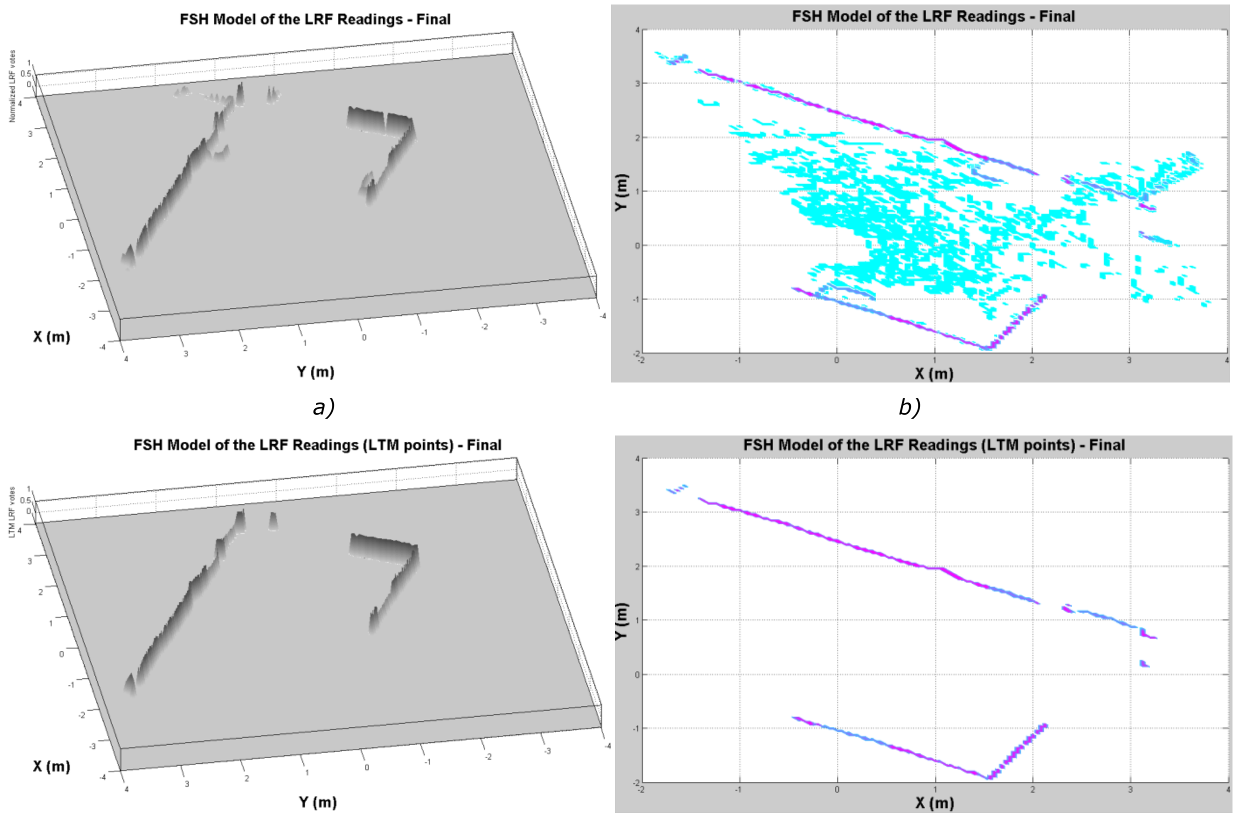
Afterwards, the two boxes are removed from the environment as can be observed in Figure 4.8a and 4.8c where their corresponding FSH votes start decreasing. At the end, the environment is restored to its initial state, however despite the fact that its appearance changed over the time, only those changes which were considered by the FSH model as stable were shown in the LTM laser readings.



a) b)

c) d)

Figure 4.7 Modifying the environment when the box No. 2 is placed. a) FSH model at the moment of placing the box No. 2. b) FSH model projected on the XY plane. c) LTM laser readings. d) LTM laser readings on the XY plane.

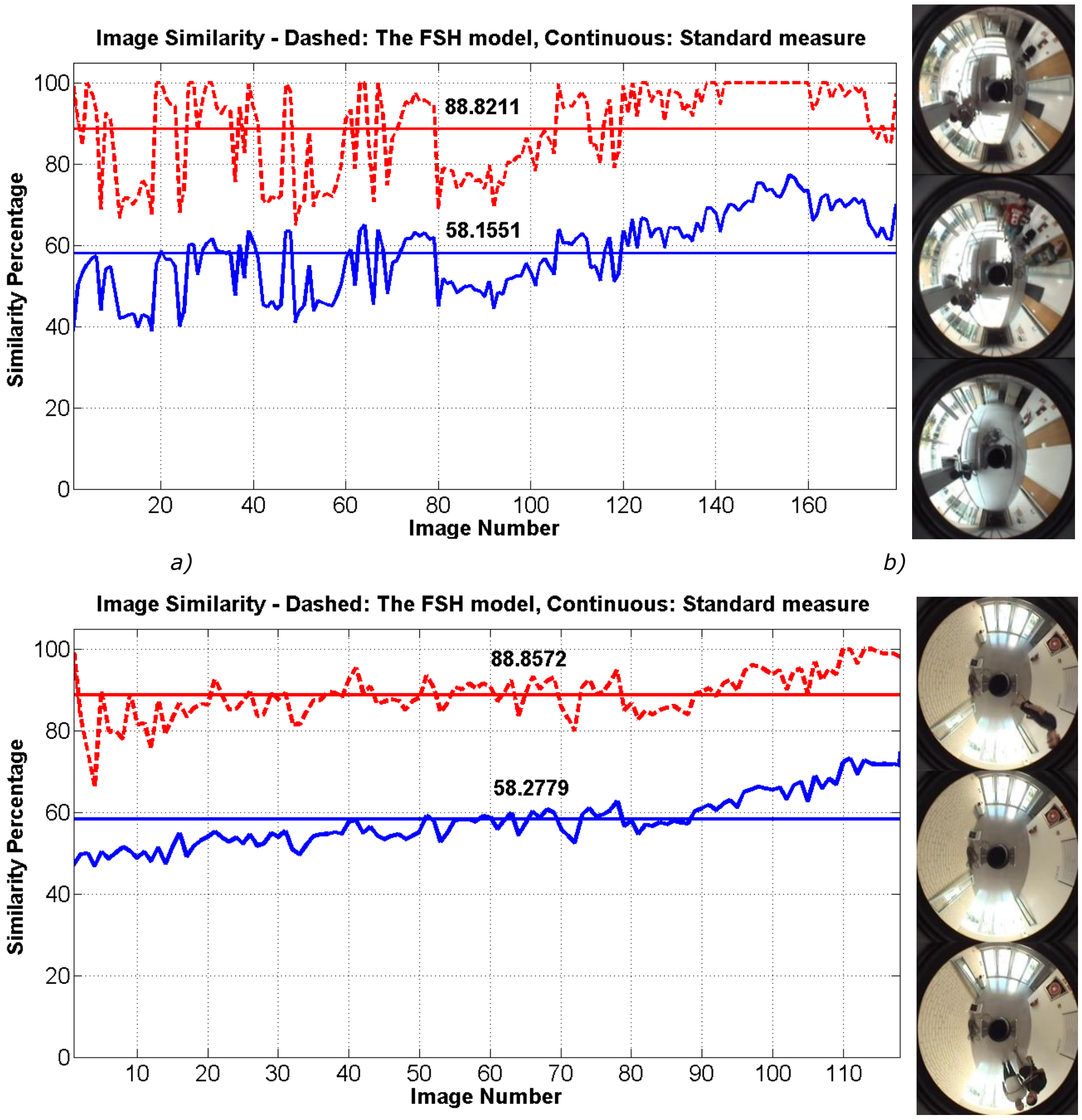


a) *b)*

c) *d)*

Figure 4.8 Final appearance of the environment. a) FSH model of the range data. b) LTM laser readings on the plane XY. c) LTM laser readings. d) LTM laser readings on the XY plane.

On the other hand, the static vision-based experiment took place in another corridor close to a big window ensuring real world conditions due to illumination changes. Figure 4.9a compares the similarity percentage of the captured images. The 180 images were acquired over five days. Figure 4.9a shows both similarity measures: the dashed curve was made using the FSH model, whereas the continuous curve was created without updating the environment appearance. Figure 4.9b shows three examples of the typical omnidirectional images obtained at this place, where one can observe the changes in illumination and occlusions due to passers-by. The image similarity means were 88.82% and 58.15% for the FSH model and without the appearance update, respectively. Dynamic environments as shown in Figure 4.9b cause low similarity measures when the representation of the environment is not updated accordingly, but in the case of an LTM-based similarity measure, this effect is reduced because most LTM features remain and a good representation of the environment is maintained. A second place was also selected. It is close to a cafeteria area. In this place a second static vision-based experiment, whose results are shown in Figure 4.9c and 4.9d took place. Again, the FSH model performed better than the classical approach without appearance update.



a) b)

c) d)

Figure 4.9 Static image similarity test. a) Similarity measure of place No. 1. b) Typical omnidirectional images of place No. 1. c) Similarity measure of place No. 2. b) Typical omnidirectional images of place No. 2.

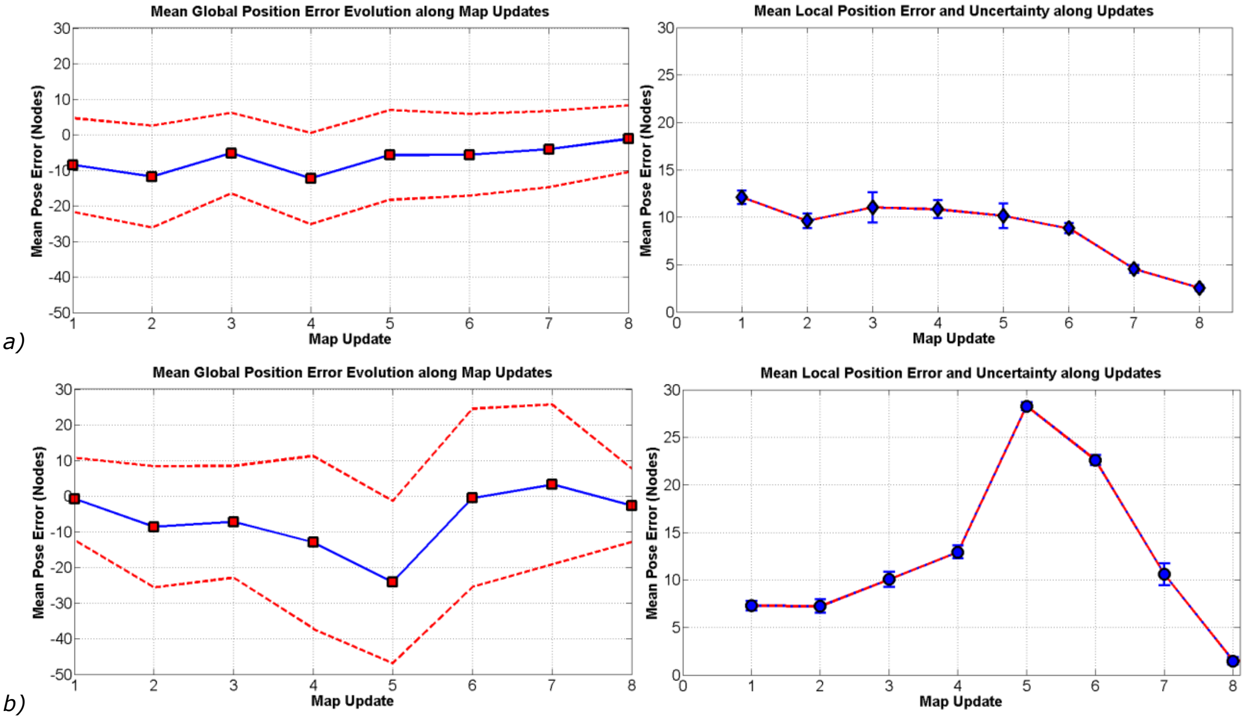
4.5.2. *Appearance-based and Topological Mapping Experiment*

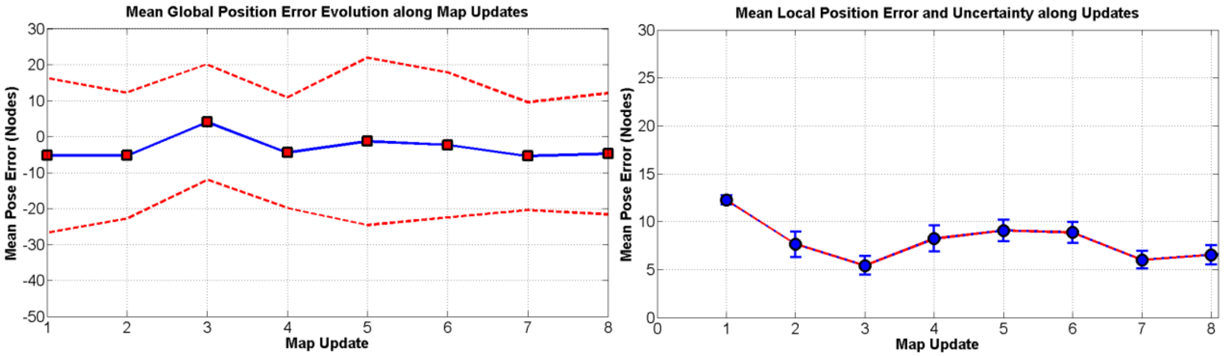
As described at the beginning of this section, comparative global and local topological localization was performed, including the approach proposed by [Dayoub et al., 2011], the Bag-of-Words method and the FSH model. Four type of test were done: the first one used the original set of test images, without noise or artificial occlusion; in the second a Gaussian noise with μ = *0* and σ = *0.15* was added to the current image, but without artificial occlusion; and in the third and fourth tests a Gaussian noise with μ = *0* and σ = *0.15* and artificial occlusion was added by randomly removing 25% and 50% of the current image features, respectively.

To evaluate the localization performance, 100 random image sequences were generated from the test dataset for each experiment. In both global and local localization, the estimated location was selected using the winner-takes-all approach. Since we have the real node that each image belongs to, the mean position error in the topological space can be obtained using the 100 random image sequences. Successful position estimation means that the maximum value of the posterior Gaussian belief is within ±1.25 nodes around the real node location in the map. At each random image sequence, global localization was evaluated using the first image of the sequence. In this way it is ensured that no previous knowledge about the location was available, since this is the first localization attempt for each image sequence. The remaining images in the sequence were used to evaluate local localization, since in this case the localization algorithm deals with tracking the robot motion along subsequent poses.

4.5.2.1. Global and local localization without noise or artificial occlusion

Figure 4.10 shows the mean position error of global and local localization along the map updates and its uncertainty bounds (3σ). Figure 4.10a, 4.10b and 4.10c correspond to the results obtained using the FSH model, the method proposed in [Dayoub et al., 2011] and the Bag-of-Words approach implemented using the Caltech toolbox [Aly et al., 2011] respectively. The left and right side of each figure shows the mean position error for global and local localization, respectively. Because the global localization error was measured using the first image in the random sequence, wide uncertainty bounds are expected to be present. It is also expected that the mean pose error will approach zero as the map updates increase. The experimental results for global localization of Figure 4.10 show that the FSH model, which uses only visual information to figure out where the robot is placed in the topological map, presents a lower position error uncertainty than the others. In the FSH model, the mean error position for global localization tends to approach zero as the map updates increase. In the method proposed by [Dayoub et al., 2011] the effect of the number of states in the finite-state-machine for the STM features (4) can be seen, because when the map update 5 was presented to the system, it started to decrease the mean position error and was approaching zero, which is not the case for the FSH model. The Bag-of-Words method holds a more or less constant global position error, but its uncertainty bounds are greater than the FSH model.



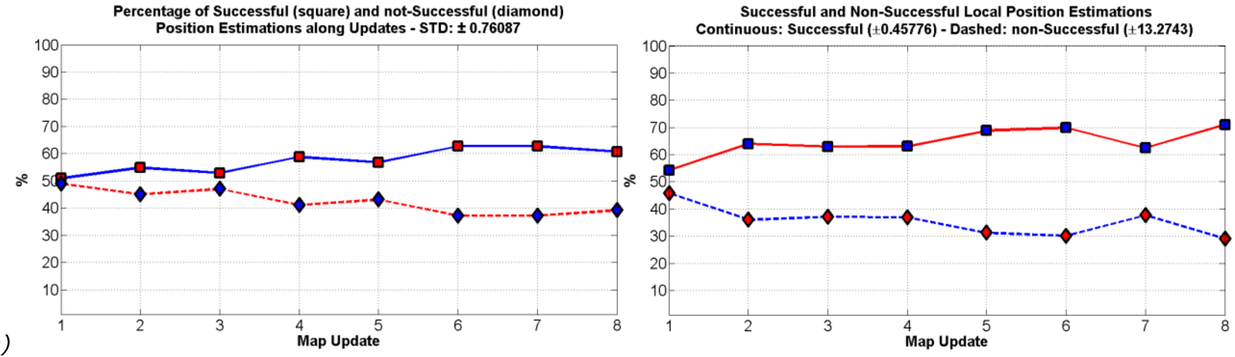


c)

Figure 4.10 Mean position error of global and local localization along map updates without noise or artificial occlusion. a) The FSH model. b) The method proposed in [Dayoub et al., 2011]. c) The Bag-of-Words method.

The mean error for local localization behaves in a similar way. As the map updates increase, the mean error tends to decrease. However, in the FSH model the effect of the environmental appearance update is more evident than in the other approaches. The method proposed by [Dayoub et al., 2011] is highly affected by the number of states in the STM finite-state-machine; and in the Bag-of-Words approach there is an increase of the mean position error between map updates 4 and 6, which belong to spring and summer according to our dataset. This caused big changes in illumination and bright spots which were considered as features but without meaningful environmental appearance information. It is worth noting that we are dealing with real world images, where natural changes in illumination, walking people and occlusions influence the feature extraction and cause increases and decreases of the mean position error for the three methods tested. However, Figure 4.10 shows that the FHS model deals with these situations better.

Figure 4.11 shows the percentage of successful and non-successful global and local position estimations. Note that in this case the FSH model performs the mapping and localization in the topological space. Successful position estimation means that the maximum value of the posterior Gaussian belief is within ±1.25 nodes of the real node location in the map; this position estimation was also considered for the other methods tested. Figure 4.11 has the same visualization format as the one for Figure 4.10 described above. The global localization results of Figure 4.11 show that the FSH model outperforms the method proposed by [Dayoub et al., 2011] and the Bag-of-Words method, because the LTM features obtained from the reference representation in the FSH model are able to maintain the representativeness of the environmental appearance from the beginning. This is done thanks to the weighted voting scheme proposed in Equations 4.5 and 4.6.



a)

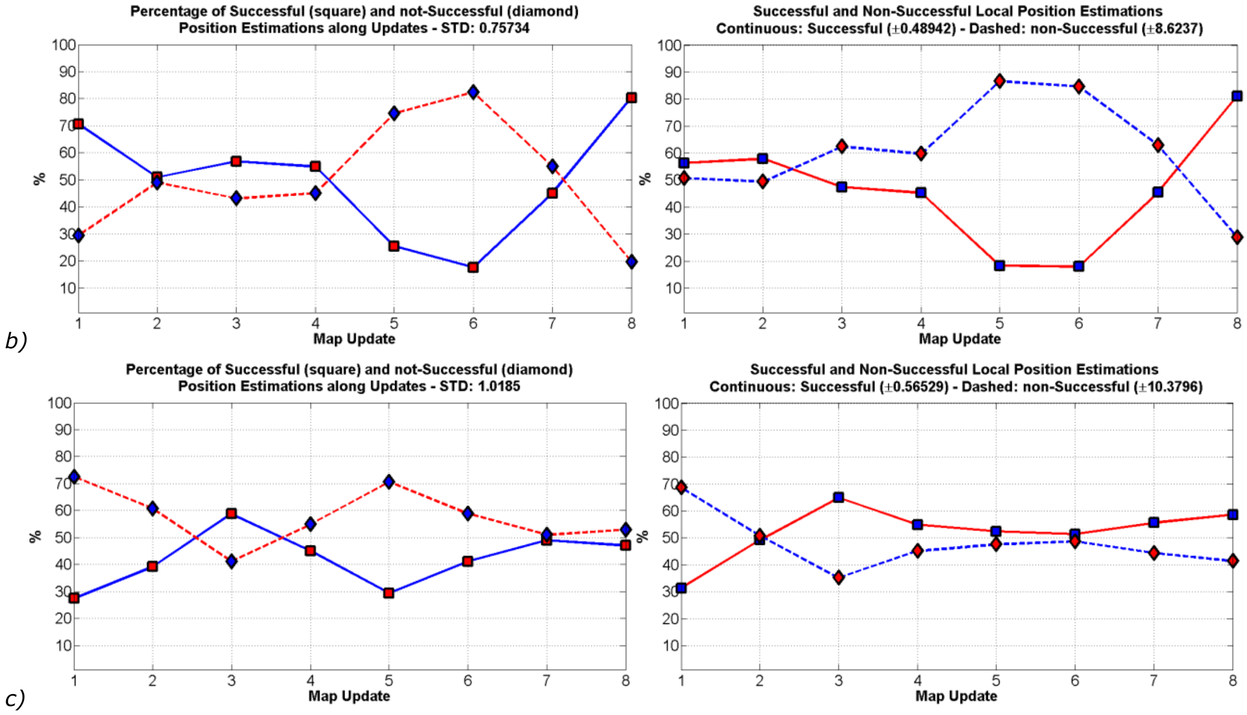


Figure 4.11 Successful and non-successful global and local position estimation along map updates without noise or artificial occlusion. a) The FSH model. b) The method proposed in [Dayoub et al., 2011]. c) The Bag-of-Words method.

The method proposed by [Dayoub et al., 2011] tends to increase it's the percentage of successful position estimations, but after the features state overcomes the STM finite-state-machine. Indoor datasets are full of perceptual aliasing. This becomes a great challenge for the Bag-of-Words method, because in global localization it finds many position hypothesis within the internal Bag-of-Words environment representation, which is given by the voting schema over the visual words.

Observing the experimental results for the successful and non-successful local position estimations, the FSH model holds its tendency to increase the successful position estimations as the map updates increase. The method proposed by [Dayoub et al., 2011] suffers again the consequences of the delayed appearance update representation, but in the end its percentage increases drastically. Despite the fact the map updates 4 to 6 are challenging, the Bag-of-Words method shows a positive difference between the successful and non-successful position estimations along the map updates. Figure 4.11a and Figure 4.11b show clear experimental differences between the FSH model and the method proposed by [Dayoub et al., 2011]; these differences are the consequence of changing the reference view model at each node in the topological map for the FSH, which considers the strength of the environment features and then classifies them as STM or LTM features.

4.5.2.2. *Global and local localization with Gaussian noise, no artificial occlusion*

Figure 4.12 shows the mean position error for global and local localization using a corrupted input image, but without artificial occlusion. This section and the two following it aim to evaluate the FSH model in the presence of Gaussian noise and occlusions. These occlusions are artificially generated by randomly removing a percentage of the input features. Note that these occlusions are in addition to the ones naturally present in the original omnidirectional images in our dataset. For the mean position error in global localization, the end the three approaches have a mean error close to zero, but the levels of uncertainty are lower in the FHS model.

This also means that the matching and outliers removal method used in the topological map building stage is performing well despite the Gaussian noise added.

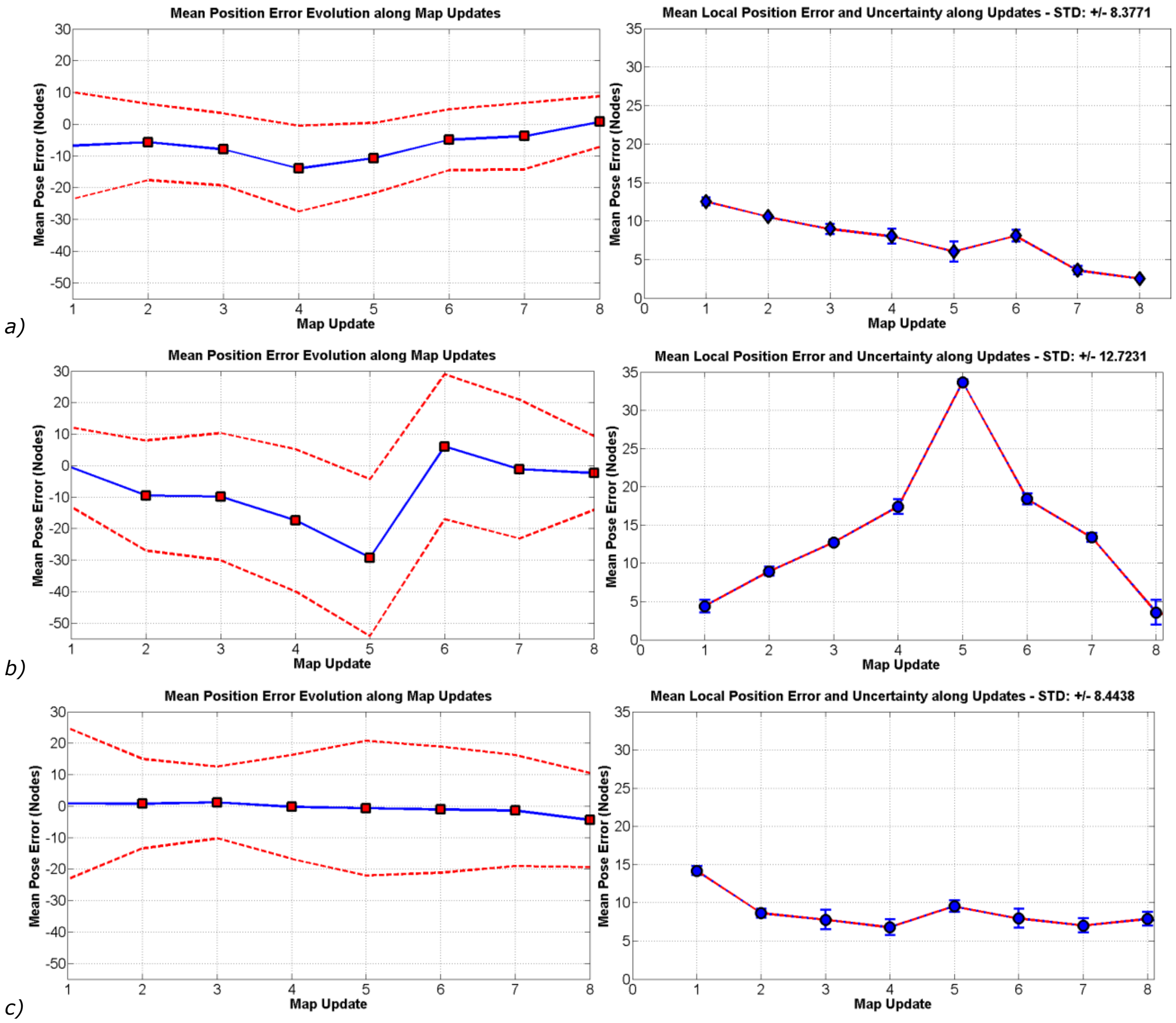


Figure 4.12 Mean position error of global and local localization along map updates with Gaussian noise, and without artificial occlusion. a) The FSH model. b) The method proposed in [Dayoub et al., 2011]. c) The Bag-of-Words method.

From the local localization point of view, Figure 4.12 shows that in the end the mean position error is similar to that achieved without adding Gaussian noise in the three methods tested, but again the levels of uncertainty are lower in the FSH model. The method proposed by [Dayoub et al., 2011] shows the negative effect of having a hard-wired finite-state machine for the rehearsal stage in the STM. The Bag-of-Words method shows an increase of the uncertainty as in both the global and the local mean position error between map updates 4 and 6, despite a two-view geometry check is being done.

Figure 4.13 shows the percentage of successful and non-successful global and local position estimations in the presence of Gaussian noise in the input image. For global localization, it is observed that at the beginning the noise added has a negative impact in the FSH model, but one map update more is enough to have a positive difference between successful and non-successful position estimation. This difference increases as the map updates increase, which does not happen with the other two approaches. This demonstrates that the FSH model coherently deals with the original illumination changes and occlusions, and the Gaussian noise added. This can be observed in the continuous curve going up and down on the left side of Figure 4.13b and 4.13c.

The right part of the Figure 4.13 shows the local position estimations. Clearly, the FSH model outperforms the other two approaches, since it always shows a positive difference between the successful and non-successful position estimations. Observing the experimental results between Figure 4.11 and Figure 4.13, the Bag-of-Words method performs better than the method proposed by [Dayoub et al., 2011], whose major weakness is the finite-state machine conception, which adds a highly sequential component to the feature classification process.

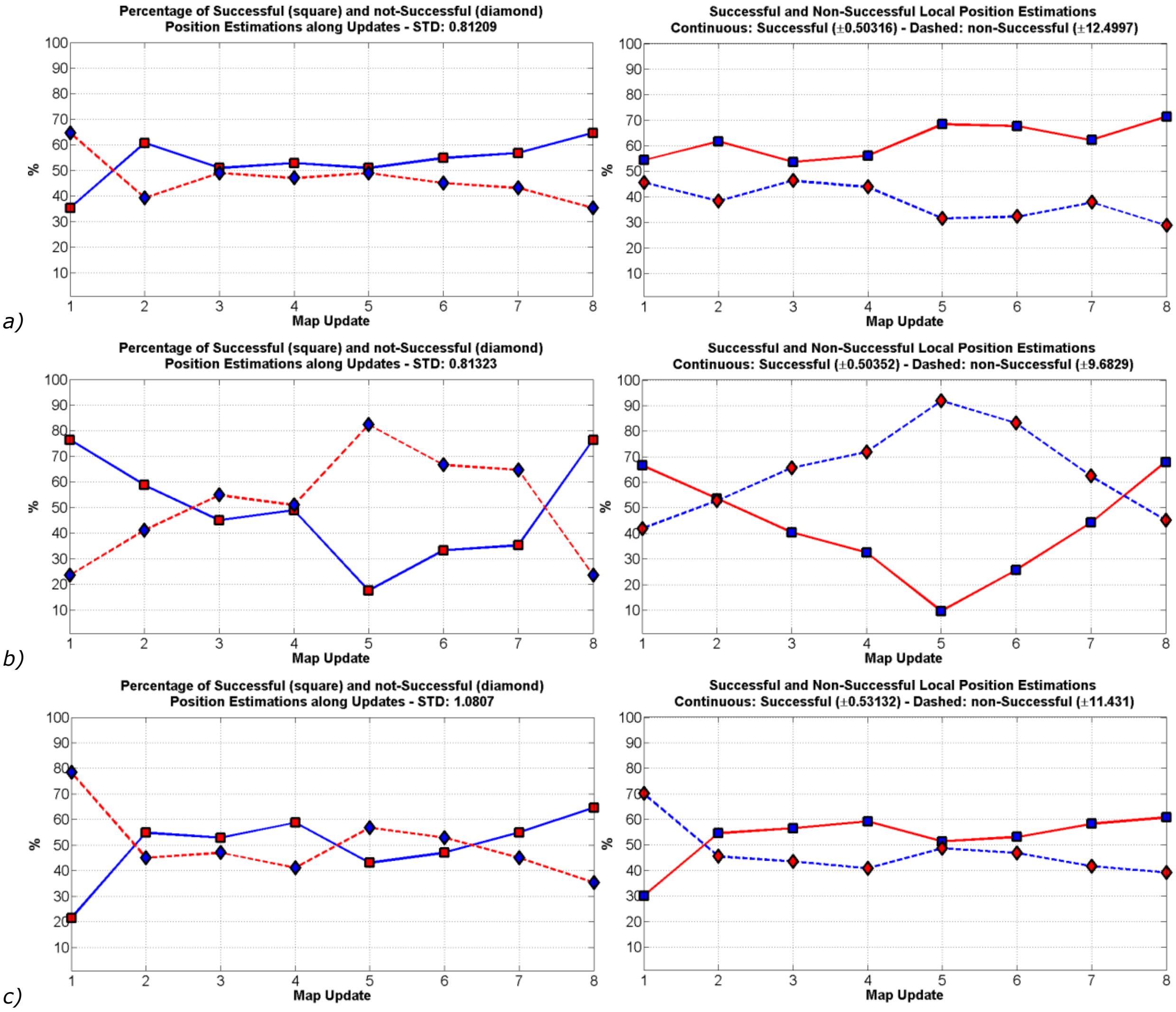


Figure 4.13 Successful and non-successful global position estimation along map updates with Gaussian noise, and without artificial occlusion. a) The FSH model. b) The method proposed in [Dayoub et al., 2011]. c) The Bag-of-Words method.

4.5.2.3. *Global and local localization with Gaussian noise and artificial occlusion of 25%*

This section describes the performance of the FSH model when the input images are corrupted with Gaussian noise and artificial occlusion of 25%. The artificial occlusion was implemented by randomly removing features from the input image. As described earlier, this occlusion is in addition to what is naturally present in the original omnidirectional images, and is caused by pedestrians, bright spots or illumination changes due season or time of the day. The left part of Figure 4.14

shows the mean position error for global localization with Gaussian noise and artificial occlusion of 25%. The FSH model maintains a decreasing error position to zero along the map updates, and has a lower uncertainty level than the other approaches. In terms of local localization error, the method proposed by [Dayoub et al., 2011] still presents an error peak at map update 5, and the Bag-of-Words method behaves similarly to the FSH model. Comparing the experimental results of Figure 4.12 and Figure 4.14, the FSH model and the method proposed by [Dayoub et al., 2011] behave in an especially similar way.

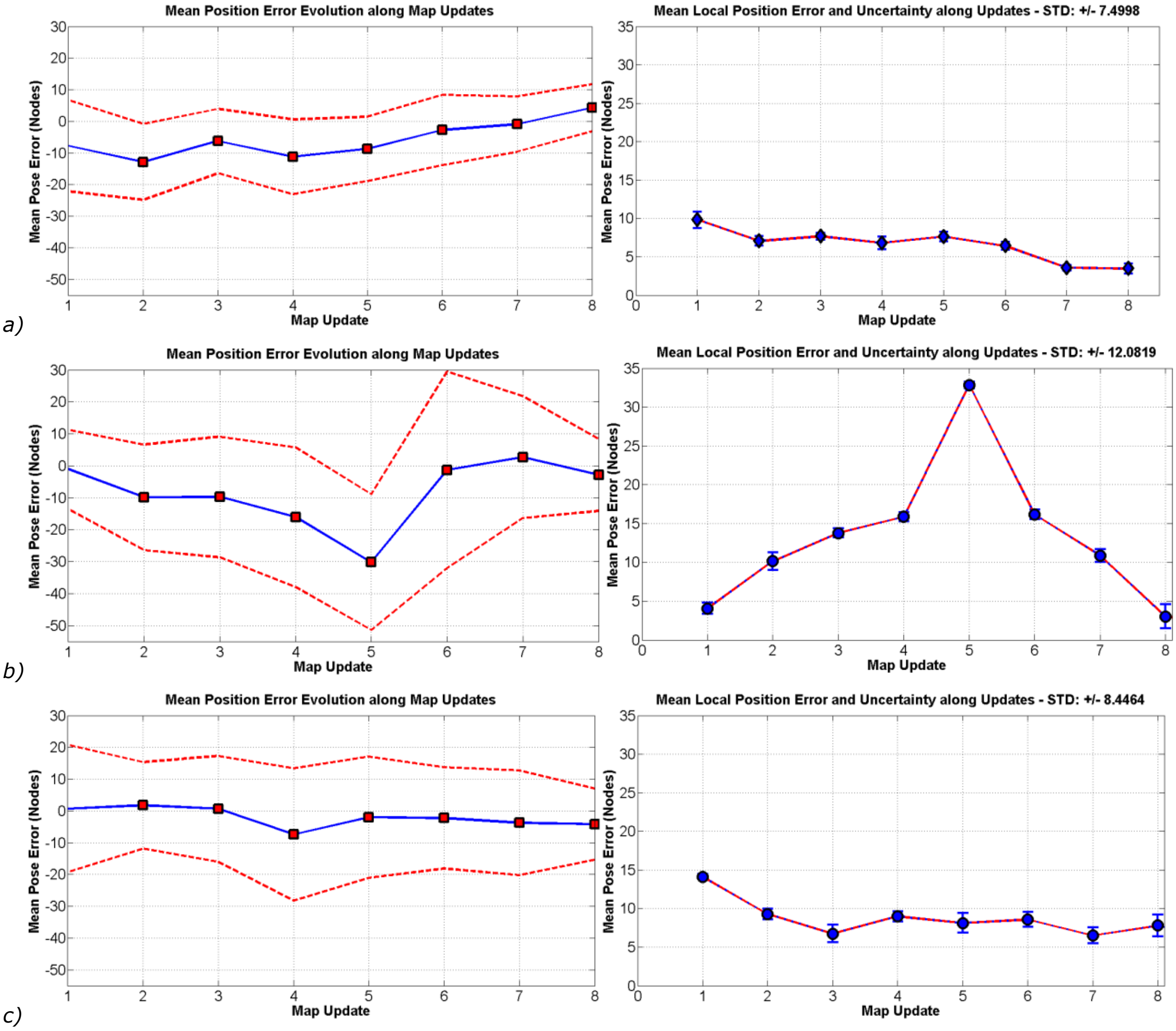
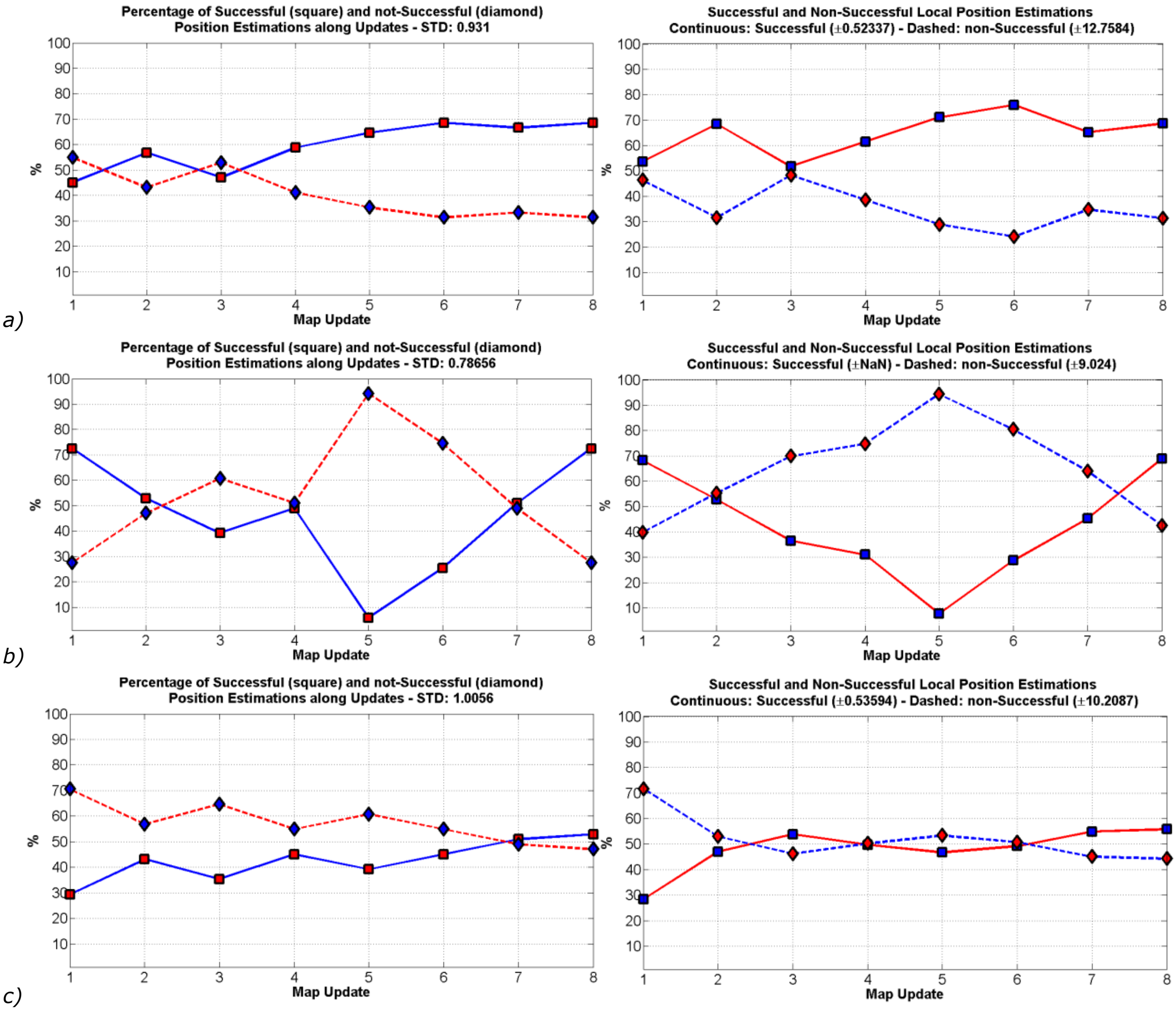


Figure 4.14 Mean position error of global localization along map updates with Gaussian noise, and with artificial occlusion of 25%. a) The FSH model. b) The method proposed in [Dayoub et al., 2011]. c) The Bag-of-Words method.

Despite the noise and artificial occlusion added, the FSH model maintains a good representativeness of the environment's appearance encoded in the LTM features, which allows it to better estimate the robot position than the other approaches. This position estimation is done saving time, computing resources and storage, because the Bag-of-Words method requires the creation of a new dictionary each time the map is updated, and the size of the dictionary greatly depends of the number of images and features. For instance, the FSH model applied to the computed topological appearance-based map has a size of 7.73Mbytes at the eighth update, while in the Bag-of-Words method the map at the eighth update has a size of 64.9Mbytes.

The percentages of successful and non-successful global and local position estimations are presented in Figure 4.15. Here, the FSH model has some difficulties at the beginning, but after the third map update the successful position estimations are better.

Given these challenging conditions of noise and artificial occlusion, the method proposed by [Dayoub et al., 2011] increases the peak values of non-successful position estimations, since less features in the STM finite-state-machine are promoted to the LTM. Despite the fact the Bag-of-Words method finally achieves a positive difference between the successful and non-successful position estimations, the noise and the artificial occlusion added increases the effect of its main weakness: dealing with perceptual aliasing, which is very common in indoor environments.


Percentage of Successful (square) and not-Successful (diamond) Position Estimations along Updates - STD: 0.931

Successful and Non-Successful Local Position Estimations Continuous: Successful (±0.52337) - Dashed: non-Successful (±12.7584)

Percentage of Successful (square) and not-Successful (diamond) Position Estimations along Updates - STD: 0.78656

Successful and Non-Successful Local Position Estimations Continuous: Successful (±NaN) - Dashed: non-Successful (±9.024)

Percentage of Successful (square) and not-Successful (diamond) Position Estimations along Updates - STD: 1.0056

Successful and Non-Successful Local Position Estimations Continuous: Successful (±0.53594) - Dashed: non-Successful (±10.2087)

Figure 4.15 Successful and non-successful global position estimation along map updates with Gaussian noise, and with artificial occlusion of 25%. a) The FSH model. b) The method proposed in [Dayoub et al., 2011]. c) The Bag-of-Words method.

4.5.2.4. Global and local localization with Gaussian noise and artificial occlusion of 50%

Figure 4.16 shows the experimental results for the global and local mean position errors when the input images were corrupted with noise and artificial occlusion of 50%. Figure 4.16a shows how the FSH model performed well on this challenging test, the global position estimation approaches zero and the local position estimation decreases as the map updates increase. The similitude of the results presented for the FSH model in all four of these tests is evidence of the importance of getting a suitable appearance representation of the environment.

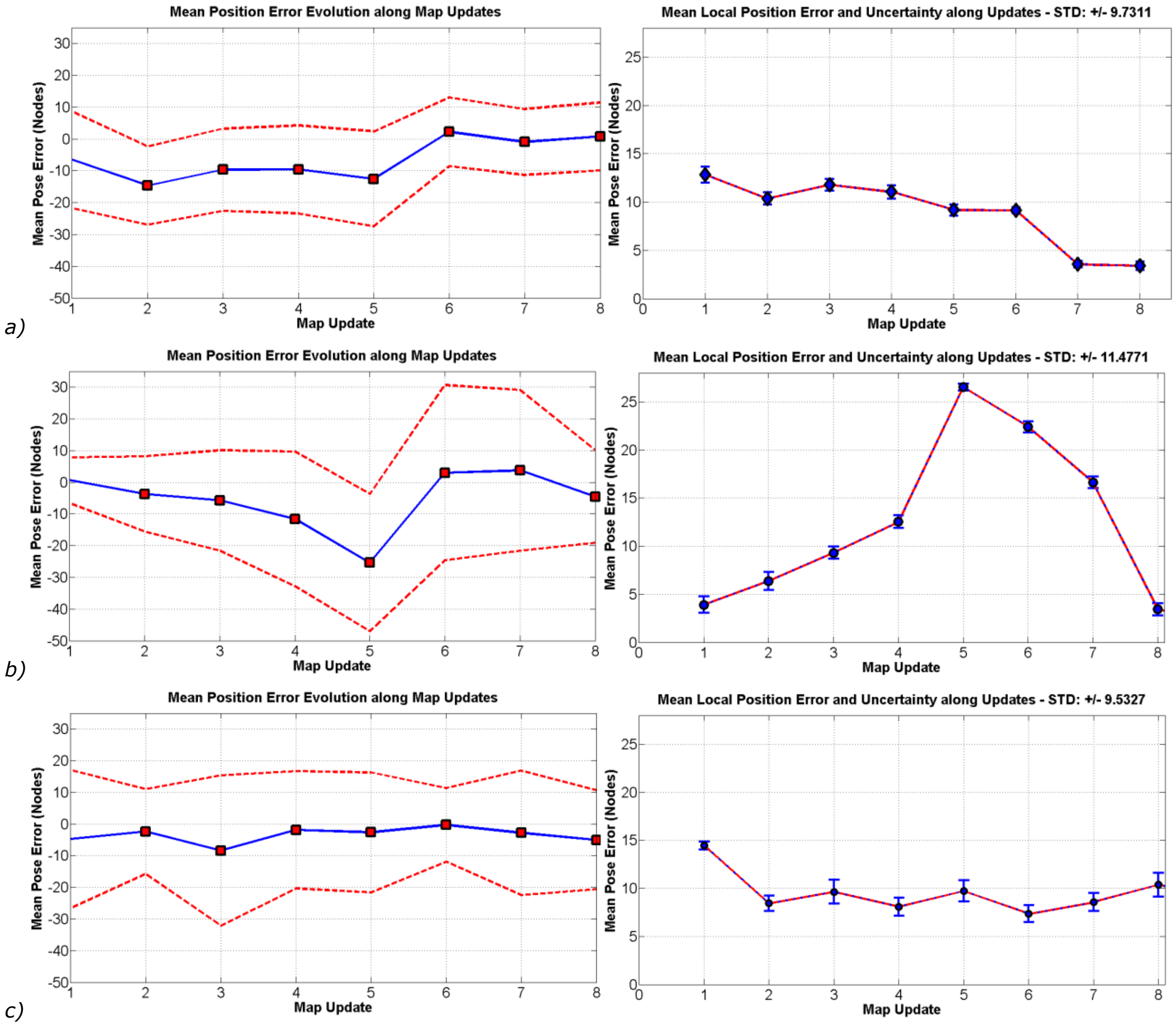


Figure 4.16 Mean position error of global and local localization along map updates with Gaussian noise, and with artificial occlusion of 50%. a) The FSH model. b) The method proposed in [Dayoub et al., 2011]. c) The Bag-of-Words method.

Figure 4.16b shows the global and local position error for the method proposed by [Dayoub et al., 2011]. The error peak at map update five is still present, but one might wonder what happens if the rehearsal stages at the STM are reduced? This test was performed, but the mean position error did not decrease at the same rate as the FSH model, because many LTM features were considered as a reference view and most of them did not deserve to be promoted. In addition, the discrete increments of the state in the finite-state machine framework do not support a real value, which causes a lack of flexibility choosing the more suitable number of states. Figure 4.16c shows the results for the Bag-of-Words method which has some similarities with the results of the FSH model, but it involves an offline process, the uncertainty levels are bigger and as described in Section 4.5.2.3 the time, computing and storage costs are high.

Figure 4.17 shows the percentages of successful and non-successful position estimation in the presence of Gaussian noise and artificial occlusion of 50%. Figure 4.17a shows the experimental results for the FSH model, which after the fourth map update achieve a positive difference between the successful and non-successful global position estimation. This figure also shows evidence of how the assumed sensor model allows progressive improvements in the local position estimation. The increasing tendency of the successful local position estimation curve is also evidence of how the posterior Gaussian belief progressively shrinks.

Figure 4.17b and Figure 4.17c show the experimental results for the method proposed in [Dayoub et al., 2011] and the Bag-of-Words method. The former recovers in the end, but while the reference view at each node does not take into account the LTM features the non-successful position estimations prevail. The latter has no successful global position estimation at all, but the local position estimations are not reliable.

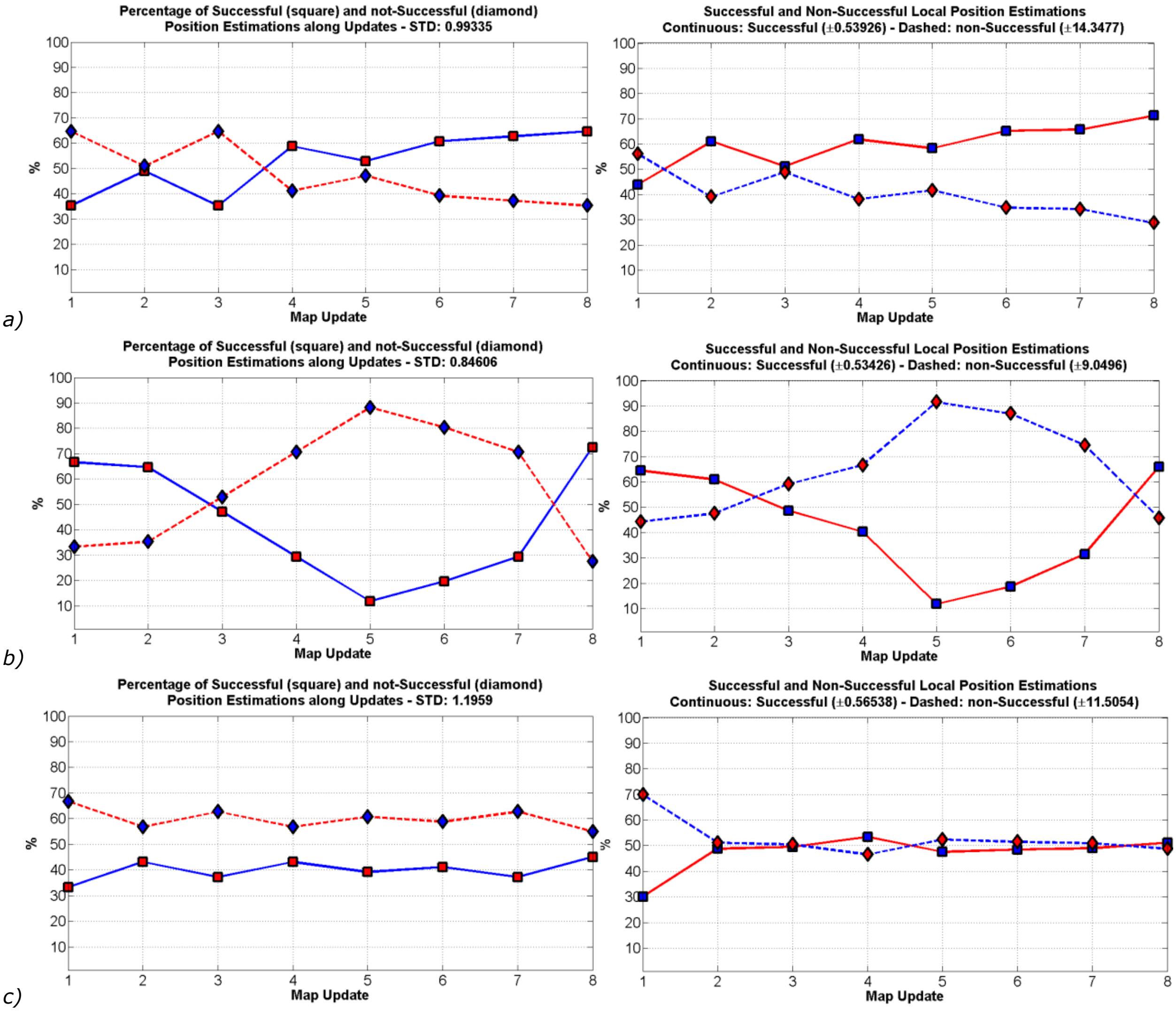


Figure 4.17 Successful and non-successful global and local position estimation along map updates with Gaussian noise, and with artificial occlusion of 50%. a) The FSH model. b) The method proposed in [Dayoub et al., 2011]. c) The Bag-of-Words method.

4.6. DISCUSSION

In this chapter the basis of the FSH model has been described and tested. The FSH model, based on a modified human memory model, implements concepts such as Long-Term (LTM) and Short-Term Memory (STM) as mechanisms to classify features as either stable or non-stable. Unlike other approaches, the FSH model considers a weighted voting scheme to outperform the Atkinson and Shiffrin memory model linearity. STM and LTM features are distinguished using a well known clustering technique as *k-means*, such that only LTM features were used for robot mapping and localization. Using the weighted voting scheme implemented to compute the FSH, it can deal with temporal occlusions caused by dynamic environments and illumination changes caused by time of the day and seasons.

The FSH model was tested in static and dynamic environments. The former included one set of LRF readings and two sets of images acquired over a long period of time; this experiment was the aim to show that the FSH model and the image similarity measure proposed offer better results than a static description of the environment. The latter used a topological map which was updated as many as eight times and a Bayesian-based localization approach for global and local localization experiments. According with these results, the FSH, the weighted voting scheme and the classification of STM and LTM features seems like a promising way to improve appearance-based mapping and localization. The LTM features obtained from the FSH model reference representation are able to maintain the representativeness of the environmental appearance from the beginning of the map creation, which does not happen with the other methods compared here.

Pure topological maps give coarse localization estimations, but they behave very well in long-term navigation and large environments, and they are a suitable method for appearance-based environment models. However, the FSH model can be applied to other SLAM solutions as depicts Equation 4.4, and with the aid of sub-mapping methods such as [Estrada et al., 2005], [Paz et al., 2008], [Pinies & Tardos, 2008], [Aulinas et al., 2010] or classical topological solutions [Thrun et al., 2005] the FSH model can be implemented using parametric and non-parametric SLAM filters. This is the aim of the next chapter, where the FSH model is tested in the context of the FastSLAM algorithm [Montemerlo & Thrun, 2003].

CHAPTER 5

5. *LONG TERM SLAM*

CONTENTS

This chapter presents experimental results obtained using the mobile robot and the dataset described in Section 2.5.1 and 2.5.2 respectively. An important property of the dataset collected is it includes many runs of the same environment at different environmental conditions. The latter covers pedestrians and changes in illumination due time of the day and seasons. Under these changing environmental conditions, the experimental validation of the FSH model was done in terms of qualitative results such as: filtering dynamic objects, map quality over updates and map update. In addition, quantitative results are also included: scan likelihood given the estimated robot position, scalability and matching effort.

5.1. INTRODUCTION

The experimental evaluation of this work was carried out at the University of Girona using the indoor dataset described in Section 2.5.2, which was recorded due to the lack of publicity of datasets in dynamic environments. The more recently work in this sense is the COLD database [Pronobis & Caputo, 2009], however the detailed intrinsic and extrinsic calibration parameters of the sensors involved are not available. This does not allows testing the sensor model proposed in this work

The collected dataset includes seven (7) runs of each floor of the building PIV at the University of Girona. These runs were collected over a period of a year and a total distance of 550m, 445m and 640m for the first, second and third floor respectively. The experimental validation was conducted as follows:

1. At each floor, the initial map built by the SLAM algorithm. In this work, a FastSLAM algorithm [Montemerlo & Thrun, 2003] was used. At global level, a stochastic topological map was built with a maximum of features per node of 10.
2. Seven further runs were performed for each map. At each run a new map is generated, which considers the respective changes in the environment. This new map is loaded in the next run.

Map update and localization accuracy experimental validation show qualitative and quantitative results. Qualitative results were designed to illustrate the following properties: firstly, filtering dynamic objects which appear into the LRF scans and environment features when the FSH model is used; secondly, the map quality and stability are not degraded when the robot re-enters to a known region due position estimation errors, or changes in the environment; thirdly, relevant changes in the environment are accordingly updated in the map, particularly it is important to show how new and relevant features can be easily become part of the LTM due the weighted vote method described in Section 4.4.1.

In the absence of ground-truth, quantitative results involve measuring the performance of the SLAM algorithm using the average likelihood of the range scan reading given the estimated robot position [Biber & Duckett, 2009], and the LTM scan model. Furthermore, to demonstrate the system scalability, the number of LTM and STM features along the map updates was considered. The aim of this test is to show the performance of the STM pruning method discussed in Section 4.4.3, and that the FSH model can deal with large environments and long periods of operation. Last, the average matching effort was measured over the map updates. In this case, the percentage of LTM and STM features with respect to the data association vector is compared with the full matching effort, which takes into account all the available features without using the FSH model.

Chapter 3 was devoted to describe the sensor model proposed in this work. The next two Sections describe the robot motion model used for the Pioneer 3DX, and for self-containment the FastSLAM probabilistic foundations.

5.1.1. *Robot Motion Model*

The Pioneer 3DX is a differential mobile robot which is commanded using a linear and angular velocity through its Aria software interface [AdeptTechnology, 2012]. The robot state can be defined in terms of the velocity motion model [Thrun et al., 2005] as depicted in Equation 5.1.

$$\boldsymbol{x_t} = \boldsymbol{f}(\boldsymbol{x_{t-1}}, \boldsymbol{u_t}) = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} = \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{bmatrix} + \begin{bmatrix} \frac{v}{w}(-\sin(\theta_{t-1}) + \sin(\theta_{t-1} + w\Delta t)) \\ \frac{v}{w}(\cos(\theta_{t-1}) - \cos(\theta_{t-1} + w\Delta t)) \\ w\Delta t \end{bmatrix} \qquad 5.1$$

Where, $\boldsymbol{x_t}$ and $\boldsymbol{x_{t-1}}$ are the current and previous robot states, *v* and *w* are the linear and angular velocities of the robot, and *Δt* is the elapsed time between instants *t-1* and *t*. The velocity motion of Equation 5.1 is valid for $w \neq 0$, then at the limit when $w \to 0$ the velocity motion model can be adapted for straight movement using the L'Hopital's rule on the rightmost term of Equation 5.1, the velocity model for $w = 0$ can be defined as Equation 5.2 shows.

$$\boldsymbol{x_t} = \boldsymbol{f}(\boldsymbol{x_{t-1}}, \boldsymbol{u_t}) = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} = \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{bmatrix} + \begin{bmatrix} v\ \Delta t\ cos(\theta_{t-1}) \\ v\ \Delta t\ sin(\theta_{t-1}) \\ 0 \end{bmatrix} \qquad 5.2$$

Equation 5.1 and 5.2 are non-linear functions of the robot state, thus they need to be linearized in order to estimate the robot pose in the prediction stage of most common SLAM solutions. To do so, the Jacobian with respect to the robot state $\boldsymbol{x_t}$ and the input controls $\boldsymbol{u_t}$ = *[v w]^T* have to be defined. Equations 5.3 to 5.6 depict these Jacobians.

$$\nabla \boldsymbol{f_x^{w\neq 0}} = \begin{bmatrix} 1 & 0 & \frac{v}{w}(-\cos(\theta_{t-1}) + \cos(\theta_{t-1} + w\Delta t)) \\ 0 & 1 & \frac{v}{w}(-\sin(\theta_{t-1}) + \sin(\theta_{t-1} + w\Delta t)) \\ 0 & 0 & 1 \end{bmatrix} \qquad 5.3$$

$$\nabla \boldsymbol{f_u^{w\neq 0}} = \begin{bmatrix} \frac{-\sin(\theta_{t-1})+\sin(\theta_t-1+w\Delta t)}{w} & \frac{v(\sin(\theta_{t-1})-\sin(\theta_{t-1}+w\Delta t))}{w^2} + \frac{v\Delta t(\cos(\theta_{t-1}+w\Delta t))}{w} \\ \frac{\cos(\theta_{t-1})-\cos(\theta_t-1+w\Delta t)}{w} & \frac{-v(\cos(\theta_{t-1})-\sin(\theta_{t-1}+w\Delta t))}{w^2} + \frac{v\Delta t(\sin(\theta_{t-1}+w\Delta t))}{w} \\ 0 & \Delta t \end{bmatrix} \quad 5.4$$

$$\nabla \boldsymbol{f_x^{w=0}} = \begin{bmatrix} 1 & 0 & -v\,\Delta t\,\sin(\theta_{t-1}) \\ 0 & 1 & v\,\Delta t\,\cos(\theta_{t-1}) \\ 0 & 0 & 0 \end{bmatrix} \qquad 5.5$$

$$\nabla \boldsymbol{f_u^{w=0}} = \begin{bmatrix} \Delta t\cos(\theta_{t-1}) & 0 \\ \Delta t\sin(\theta_{t-1}) & 0 \\ 0 & 0 \end{bmatrix} \qquad 5.6$$

5.1.2. *FastSLAM Probabilistic Foundations*

In this work, the FastSLAM 2.0 algorithm [Montemerlo et al., 2003] was used for map building and localization. It uses particle filtering and it has been adapted to support the sensor model described in Chapter 3. In this SLAM algorithm, conditionally independence is assumed given the robot poses, and then the posterior can be factored as depicted in Equation 5.7.

$$p(\boldsymbol{x_t}, \boldsymbol{M}|\boldsymbol{z_{1:t}}, \boldsymbol{u_{1:t}}) = p(\boldsymbol{x_t}|\boldsymbol{z_{1:t}}, \boldsymbol{u_{1:t}}) \prod_{n=1}^{N} p(\boldsymbol{m_n}|\boldsymbol{x_t}, \boldsymbol{z_{1:t}}) \qquad 5.7$$

Where, *t* is the current time step and *N* is the current number of features. In FastSLAM each particle is denoted by $\boldsymbol{y_t^k} = \langle \boldsymbol{x_t^k}, \boldsymbol{\mu_{1,t}^k}, \boldsymbol{\Sigma_{1,t}^k}, \dots, \boldsymbol{\mu_{N,t}^k}, \boldsymbol{\Sigma_{N,t}^k} \rangle$ where, *k* is the particle index, $\boldsymbol{x_t^k}$ is the path estimate of the robot, and $\boldsymbol{\mu_{n,t}^k}$ and $\boldsymbol{\Sigma_{n,t}^k}$ are the mean and variance of the Gaussian representing the *n-th* feature location of the *k-th* particle which are estimated using independent Kalman filters. The filtering process overview is explained as follows:

1. **Getting measures** – Vertical edge positions with respect to the LRF are gathered: $\boldsymbol{z_{n,t}}$ = *[$\rho_{n,t}$ $\phi_{n,t}$]^T*, where $\rho_{n,t}$ and $\phi_{n,t}$ are the range and the azimuth of the *n-th* vertical edge at time *t* (Chapter 3).
2. **Sampling new poses** – A new pose $\boldsymbol{x_t}$ is sampled using the motion model described in Section 5.1.1. This is done drawing a sample according to the motion posterior $\widehat{\boldsymbol{x_t^k}} \sim p(\boldsymbol{x_t^k}|\boldsymbol{x_{t-1}^k}, \boldsymbol{u_t})$ where, $\boldsymbol{x_{t-1}^k}$ is the posterior estimate for the robot location at time *t-1* in the *k-th* particle, and $\boldsymbol{u_t}$ is the command motion.
3. **Data association** – Given the current set of features for the *k-th* particle, the current set of observations $\boldsymbol{z_t}$ and the current predicted pose $\widehat{\boldsymbol{x_t^k}}$, the data association method described in Section 3.5.2 and Section 4.4.3 is used to find the observed features $\boldsymbol{H_t}$.
4. **Update observed features** – For each observed feature in each *k-th* particle, the standard EKF expressions were applied to obtain the mean ($\boldsymbol{\mu_{H_t,t}^k}$) and covariance ($\boldsymbol{\Sigma_{H_t,t}^k}$) of these features as shown Equations 5.8 to 5.10.

$$\boldsymbol{K}_t^k = \boldsymbol{\Sigma}_{H_t,t-1}^k . \boldsymbol{J}_{m,t}^{k.T} \left(\boldsymbol{J}_{m,t}^k . \boldsymbol{\Sigma}_{H_t,t-1}^k . \boldsymbol{J}_{m,t}^{k.T} + \boldsymbol{Q}_t\right)^{-1} \quad 5.8$$

$$\boldsymbol{\mu}_{H_t,t}^k = \boldsymbol{\mu}_{H_t,t-1}^k + \boldsymbol{K}_t^k (\boldsymbol{z}_t - \widehat{\boldsymbol{z}_t^k}) \quad 5.9$$

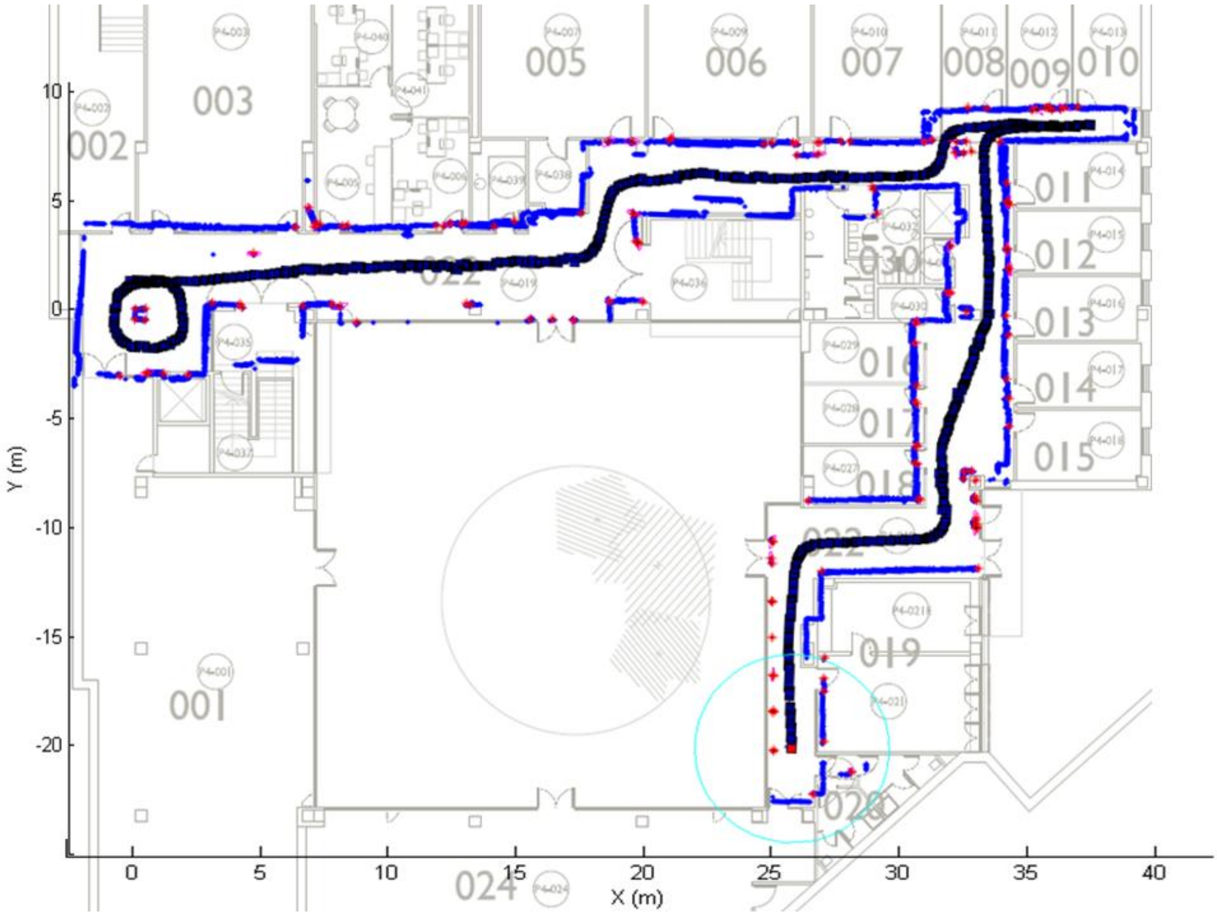
$$\boldsymbol{\Sigma}_{H_t,t}^k = \left(\boldsymbol{I} - \boldsymbol{K}_t^k \boldsymbol{J}_t^k\right) . \boldsymbol{\Sigma}_{H_t,t-1}^k \quad 5.10$$

Where, $\boldsymbol{J}_{m,t}^k$ is the Jacobian of the measurement model with respect to the feature coordinates (see Section 3.5.1) and $\boldsymbol{Q_t}$ is the sensor uncertainty.

5. **Re-sampling** – The low variance re-sampling method is used in this work, where the importance factor of each particle depends on their measurement probability.

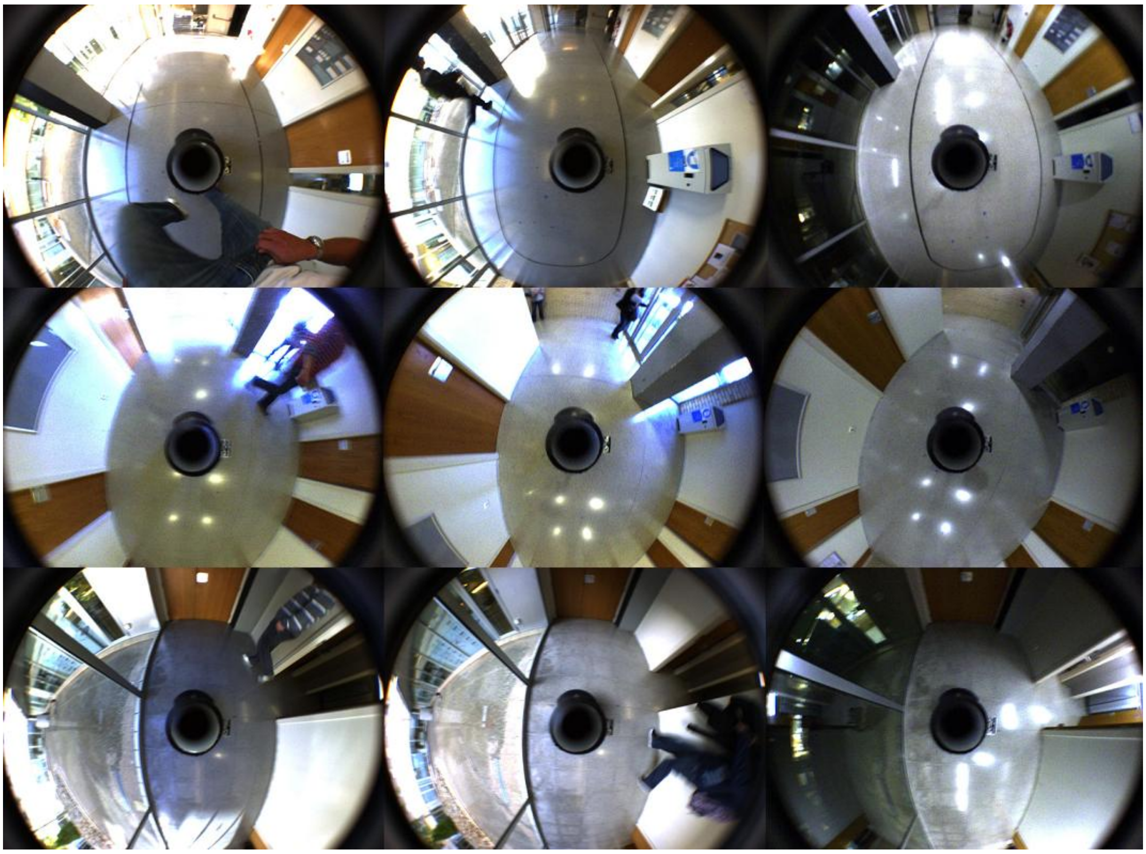
5.2. EXPERIMENTAL CONDITIONS

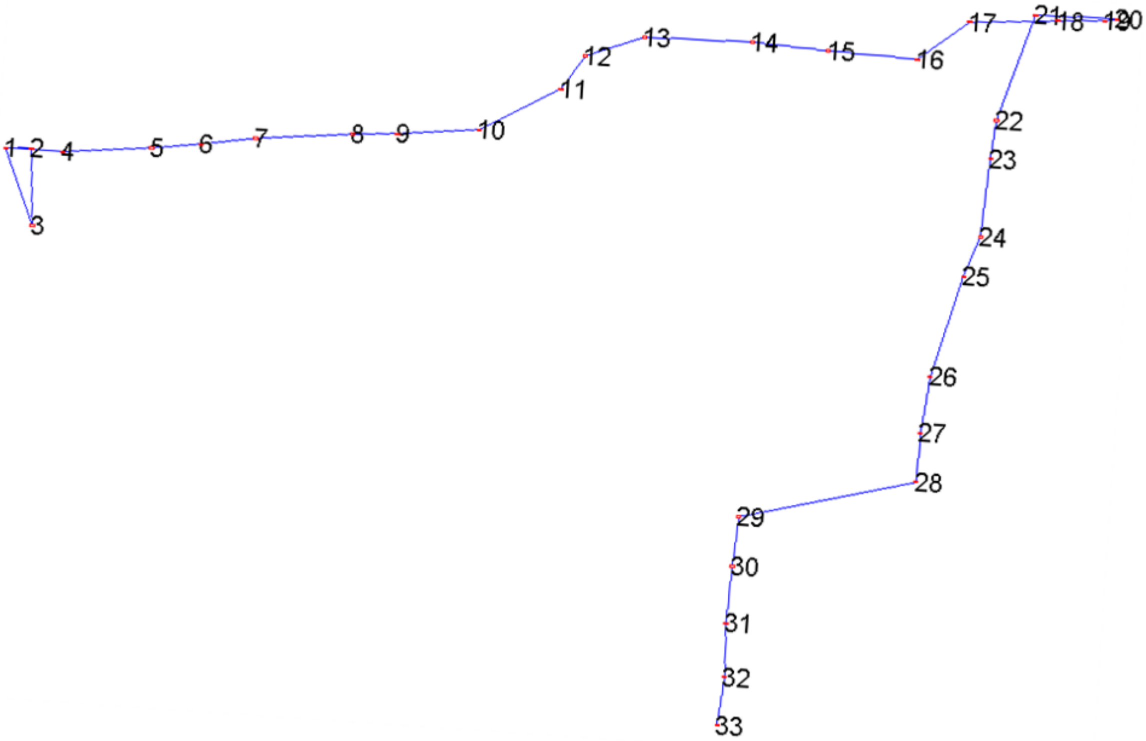
Figure 5.1 shows the estimated maps of level P0, P1 and P2 of the building PIV at the University of Girona, a set of typical images and the map graph. These maps are on the corresponding CAD maps in order to observe the match of the registered LRF scans, the environmental features colored in red with respect the CAD map. The maps shown in Figure 5.1 correspond to the first SLAM run of the seven runs performed to obtain the results discussed in this chapter. It can be observed from Figure 5.1a, 5.1d and 5.1g that the mapped area for each floor corresponds to an area of 30m by 40m (height x width) for level P0, 20m by 40m (height x width) for level P1 and 40m by 40m (height x width) for level P2.



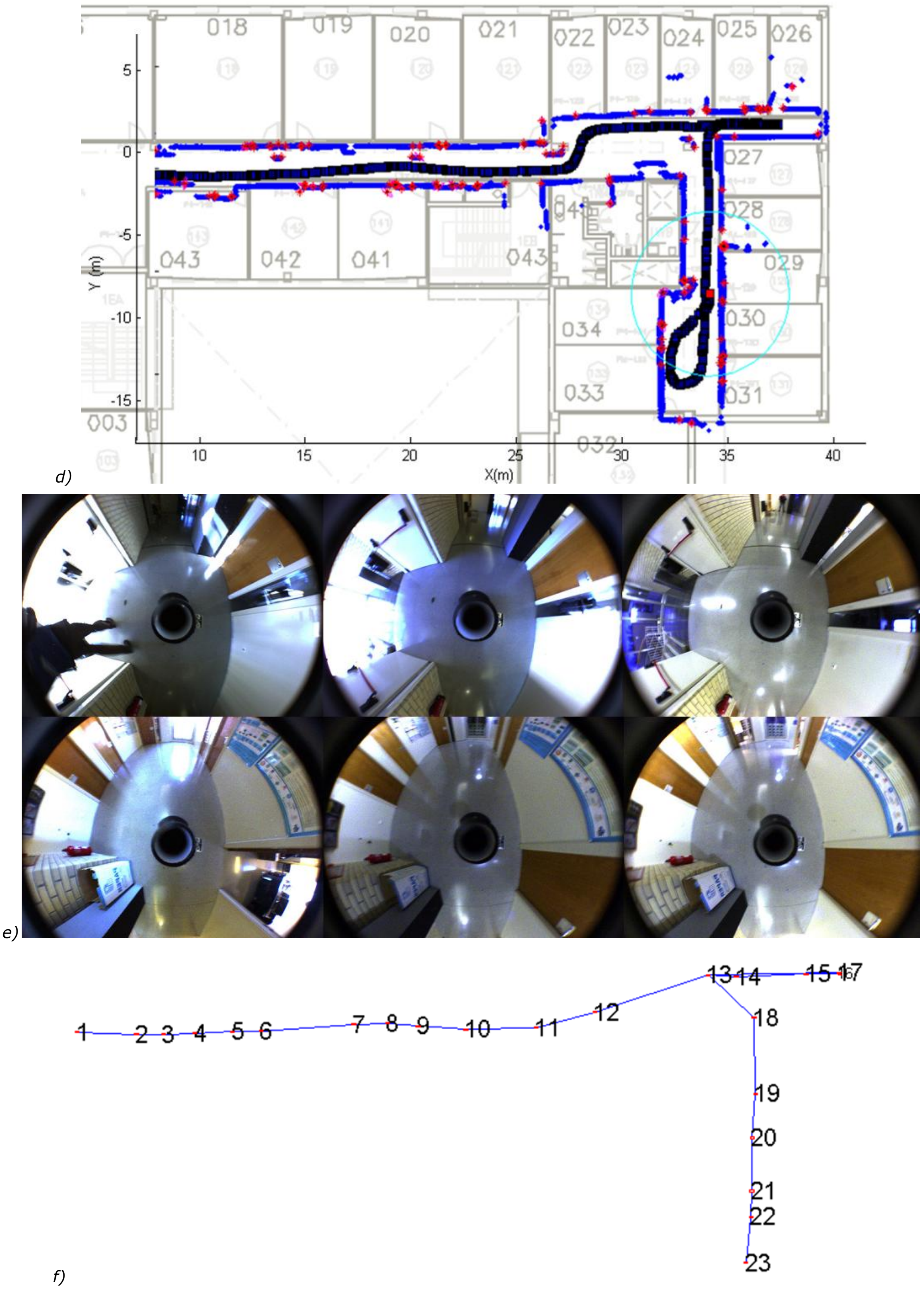
a)

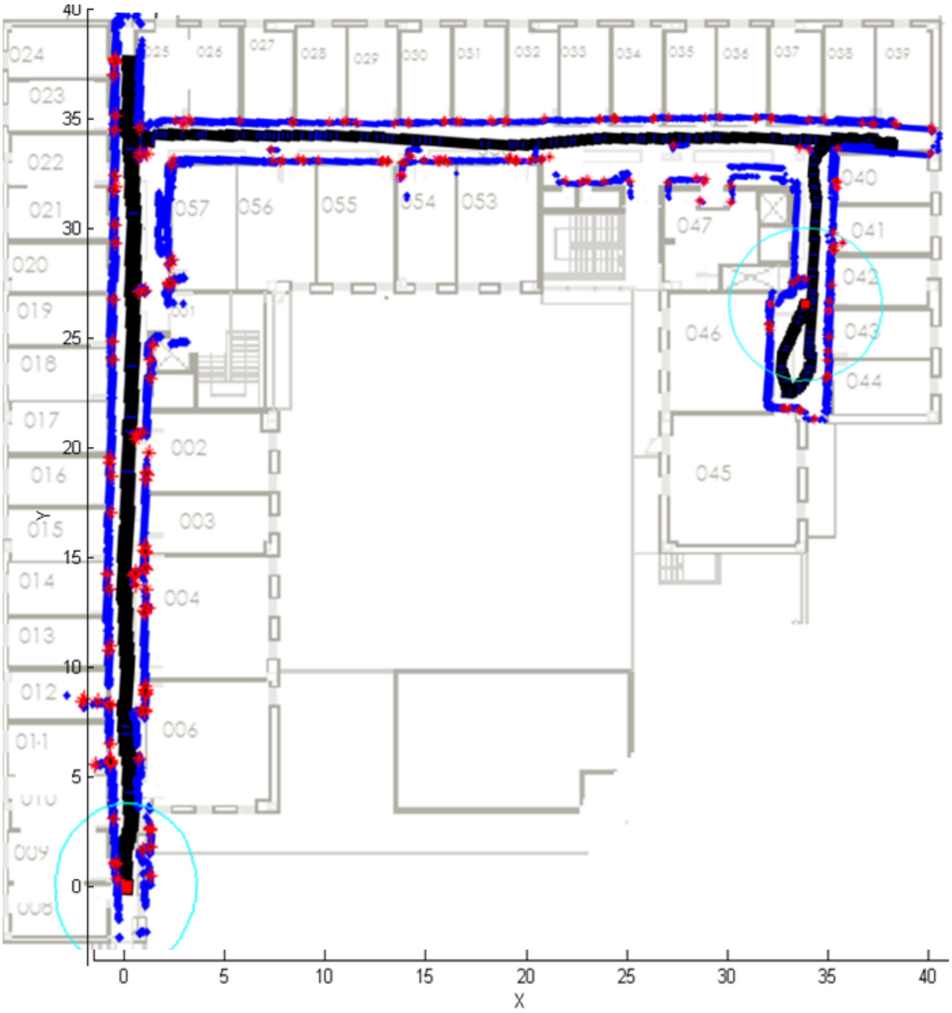
b)




1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
21
18
19
20
22
23
24
25
26
27
28
29
30
31
32
33

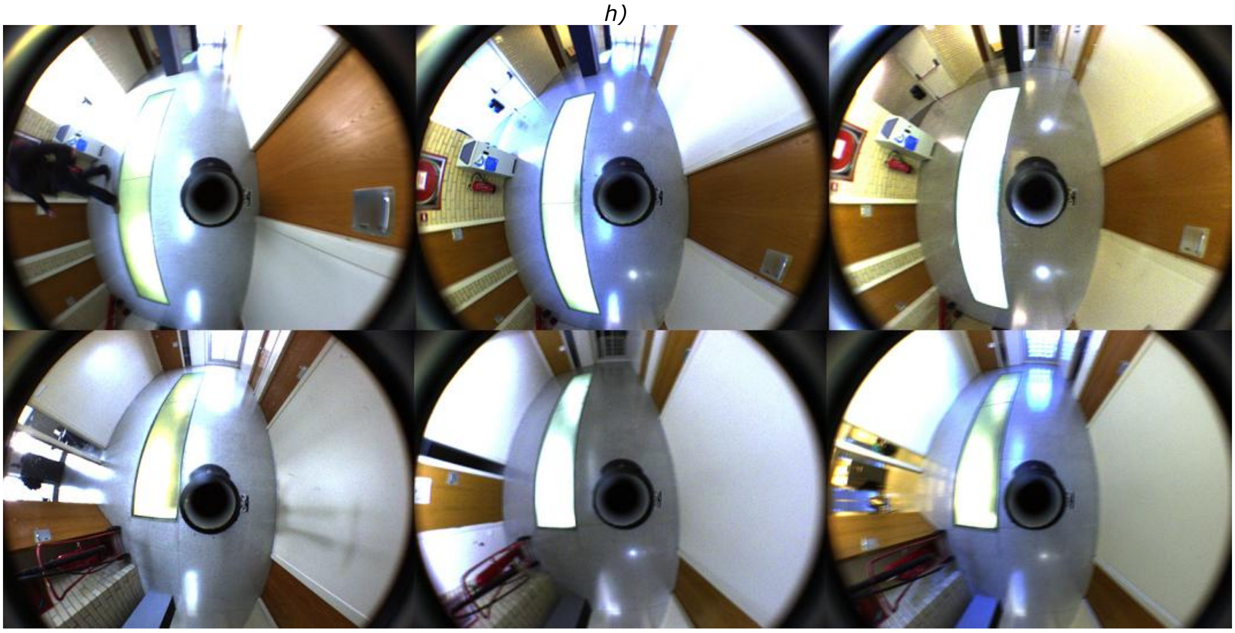
c)


d)
018
019
020
021
022
023
024
025
026
027
028
029
030
031
032
033
034
041
042
043
003
5
0
-5
-10
-15
Y (m)
10
15
20
25
30
35
40
X(m)
e)
f)
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23


40
35
30
25
20
15
10
5
0
Y
0
5
10
15
20
25
30
35
40
X
024
023
022
021
020
019
018
017
016
015
014
013
012
011
010
009
008
025
026
027
028
029
030
031
032
033
034
035
036
037
038
039
040
041
042
043
044
045
046
047
053
054
055
056
057
002
003
004
006

g)

h)



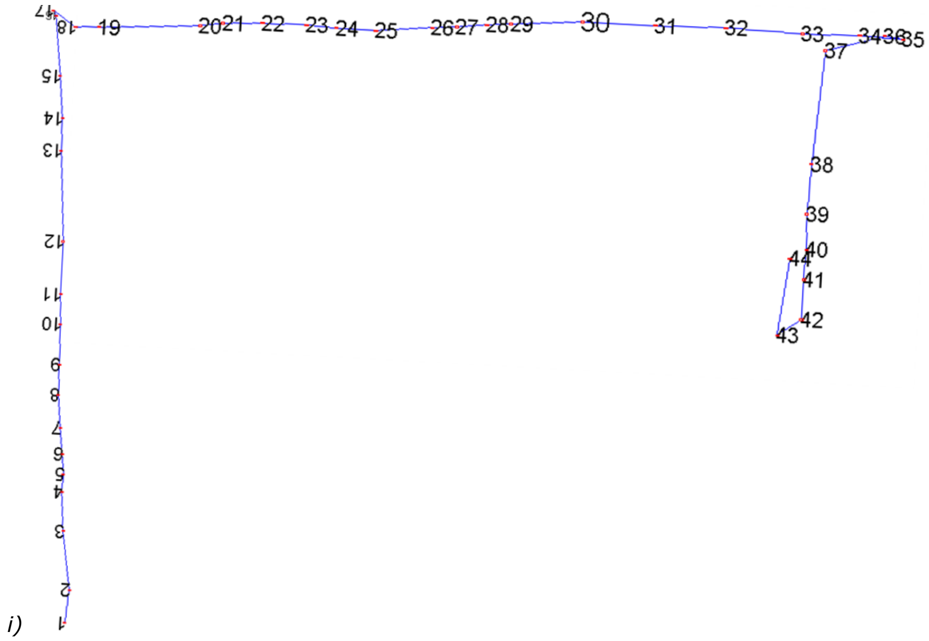


Figure 5.1 Building PIV of the University of Girona. a-c) Map, typical images and graph of level P0.d-f) Map, typical images and graph of level P1.g-i) Map, typical images and graph of level P2.

As can be observed from Figure 5.1, the size of the map is large causing visualization problems if a detailed view is needed. For this reason, Figure 5.1c, 5.1f and 5.1i show the equivalent graph of each map. This topological representation of the environment makes easy the visualization of the results presented in this chapter, because it allows observing how the dynamic objects are filtered out, or detailing the map quality or how the map is repaired in further SLAM runs can be better analyzed using portions of the whole map. Particularly, portions of the map where pedestrians and changes in illumination are more present.

The typical omnidirectional images shown in Figure 5.1b, 5.1e and 5.1h display the change of the environment illumination, and the pedestrians crossing by. For instance, each row of Figure 5.1b corresponds to nodes 7, 26 and 33 of the map of level P0; where big windows are present modifying the appearance of the environment from the illumination point of view. In these situations the vertical edge detection is not easy. Each row of Figure 5.1e corresponds to nodes 10 and 21 of the map of level P1; here, the camera cannot find the right gain due the big changes in illumination, which is low along the corridor but it is high through the left window. Each row of Figure 5.1h corresponds to nodes 29 and 41 of the map of level P2; in these images as well as in the previous ones, it can be observed that the environment does not stay static, some doors are open and later these are closed, or the trash cans change their position.

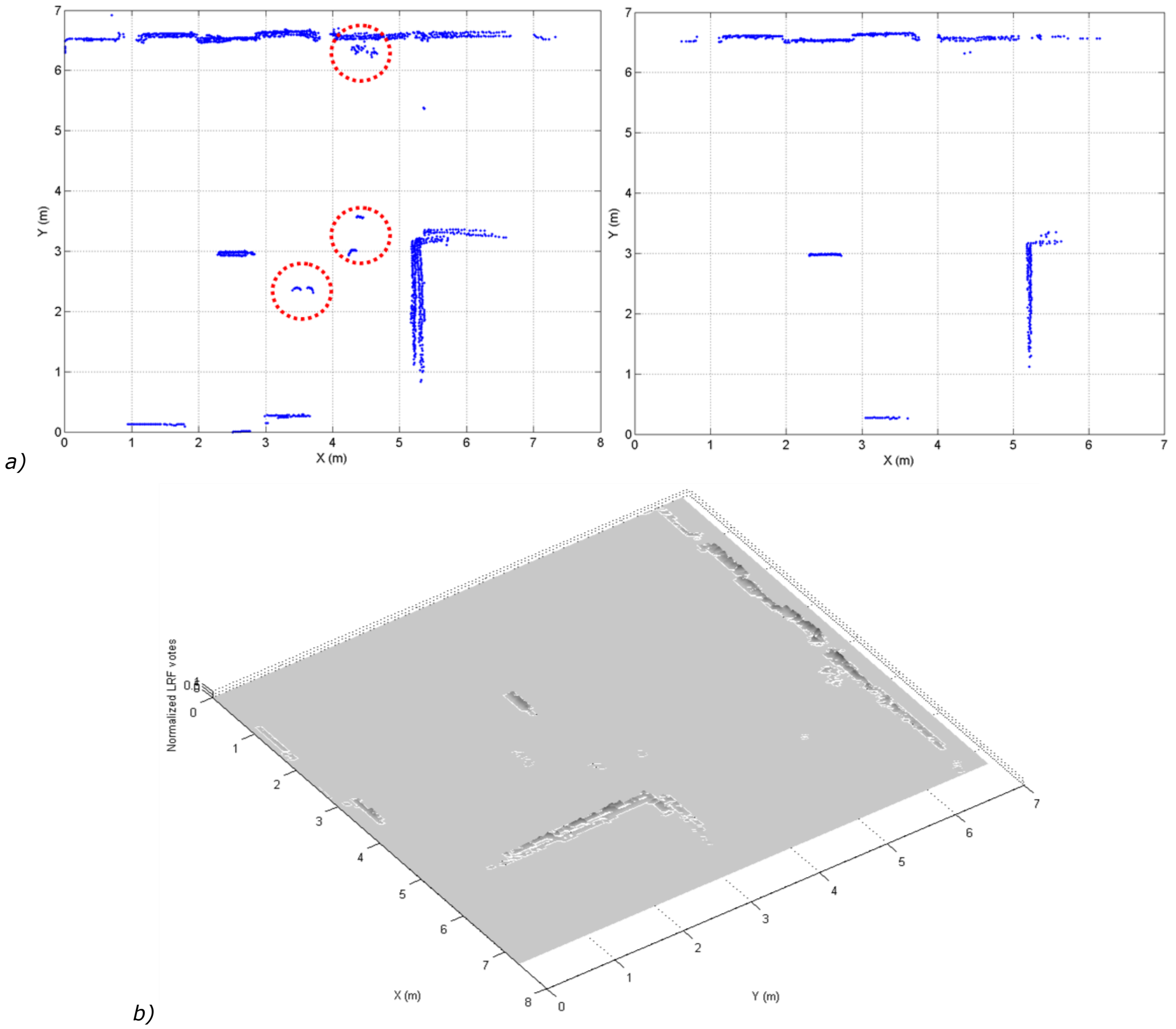
5.3. FILTERING DYNAMIC OBJECTS

Dynamic objects cause basically two main problems: firstly, if they are not handled properly, these objects introduce localization errors [Burgard et al., 2007]; and secondly, the measurements of dynamic objects carry no information in order to estimate the vehicle pose [Wang et al., 2003], in addition these measurements cause spurious features which consume computing time in the data association process if they are not filtered out properly.

Figure 5.2 show the LRF readings of the map of level P0, particularly the nodes 1, 13 and 33 of the topological map. Observing Figure 5.1a and 5.1c, these nodes corresponds to populated places. The left column of Figure 5.2(a, c, e) shows the

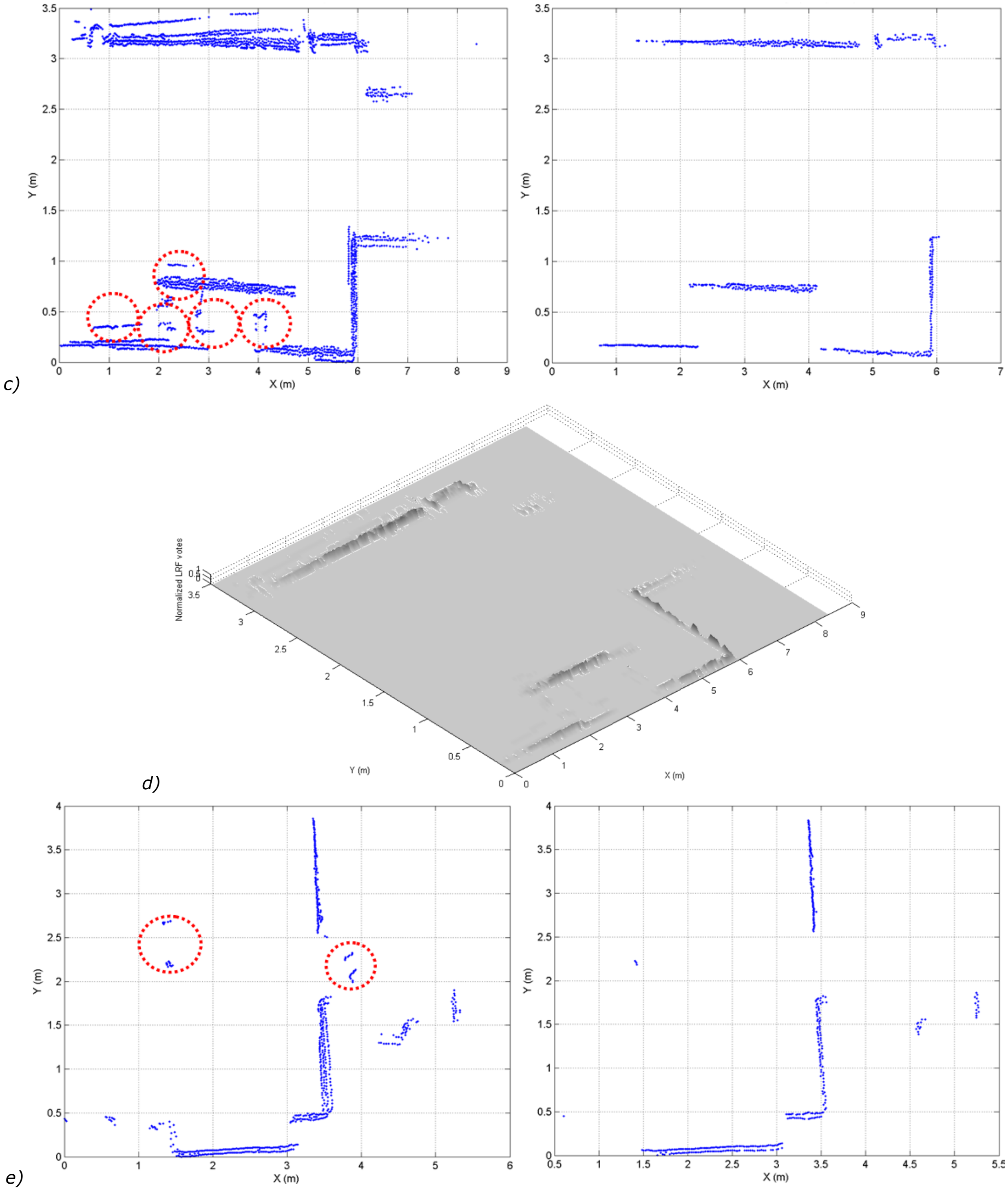
LRF readings without considering the FSH model. Here, the dotted circles show the pedestrians crossing by, and these graphs show the alignment errors of the laser scans which tends to generate thick and blurred walls.

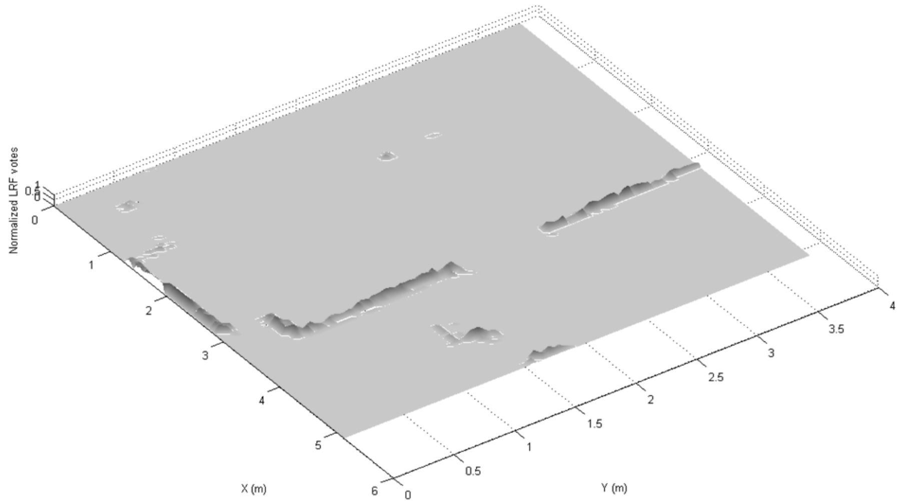
On the other hand, the right column of Figure 5.2(a, c, e) shows the LTM laser scans extracted from the FSH model. As a result of applying the FSH model, not only the pedestrians are filtered out, but the more stable laser readings persist showing a more accurate appearance of the environment. Particularly, Figure 5.2c which corresponds to node 13 shows in the dotted circles a group of persons sitting on a bench, and moving around; this situation cause evident registration errors, however the LTM laser scans shows a more accurate scan alignment.



a)

b)

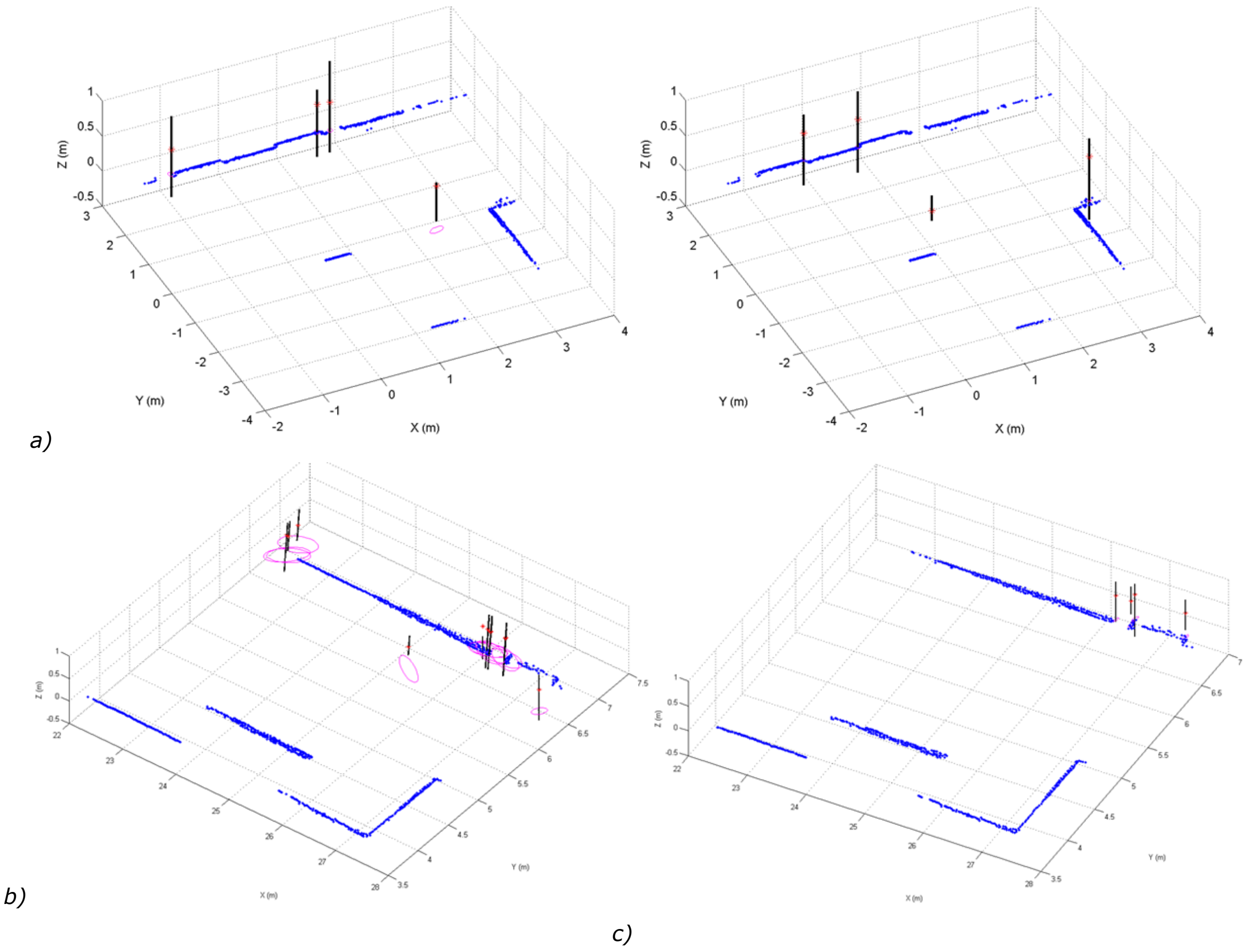

c)
Y (m)
X (m)
0
0.5
1
1.5
2
2.5
3
3.5
1
2
3
4
5
6
7
8
9
d)
Normalized LRF votes
Y (m)
X (m)
e)
Y (m)
X (m)
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
5.5



f)

Figure 5.2 LRF readings of the map of level P0 without (left column) and with (right column) the FSH model. a) Node 1. b) LRF votes for Node 1. c) Node 13. d) LRF votes for Node 13. e) Node 33. f) LRF votes for Node 33.

Figure 5.2b, 5.2d and 5.2f correspond to the LRF votes from which the LTM scans were computed. Observing these figures, a common property is the wide thickness of the walls, which is not observed in the right column of Figure 5.2a, 5.2c and 5.2e. In addition, in Figure 5.2b, 5.2d and 5.2f the lighter the color the more spurious is the LRF reading; however, despite of the high value of the LTM scan votes, the spurious LRF readings can be appreciated.



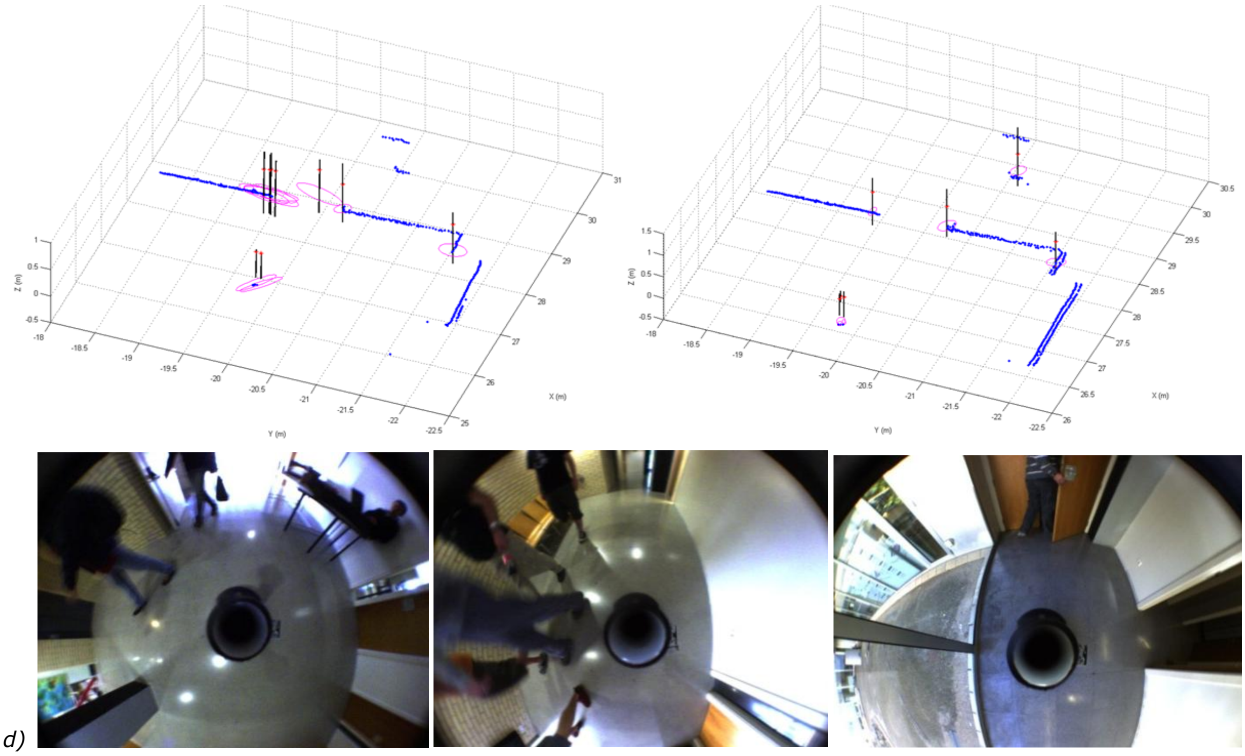


Figure 5.3 STM (left column) and LTM (right column) features of the corresponding nodes in Figure 5.2. a) Node 1 of the map of level P0. b) Node 13 of the map of level P0. c) Node 33 of the map of level P0. d) Omnidirectional images corresponding to the Node 1, 13 and 33.

Figure 5.3 shows the STM and LTM features of the corresponding nodes in Figure 5.2 in the left and right column respectively. STM features can be temporarily created by dynamic objects; however they are not part of the map until they are rehearsed (re-observed) and then become part of the LTM. Comparing the pedestrian in Figure 5.2a (left column) and the Figure 5.3a (left column), it can be observed that a spurious vertical edge was created. Nonetheless, the LTM version of the map does not show this spurious feature, even though this spurious feature is likely to be deleted.

This situation is repeated in the map of the node 13 (Figure 5.3b) and in the map of the node 33 (Figure 5.3c). Focusing the attention on the right column of Figure 5.3, it is worth noting that the LTM scans and the LTM vertical features are consistent with the corresponding vertical edges in the scene. Figure 5.3d shows three omnidirectional images corresponding to nodes 1, 13 and 33 of the map of level P0. These images show the environmental conditions when the LRF readings and the vertical edges were taken. Note that these disturbances do not appear in the LTM scans of Figure 5.2 or the LTM features of Figure 5.3.

5.4. MAP QUALITY OVER UPDATES

SLAM solutions are not error free, this causes erroneous feature position estimations and in consequence LRF scan alignment errors. After this first error, further errors are more likely. In long runs, this situation probably could make the filter diverges [Kretzschmar et al., 2010]. Therefore, an important result is that the FSH map model of the environment is stable over time.

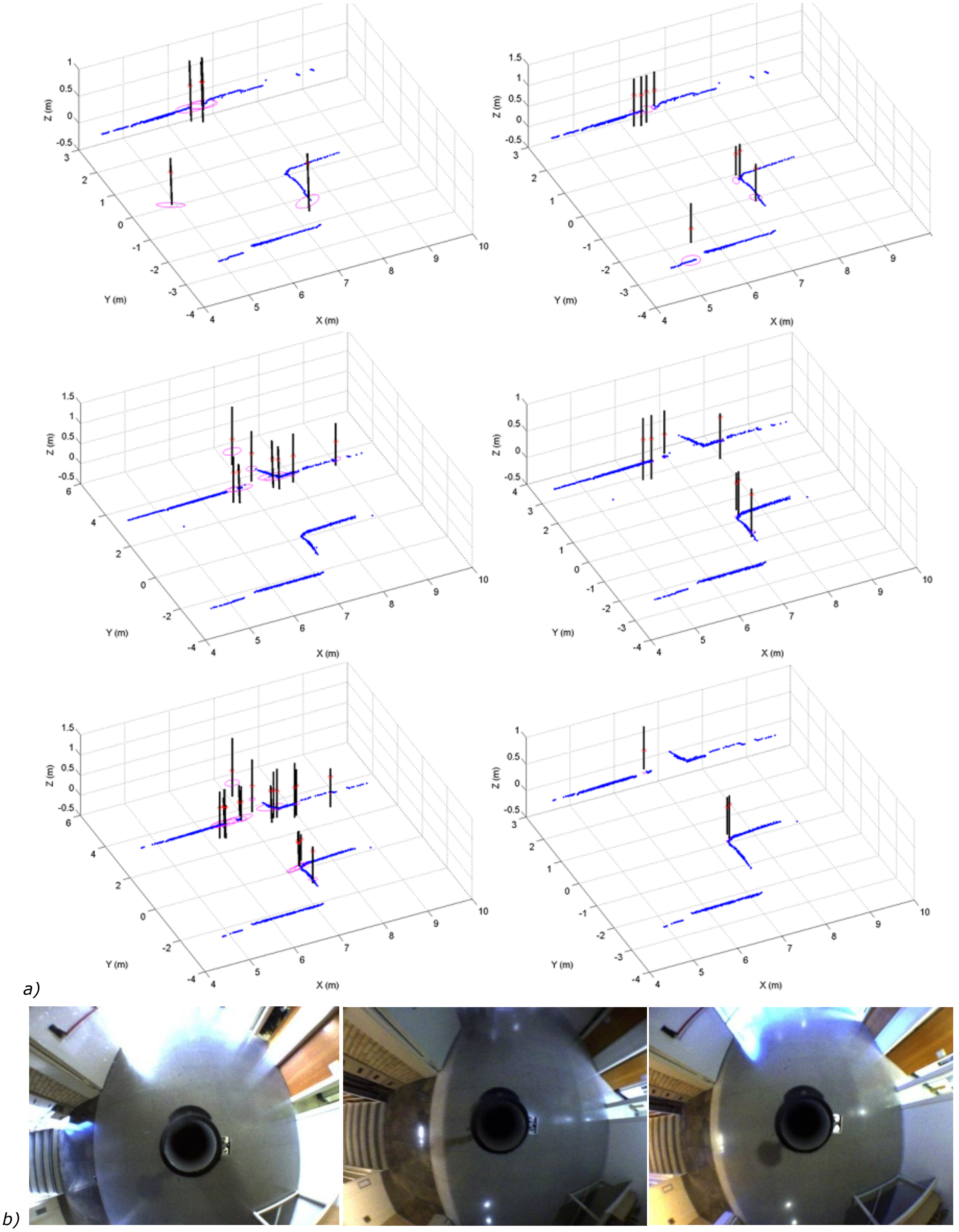


Figure 5.4 a) STM and LTM map versions of node 5 at level P0. Each row corresponds to one different map update. The left and right columns show the STM and LTM map respectively (laser scans and features). b) Omnidirectional images corresponding to each row in part a).

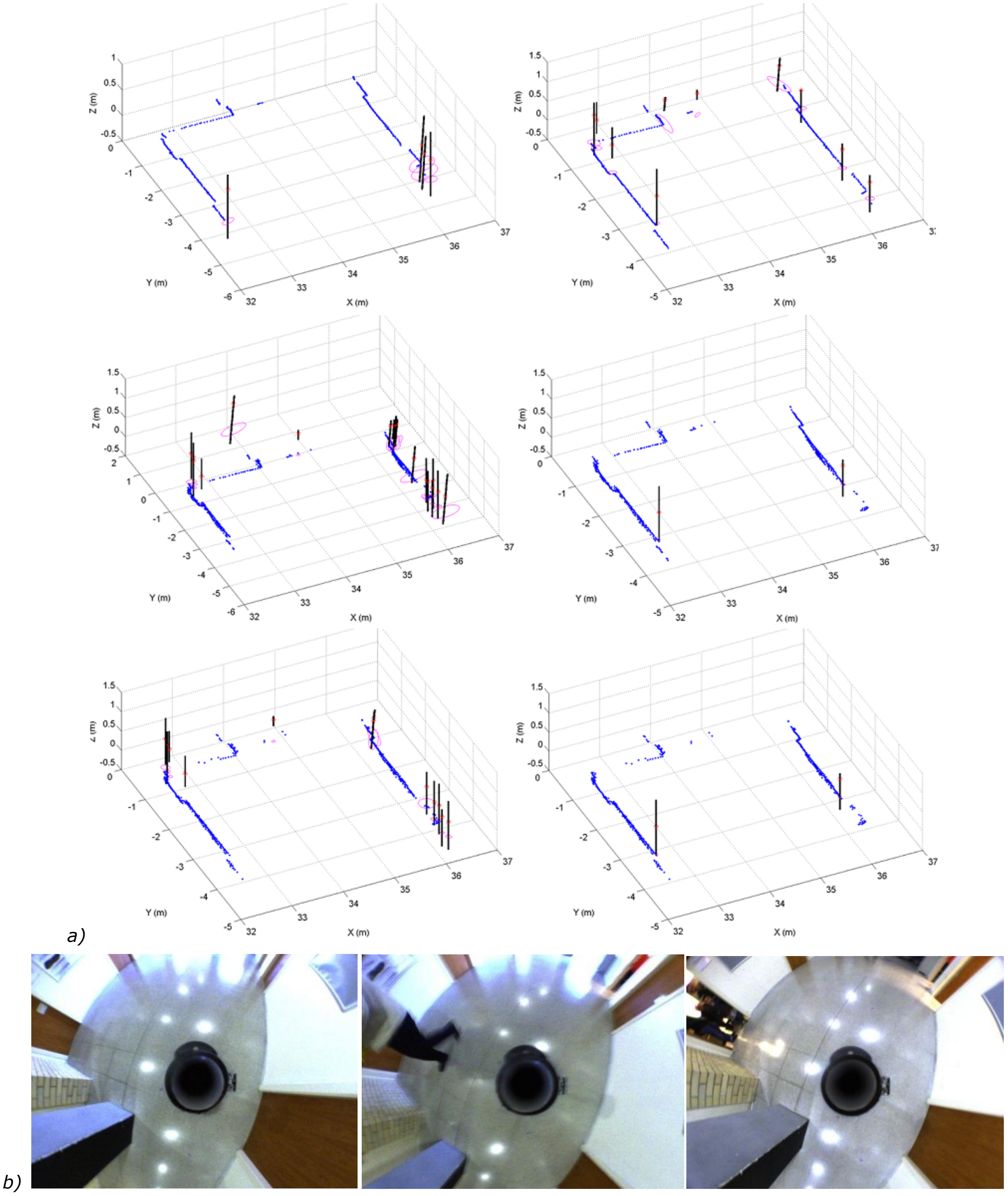


Figure 5.5 a) STM and LTM map versions of node 24 at level P0. Each row corresponds to one different map update. The left and right columns show the STM and LTM map respectively (laser scans and features). b) Omnidirectional images corresponding to each row in part a).

In this work, three different maps were estimated namely the level P0, P1 and P2 of building PIV of the University of Girona. However, observing the map quality of these maps as a whole is difficult. For this reason, Figure 5.4 to Figure 5.8 show a set of local maps and their change over the map updates. These figures show only 3 of 7 map updates performed (update 1, 4 and 7). The left and right columns of these figures correspond to the STM and LTM map versions respectively. Figure 5.4 and 5.5 show the node 5 and 24 of the map at level P0. Both figures show the complete map: the LTM laser scan registered using the estimated pose of the mobile robot, and the STM and LTM vertical edge features. Observing the laser scans only, the accuracy of the local maps increased over the map updates. For instance, despite the fact that various map updates were performed, the straightness of the walls remains consistent. From the vertical edge features point of view, Figure 5.4a and 5.5a show a clear evidence of the difference between the STM and LTM features, which can be observed comparing the uncertainty ellipses draw on the 2D plane of the laser scan. At the end of the map updates, the more stable vertical edges are shown, and they are consistent with the appearance of the environment. Figure 5.4b and Figure 5.5b give an idea of the environmental appearance at the moment of capturing the corresponding LTM scans and features.

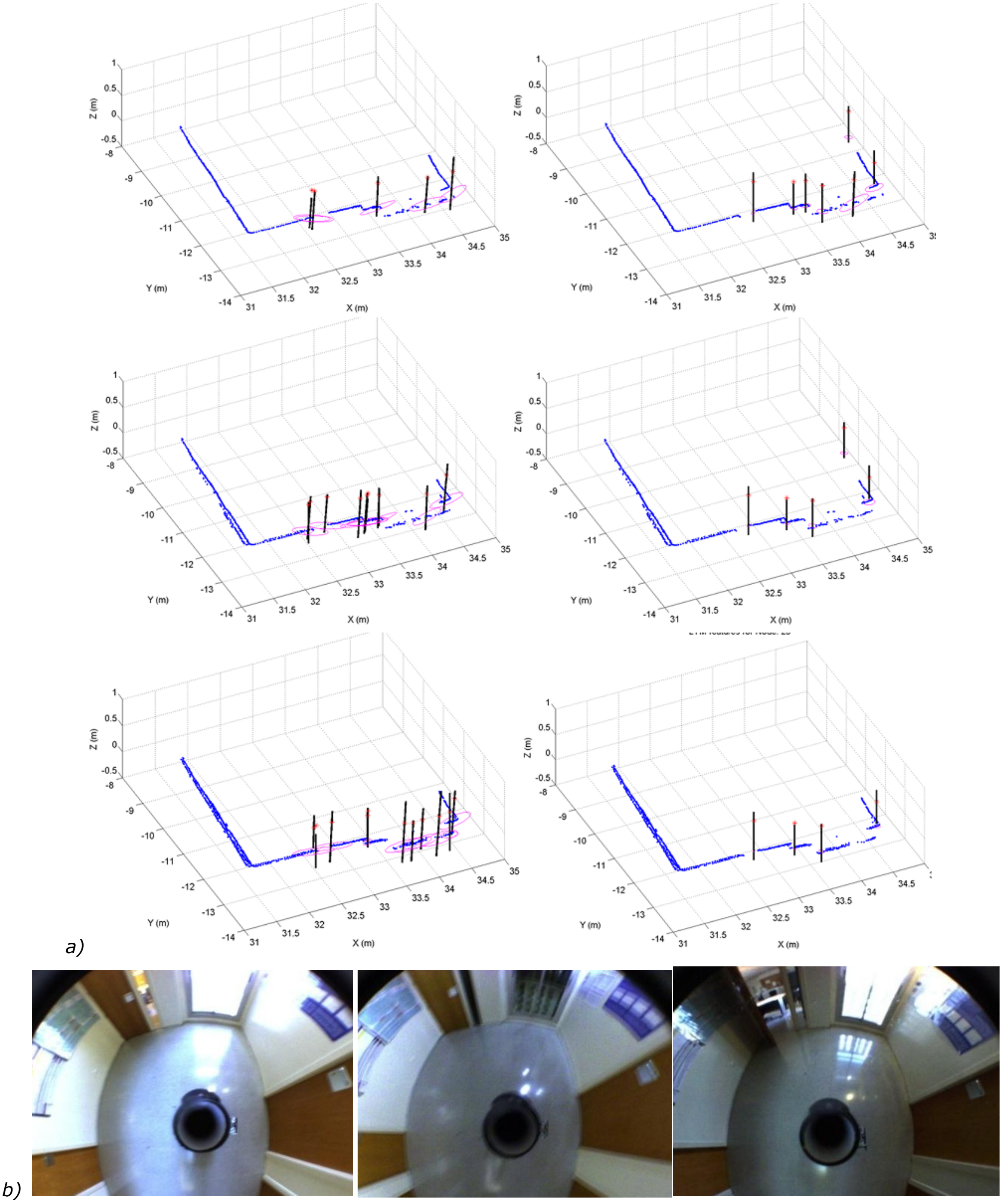


Figure 5.6 a) STM and LTM map versions of node 23 at level P1. Each row corresponds to one different map update. The left and right columns show the STM and LTM map respectively (laser scans and features). b) Omnidirectional images corresponding to each row in part a).

Figure 5.6 show the node 23 of the map at level P1. The environment structure of this level is very similar to the level P2, however the node 23 of this level is challenging because the mobile robot faces a small fence outside a window, which not only causes changes in illumination but spurious laser reflections. Observing the evolution of the LTM map (laser scan and vertical edge features) over the map updates, it is clear that the local map accuracy is improved. The STM map of the last update also shows many vertical edge candidates with high uncertainties, which is the effect of the illumination changes through the window and the spurious laser reflections. Nevertheless, these features disappear in the LTM map version.

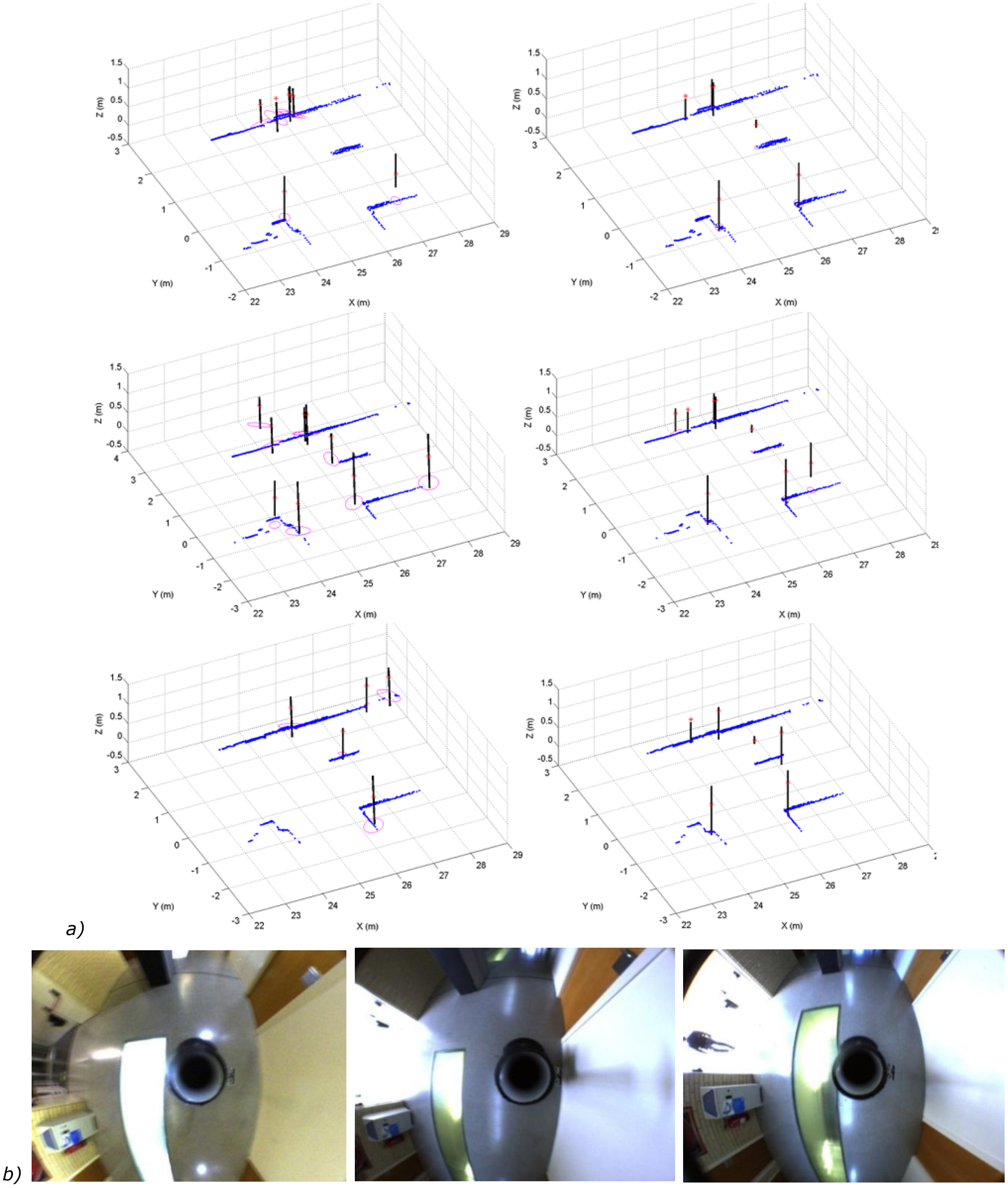


Figure 5.7 a) STM and LTM map versions of node 12 at level P2. Each row corresponds to one different map update. The left and right columns show the STM and LTM map respectively (laser scans and features). b) Omnidirectional images corresponding to each row in part a).

Figure 5.6b shows three omnidirectional images corresponding to each row of Figure 5.6a. These images show the drastic changes in illumination caused by the window in front of the robot.

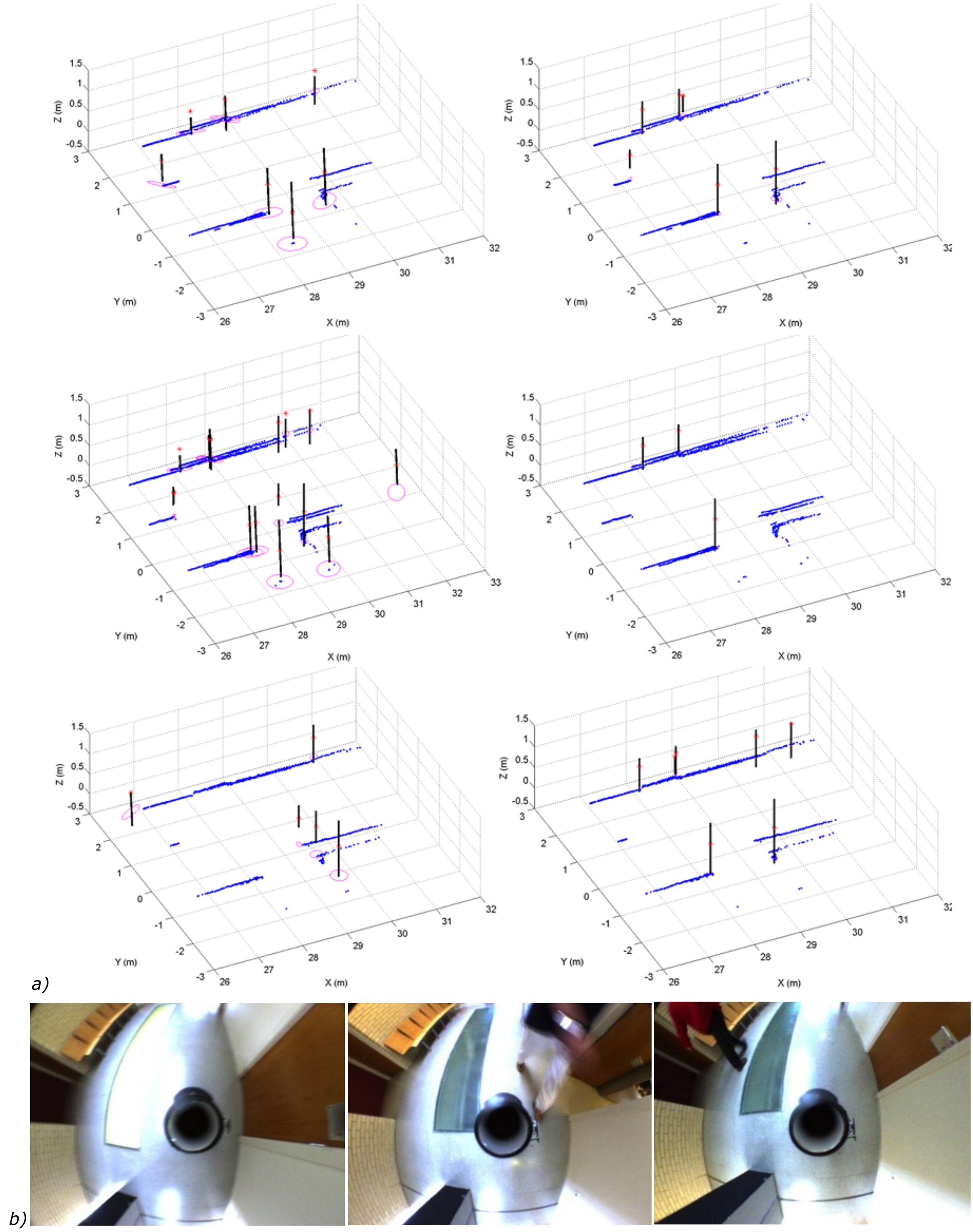


Figure 5.8 a) STM and LTM map versions of node 31 at level P2. Each row corresponds to one different map update. The left and right columns show the STM and LTM map respectively (laser scans and features). b) Omnidirectional images corresponding to each row in part a).

Figure 5.7 and 5.8 show the STM and LTM maps of nodes 12 and 31 of the level P2. This map is the largest of all three maps. Despite the fact that in these nodes big changes of illumination and pedestrians are present due the door with a big window placed across the corridor, the evolution of the LTM map shows a laser scan and a set of vertical edge features consistent with scene. Figure 5.7b shows three omnidirectional images that give an idea of the environmental conditions, particularly the changes in illumination due the big window in the left.

Observing the second row of Figure 5.8a and its corresponding omnidirectional image in Figure 5.8b, there is a notable change in the appearance of the LTM map due to the pedestrian crossing by. However at the end the LTM map model shows the more stable LRF readings

5.5. MAP UPDATE

Other important result is the capability to update the map learned [Konolige & Bowman, 2009]. In typical indoor office environments, the state of the doors change, the furniture is moved or structural changes in the environment are made. In these situations, it is desirable to update the map accordingly. The FSH model proposed in this work embeds the map update capability into the LTM map version. As a result, the configuration of the environment is learnt over time.

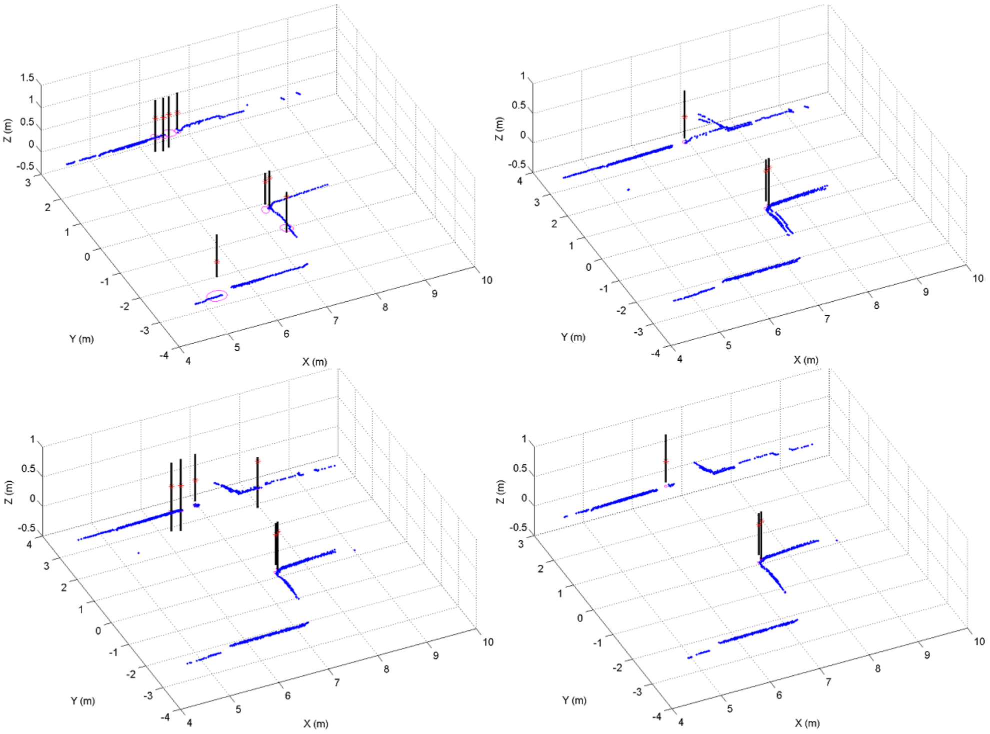


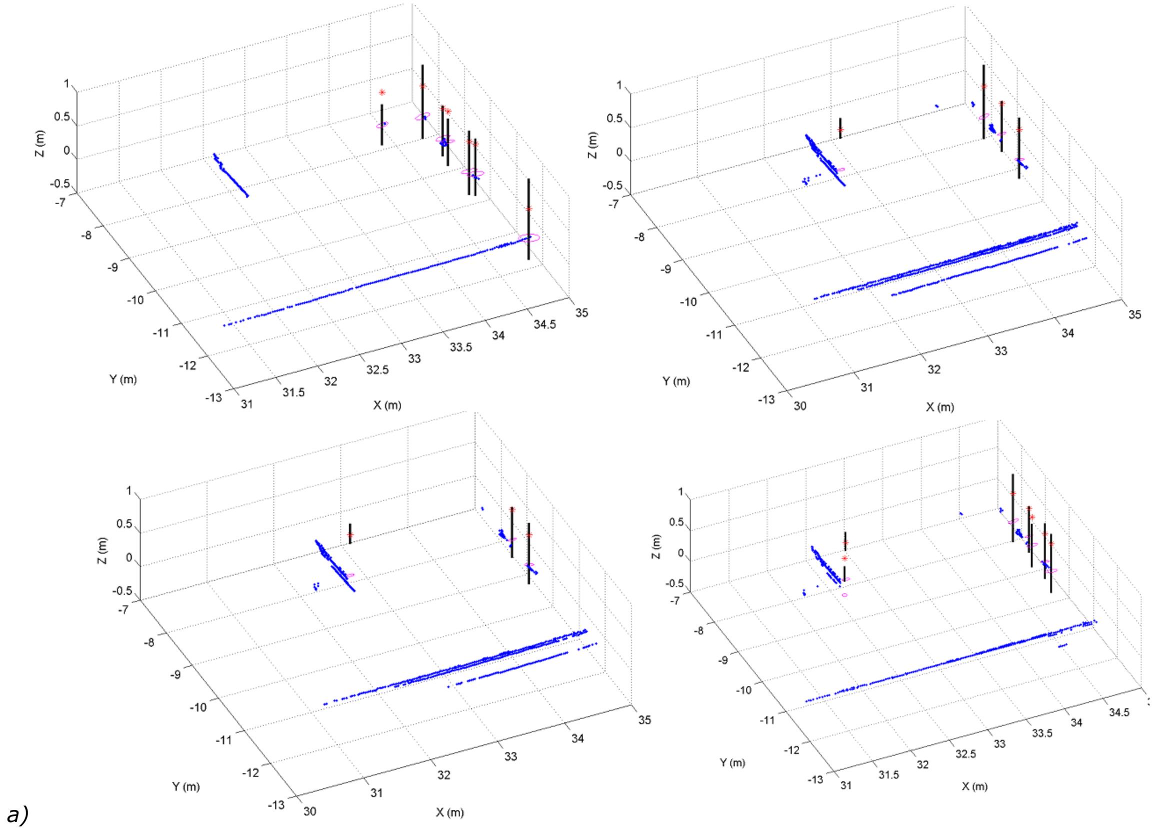
Figure 5.9 Update of the FSH model. LTM map version of node 5 at level P0 over 4 map updates. From top to bottom and left to right: update 1, 3, 5 and 7.

Figure 5.9 shows the LTM map of node 5 at level P0 over 4 map updates (top to bottom and left to right). Here, it can be observed how the LTM map is continuously modified in order to take into account the state of the door. At the beginning this door is closed, however further map updates show that the new state of the door is properly updated. In [Biber & Duckett, 2009] various time constants are considered to hold different environment configurations, then the current observations are matched with these different versions of the environment, and the map version selected is that better explains the current measurements. In this work, the FSH model holds one model of the environment namely the LTM map, which embeds the more stable appearance of the environment. This can be observed in Figure 5.9, where the top-right LTM map shows the state of the door open and closed, however this happens temporarily while the LTM map is updated properly.

In dynamic environments, where occlusions, pedestrians, changes in the furniture position and illumination changes are present, the common SLAM solutions have erroneous robot position estimation due to spurious features caused by dynamic objects, or laser scan misalignments [Bailey & Durrant-Whyte, 2006] [Kretzschmar et al., 2010]. In these cases, the map needs to be changed in order to represent not only a suitable representation of the environment, but to reflect a change of the environment structure [Konolige & Bowman, 2009].

Figure 5.10 and 5.11 show the LTM map of nodes 27 and 37 at levels P0 and P2 respectively over 4 map updates. In these figures two cases of map repair are presented. Firstly, the node 27 at level P0 is a populated entry of the building PIV, in addition the big windows provide big illumination changes; as a result, the probability of making error in the robot pose estimation increases. This can be observed in the second map update (top-right LTM map of Figure 5.10a), where the wall appears in a different position with respect to the first map update (top-left LTM map). Further map updates rehearse the last hypothesis on the LTM map, for this reason the last map update shows the wall in the new position and removes the previous hypothesis.

Figure 5.10b shows four omnidirectional images at node 27 of level P0, each one corresponding to each map updated of Figure 5.10a. As can be observed, the big window at the right and the pedestrians crossing by cause the appearance of the environment changes a lot. This change has a negative effect in the P0 level map; however using the FSH model throughout the map updates the most stable hypothesis remains.



a)

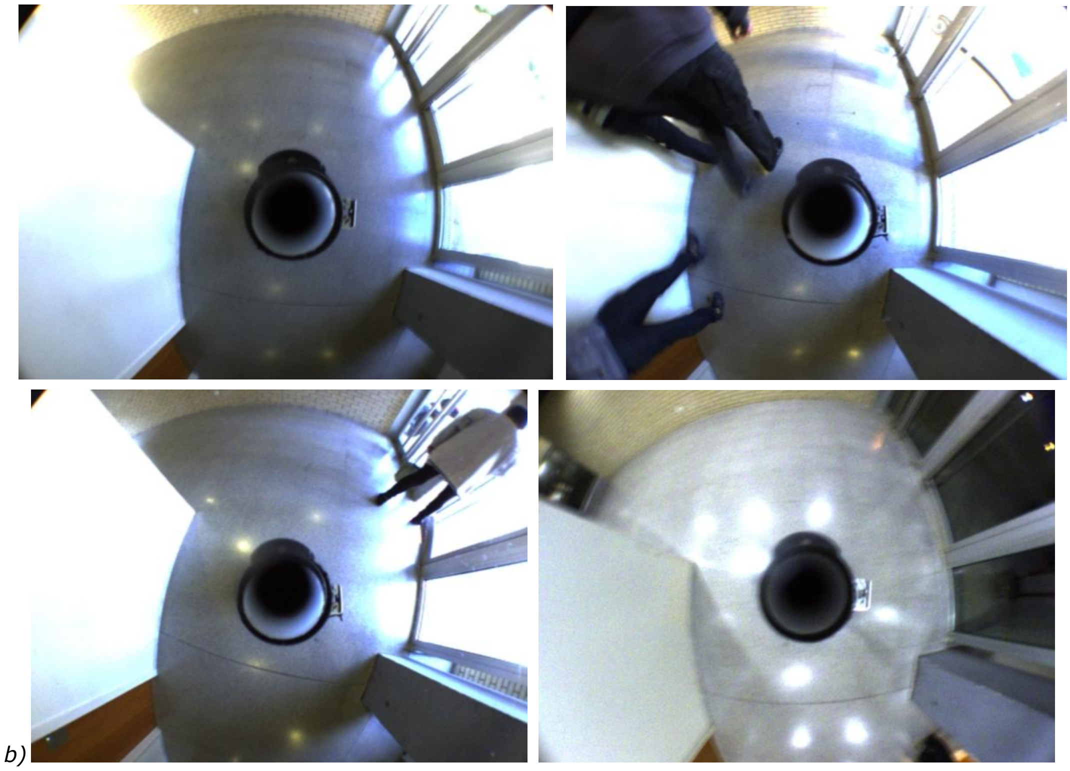


Figure 5.10 Map repair using the FSH model at node 27 of level P0 over 4 map updates. From top to bottom and left to right: update 1, 3, 5 and 7.

In the same way as Figure 5.10, Figure 5.11 shows another map repair situation occurred in node 37 at level P2. Along all the runs performed with the mobile robot, this node in particular showed many students crossing by. As a result, it can be observed that in the map updates 3 and 5 there are notable laser scan alignment errors. At the end, in the map update 7 many of these laser scan alignment errors are filtered out. In addition, in Figure 5.11 all the local map was moved to the left, which can be visually observed in the right wall in this local map. At the beginning, there was one hypothesis about the right wall position, however over the map updates this hypothesis was not rehearsed; on the contrary, moving the local map position to the left was rehearsed over the other map updates.

Map repair and update was explicitly considered in [Burgard et al., 2007], [Biber & Duckett, 2009], [Konolige & Bowman, 2009] and [Dayoub et al., 2011]. In [Burgard et al., 2007] and [Konolige & Bowman, 2009] the authors implement clustering techniques in order to represent similar and persistent the environment appearance. On the other hand, [Dayoub et al., 2011] proposes a view update using the original human memory model [Atkinson & Shiffrin, 1968]. Comparing the work presented here with respect to those approaches in [Burgard et al., 2007] and [Konolige & Bowman, 2009], this work follows and alternative solution which estimates the LTM map of the environment. The LTM map was computed applying the FSH model and it holds an integrated representation of the environment appearance. The FSH model can be used with different type of sensors, which is not clear enough in [Burgard et al., 2007] and it does not needs to maintain different configurations of the environment at different time scales as proposed by [Biber & Duckett, 2009]. Finally, the LTM map representation shows the more stable features (vertical edges in this case), and the more persistent environment configuration using a modified human memory model, improving the rehearsal method with respect to [Dayoub et al., 2011] which implements the classical human memory model has drawn criticism from psychologists and neuroscientists due to its extremely linear representation of the memory process [Baddeley, 2003], [Llinas, 2002].

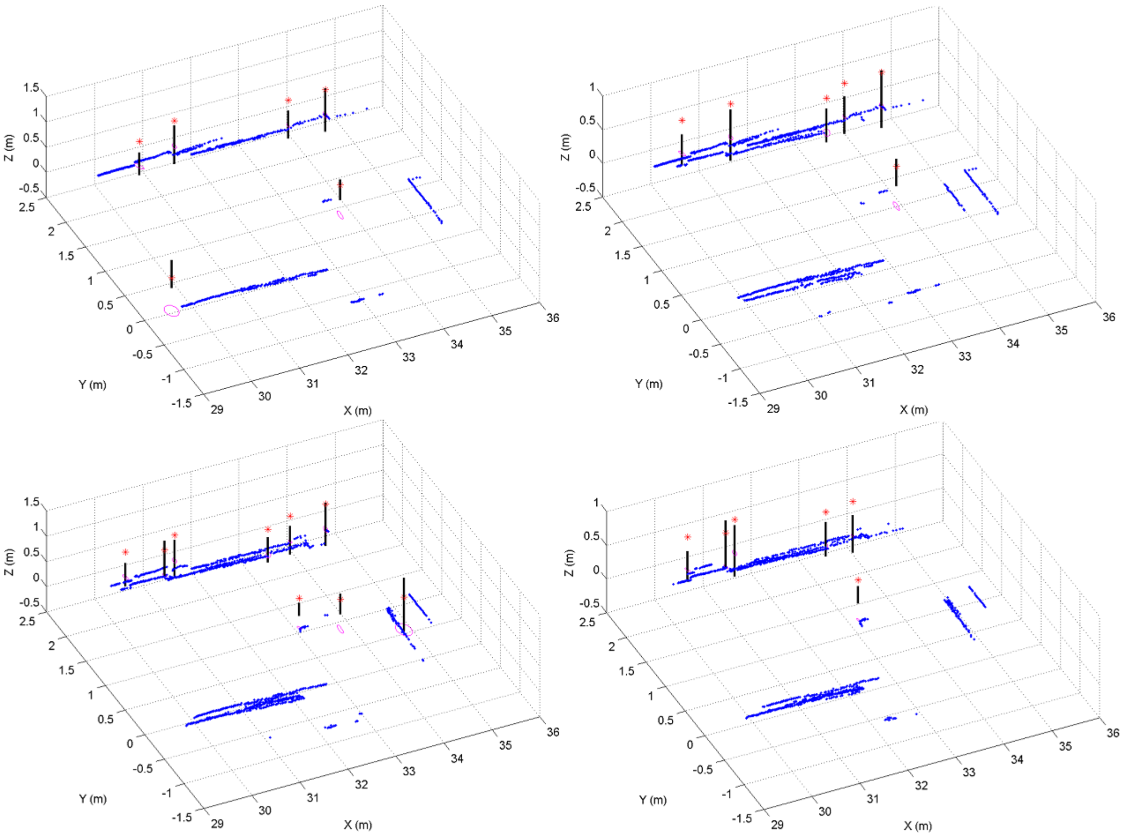


Figure 5.11 Map repair using the FSH model at node 37 of level P2 over 4 map updates. From top to bottom and left to right: update 1, 3, 5 and 7.

5.6. QUANTITATIVE RESULTS

In this section quantitative results are presented. Measuring the real performance of SLAM solutions requires ground-truth data, which in most cases is hard to obtain. Therefore, in the absence of ground-truth data, the following performance measures were selected:

1. The mean laser scan likelihood over the map updates given the robot position estimations and the current LTM map. This likelihood is computed using the Hausdorff fraction [Rucklidge, 1996], which is a metric used to measure the distance between two sets of points. Given the LTM map, the current laser scan is registered at the estimated robot position, and then using the Hausdorff fraction a measure of the scan relevance can be computed.
2. The mean number of the LTM, STM and deleted features over the map updates can provide evidence for the scalability of this work.
3. The mean matching effort over the map updates when the FSH model is used (LTM and STM features), and without using the FSH model.

5.6.1. *Scan Likelihood over Map Updates*

Figure 5.12 shows the mean laser scan likelihood for the map of level P0 (5.12a), P1 (5.12b) and P2 (5.12c) over the map updates. In both cases, it can be observed that the more map updates are performed, the highest is the scan likelihood when the FSH model is used. The scan likelihood depicted in Figure 5.12 is a measure of the localization accuracy given the computed LTM features, because over the map updates only the LTM features were used to estimate the robot position. The initial scan likelihood in both cases is not bad: 71.31% and 69.73% for the map of level P0 and P2 respectively, which is an indirect measurement of the performance of the sensor model proposed in this work (Chapter 3). However, the scan likelihood increases as more map updates are performed, meaning that the localization accuracy also increases when the FSH model is used.

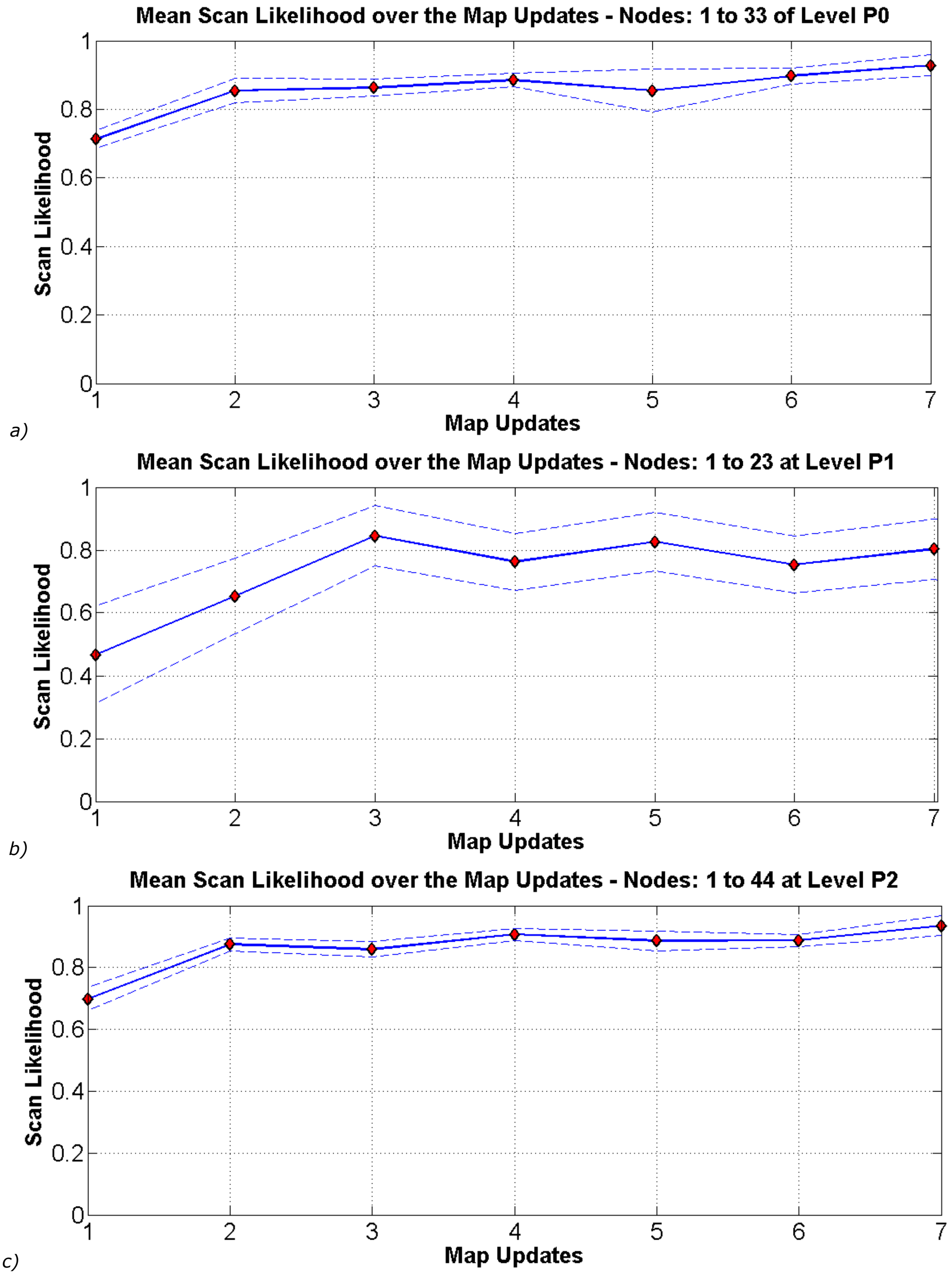
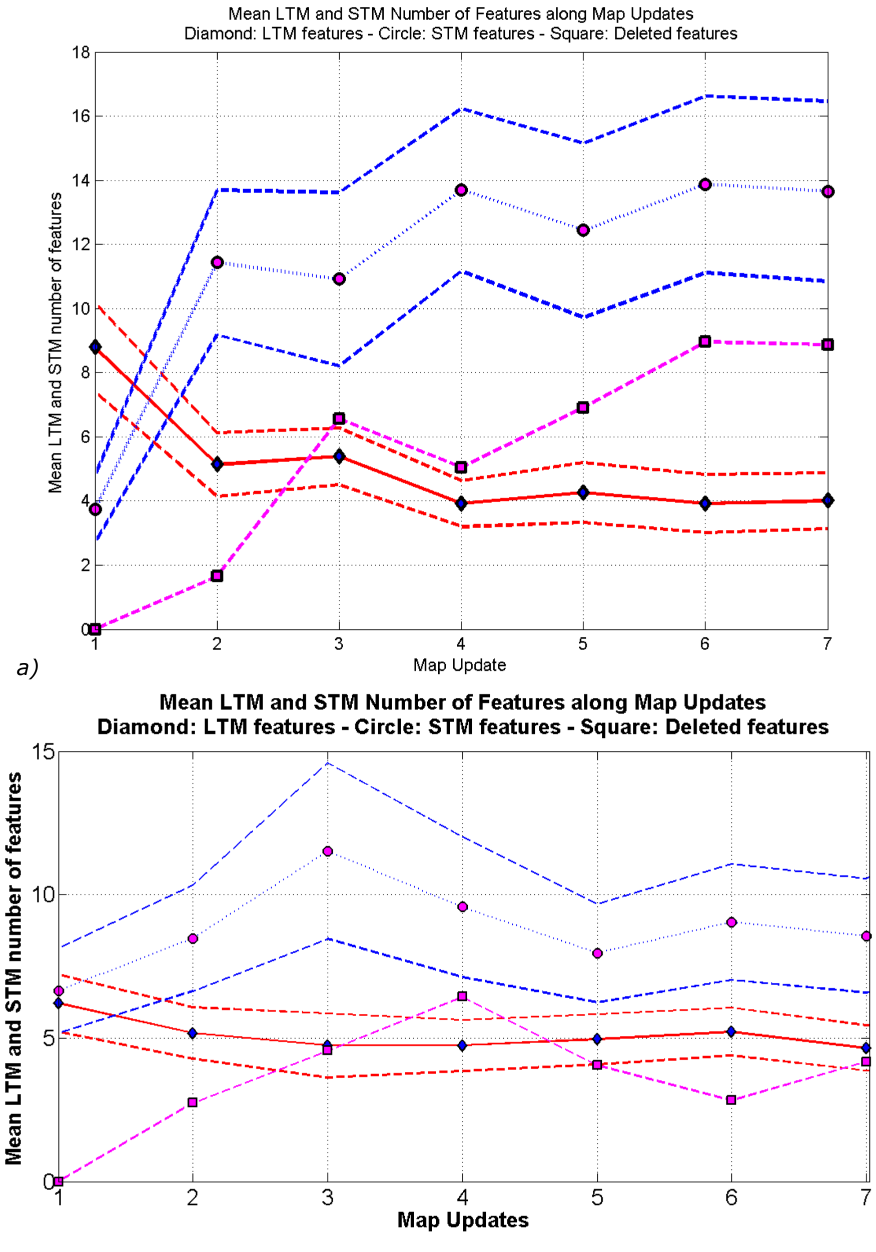


Figure 5.12 Mean scan likelihood over the map updates. a) Level P0. b) Level P1. c) Level P2.

The highest variation in the scan likelihood uncertainty is present in Figure 5.12b, which corresponds to the map at level P1. The SLAM algorithm in conjunction with the FSH model faces a big challenge here. Observing Figure 5.1f, the node 6 in the map at level P1 has not enough features, and it covers a space of about 5 meters, which is greater than the LRF maximum range. In addition, the level P1 is traversed by many pedestrians. Despite the fact that over the map update the SLAM algorithm presented slight divergences, they were not catastrophic enough to get the robot lost.

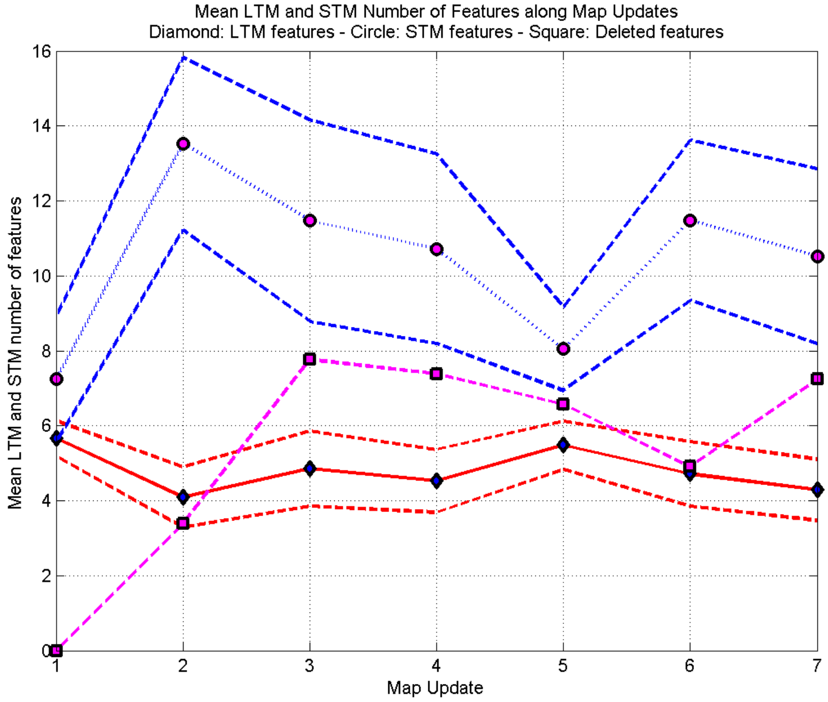
It is important noting that the increasing behavior of the scan likelihood over map updates means that the localization algorithm prefers the persistent configuration of the environment. In addition, the LTM map of the environment (including the vertical edge features and the laser scans) is changed properly as the configuration of the environment is modified over time.

5.6.2. *Scalability*



a)

b)



c)

Figure 5.13 Mean LTM, STM and deleted number of features over map updates. a) Map at level P0. b) Map at level P1. c) Map at level P2.

An important motivation behind this work is being able to deal with large environments and long-term navigation. Then, the mean number of the LTM, STM and deleted features by node provide evidence for the scalability of this work. Figure 5.13 shows the evolution of the number of LTM, STM and deleted features by node as the map updates increase. The diamond points corresponds to the evolution of the number of LTM features, the circle points are the evolution of the number of STM features and the square points are the evolution of the number of STM deleted features. The dashed curves show their corresponding uncertainty.

Observing the evolution of the LTM and STM features in both figures, it is a clear tendency of the LTM features to remain almost constant. At the beginning, the number of LTM features is greater than the number of LTM features in the other map updates. An explanation of this behavior is that at the very beginning most of the features are considered as LTM, however as the map update increases the number of LTM features tends to decrease because the more stable vertical edges remains. On the other hand, the number of STM features is greater than the LTM features as expected, but thanks to the pruning method discussed in Section 4.4.2 the number of STM features does not increases boundless.

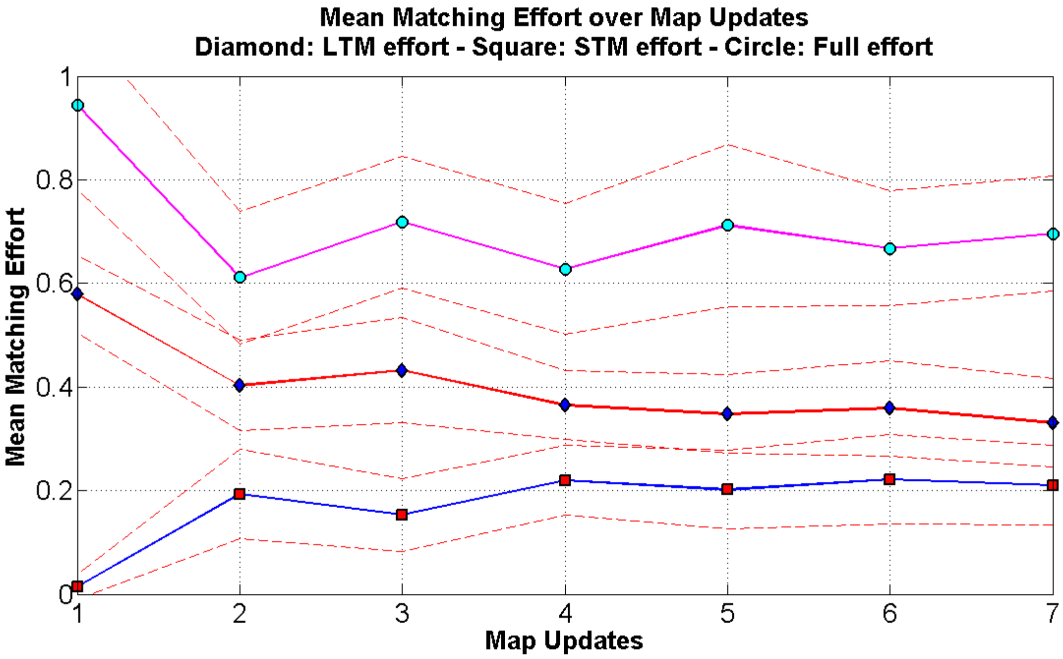
Therefore, this means that the FSH model deals well with large environments, because LTM features are only used for robot mapping and localization, and useless or old STM features are deleted properly.

5.6.3. *Matching Effort*

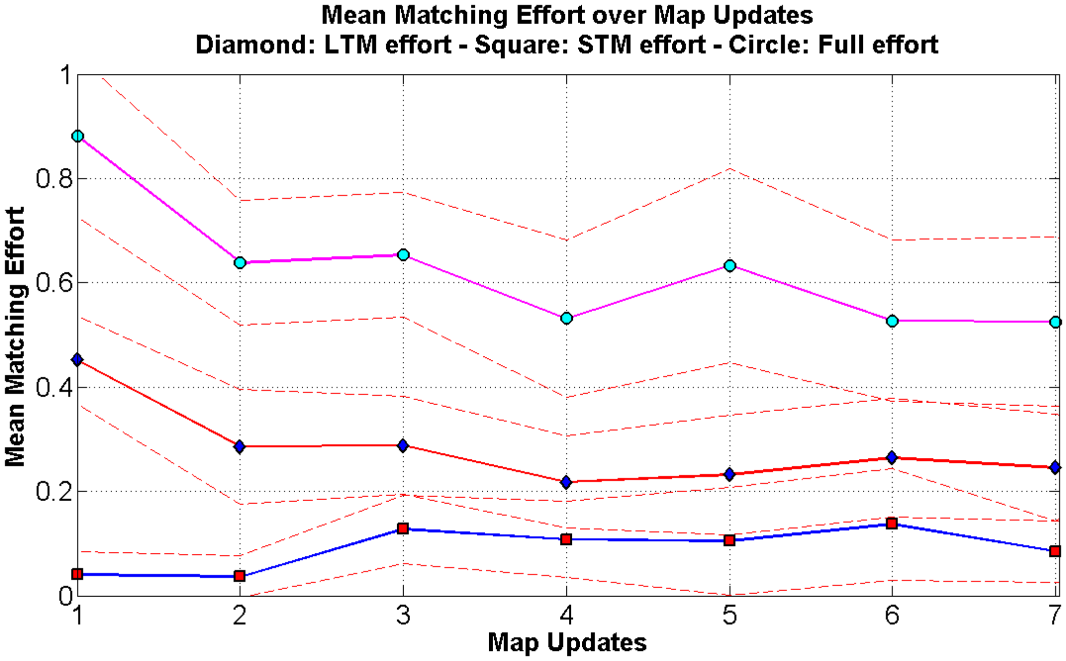
Classifying the environmental features as STM or LTM has another interesting result; it reduces the mean matching effort comparing it with respect to the full matching effort. Figure 5.15 shows the mean matching effort for the map at level P0, P1 and P2 over updates when the FSH model was used, and without using it. In this figure, diamond points correspond to the LTM matching effort over updates, square points are the STM matching effort and circle points are the full matching effort (without using the FSH model).

Observing Figure 5.15, the LTM matching effort is greater than the STM matching effort because the more stable the features (LTM features) are the more likely are found compared with the STM features. This result in conjunction with the scan likelihood depicted in Figure 5.12 shows evidence that the FSH model proposed in this work responds to the changes in the environment, and also the FSH model includes these changes to increase the localization accuracy.

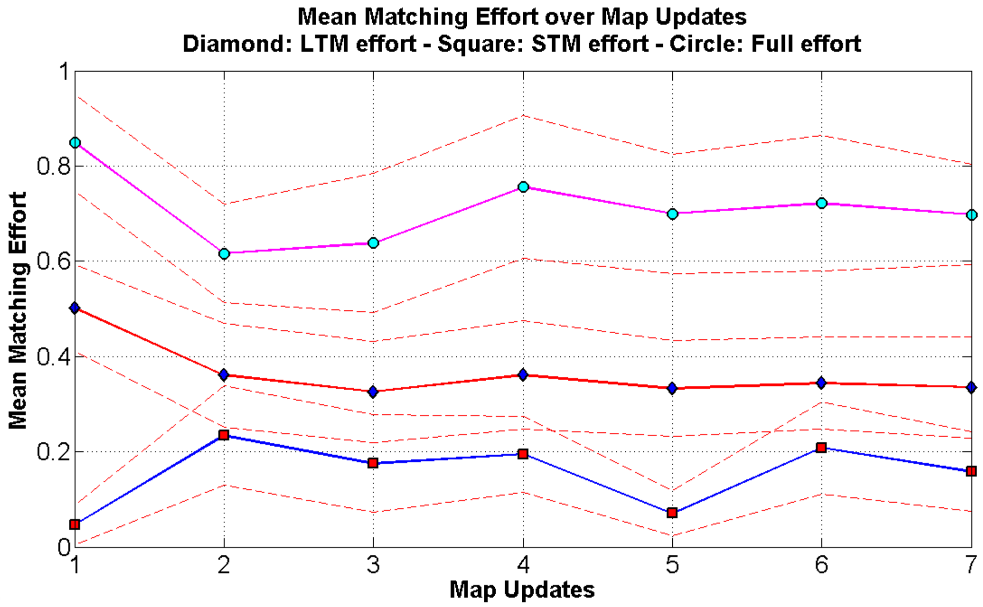
The full matching effort was measure without using the FSH model, it means using all the features available, without classifying them as STM or LTM, and without pruning the useless or old STM features. Figure 5.15a and 5.15b show the full matching effort is greater than the LTM and STM matching effort. As a result, reducing the number of matching candidates also reduces the data association effort for long-term runs, and increases the robustness in dynamic environments reducing the effect of outliers increasing the robot position estimation error.



a)



b)



b)

Figure 5.14 Matching effort over map updates with the FSH model (LTM and STM features, diamond and square points respectively) and without the FSH model (full effort, circle points). a) Map of level P0. b) Map of level P1. c) Map of level P2.

5.7. DISCUSSION

In this chapter a set of three real world and long run experiments were performed. The experiments were conducted in the levels P0, P1 and P2 of building PIV at the University of Girona. The SLAM solution used was the FastSLAM algorithm, which was modified to include the sensor model and the FSH model proposed in this work. Unlike of the experiments conducted in Chapter 4, where a visual topological map was used, in this Chapter a metric map and a stochastic topological map were implemented.

The first part of this Chapter was devoted to the qualitative results obtained from performing seven updates to the three maps of levels P0, P1 and P2. Firstly, using the FSH model the dynamic objects as pedestrians crossing by were filtered out avoiding robot position errors estimation, and then reducing the alignment errors, which tends to generate thick and blurred walls. Dynamic objects also cause spurious features, however they are filtered out and they do not appear in the LTM map which is used for robot localization. Secondly, the map quality increases over time, because the more stable features are continuously rehearsed and the configuration of the environment is updated in the LTM map. As a result, the more updates are performed, the more consistent are the vertical edges with the appearance of the environment when the FSH model is used. Last, updating the map accordingly to the changes observed in the environment was also demonstrated. The LTM map of the environment embeds the changes of the environment, which means that not only the configuration of the environment is learned, but also correct a local map representation after robot position error estimation has took place.

The second part of this Chapter was devoted to describe the quantitative results, which in absence of a ground-truth three performance measures were carried out: the mean laser scan likelihood over the map updates given the robot position estimations and the current LTM map; the mean number of LTM, STM and deleted features over map updates; and the mean matching effort over map updates with and without the FSH model. Firstly, the laser scan likelihood over updates has shown an increasing behavior, meaning that the localization accuracy also increases when the FSH model was used. In addition, as the environment changes, the LTM map of the environment change accordingly. Secondly, the mean number of the LTM, STM and deleted features provide evidence for the scalability of this work; in this way, the results reported in this Chapter demonstrates that the FSH model integrated in a SLAM solution deals well with large environments because LTM features are only used for robot mapping and localization, and useless or old STM features are deleted properly. Last, the LTM, STM and full matching efforts provide evidence of reducing the effect of outliers increasing the robot position estimation error, and also taking into account the scan likelihood results the localization accuracy is increased when the FSH model is used, because it incorporates the changes of the environment into the LTM map.

The FSH model has the main advantage of incorporating the changes of the environment in terms of its configuration and the features extracted from it, and incrementally increasing the map quality. This result arises from the fact that rating the map landmark allows classifying them as STM or LTM in the context of the modified human memory model [Bacca et al., 2011] [Atkinson & Shiffrin, 1968]. As a result, the localization accuracy is increased in further runs, and the data association effort is reduced thanks to the map landmark classification approach into STM and LTM, as well as the pruning method applied to the useless and old STM features. Therefore, integrating the FSH model into a SLAM solution improves their behavior in long-term navigation.

CHAPTER 7

6. *CONCLUSIONS*

CONTENTS

In this Chapter the conclusions and future perspectives of this work are presented. To start with, Section 7.1 summarizes the content presented in each chapter. Afterwards, in Section 7.2 the scientific contributions extracted from the proposals and experiments carried out are discussed. Furthermore, in Section 7.3 the short-term and long-term future perspectives are discussed as well as some interesting future research issues. Finally, in Section 7.4 the list of publications related with this work is presented and the scientific collaborations performed.

6.1. CONCLUSIONS

This thesis addresses the problem of updating the robot internal representation of the environment when its appearance changes over time. The real-world environments are complex and dynamic, and the robot must adapt to the new working conditions and fulfill its task. To do so, on-board sensors and their associated model are the information used to estimate the robot state and the map of the environment. The SLAM algorithms consider all the map features equally important, but in long-term operation there is the opportunity to classify the map features according to the number of times a feature is re-observed. However, many solutions of the SLAM problem assume a static environment representation, such that moving objects or in general the change of the environment is considered as noise or outliers that should be rejected by the SLAM algorithm [Durrant-Whyte & Bailey, 2006] [Bailey & Durrant-Whyte, 2006]. A brief revision of these techniques was performed in Chapter 2. Taking into account long-term mapping and localization, dealing with dynamic environments and the current SLAM solutions, the second part of Chapter presents a survey of long-term SLAM methods. This survey classifies the long-term SLAM approaches into six classes describing the different methods to deal with dynamic environments namely: Memory management models, Landmark visibility and rating, Detecting/using dynamic objects, Dynamic changes integrated in SLAM, Pruning methods, and Multiple map representations. Based on this survey, the main goal for this section was that it is worth designing and implementing long-term SLAM methods in a way that it can be applied to the current SLAM solutions. Afterwards, a brief review of different environment modeling techniques in the field of mobile robotics was presented. It included: range data features, image features extracted from standard and omnidirectional cameras. The main outcome of this Chapter was the finding to focus effort on combining the 2D LRF and the omnidirectional vision sensors in order to extract salient features from the environment.

One of the contributions presented in this work was presented in Chapter 3. In this Chapter the sensor model based on a range-augmented omnidirectional sensor was presented to extract vertical edge features from indoors scenes. Firstly, in order to reduce the range measurements of the 2D LRF, the LRF calibration procedure was described and the LRF linear model was obtained. Secondly, different LRF features are introduced namely: LRF breakpoints and LRF line segments. Thirdly, a brief review of the omnidirectional vision features was presented. Since, intuitively the appearance-based model of the environment describes the environment as it is, and taking advantage of its natural structure, the omnidirectional camera was used to extract vertical edges, which afterwards was combined with the LRF data projected onto the omnidirectional image to estimate the edge position. Fourthly, a detailed description of how embedding range information in omnidirectional images using the extrinsic calibration between the LRF and omnidirectional camera was presented. Here, two methods were experimentally evaluated: simultaneously and non-simultaneously parameter estimation; however, the simultaneous parameter estimation is more likely to get a successful calibration result, because it does not assume any alignment constraint between the LRF and the omnidirectional camera, and it considers the intrinsic coupling between rotation and translation. Fifthly, merging the previous work related with the extrinsic calibration of the LRF and the omnidirectional camera, and the vertical edges detection, the sensor model to extract the 3D position of vertical edges in indoor environments was presented. Considering the data set includes real-world conditions, the dynamic objects of the environment and the robot motion robust vertical edge position estimation with respect to the mobile robot was described. This included the projection of the breakpoint uncertainties using the catadioptric projection model onto the omnidirectional image, the projection of the laser segments as conics onto the omnidirectional image using the model proposed by [Barreto & Araujo, 2005], computing the intersects between the vertical edges and the conic representation of the LRF segments, and the data association between the LRF breakpoints and the set of intersects using the JCBB test [Neira & Tardos, 2001]. Sixth, the map landmark data association was based on the JCBB test, however depending on the uncertainty this method could fails. In Chapter 3 was also described a complementary method for data association, which depends on geometric constraints based on the unified projection model for catadioptric cameras [Geyer & Daniilidis, 2000]. The main outcome of this Chapter is the validation of the sensor model proposed in a set of real-world experiments, which in absence of a ground-truth they were compared with respect to the G^{2}O framework, obtaining promising results in a very challenging environment, where illumination changes and occlusions by pedestrians were part of the data set used.

The main contribution of this work was presented in Chapter 4. The Feature Stability Histogram (FSH) is a solution proposed in this work to deal with changing environments and long-term mapping and localization. The main idea behind this is classify the features of the environment in stable and non-stable ones. To do so, the FSH is inspired on the human memory model proposed by [Atkinson & Shiffrin, 1968], in order to sequentially build a histogram of the feature strengths which is updated once the feature is re-observed. Stable features, belonging to the Long-Term Memory (LTM), are used for localization and mapping. On the other hand, non-stable features, belonging to the Short-Term memory (STM), can be part of the LTM depending of their strength. This Chapter first presents a description of the original human memory model, and then the modified human memory model used in this work. The main modifications include: the feature classification (STM or LTM) is not linear since the rehearsal process takes into account the feature strength, and the reference view is composed of both memories the STM and LTM. Afterwards, a description of how the FSH model can be integrated into typical SLAM solutions was presented as well as the probabilistic foundations of this integration process. Furthermore, the LTM/STM feature classification was discussed, which was done using a dynamic threshold that acts on the feature strengths. Taking advantage of the feature rating and how it was used to build the FSH, the LTM/STM feature classification problem was solved using *k-means* [Lloyd, 1982]. STM feature removal was also described considering the STM feature strengths likelihood and how old a STM feature is. Experimental results using static LRF-based and vision-based experiments were performed. Results also included topological map building and localization, where global and local localization with and without noise and artificial occlusion were conducted. The main outcome of this Chapter was that the FSH model map representation is able to maintain the representativeness of the environmental appearance despite the environmental changes.

Chapter 5 presented long-term SLAM results using the FSH model integrated into the FastSLAM algorithm, which is a different sort SLAM solution compared with the visual-based topological map building and localization implemented in Chapter 4. Firstly, the motion model of the Pioneer 3DX mobile robot was described as well as the probabilistic foundations of the FastSLAM algorithm. The results were divided in qualitative and quantitative. The former includes the capability of filtering dynamic objects, visually observe map quality over updates and map update. The quantitative results included: the scan likelihood given the estimated robot position over the map updates; evidence of the system scalability observing the evolution of the LTM, STM and deleted features over updates; and a comparison of the matching effort with and without the FSH model. The main outcomes of this Chapter are: the FSH model was able to filter out spurious laser scans and features present in the environment due dynamic objects; the accuracy of the local maps (LTM scans and vertical edges) increased over the map updates when the FSH model was used; the FSH model holds one model of the environment namely the LTM map and it embeds the more stable appearance of the environment, unlike the approaches of [Biber & Duckett, 2009], [Burgard et al., 2007] and [Konolige & Bowman, 2009] who follow a different way maintaining multiple map representations; the scan likelihood increases as more map updates are performed, meaning that the localization accuracy also increases when the FSH model is used; the FSH model deals well with large environments, because LTM features are only used for robot mapping and localization, and useless or old STM features are deleted properly; and reducing the number of matching candidates also reduces the data association effort in long-term runs, and increases the robustness in dynamic environments reducing the effect of outliers in increasing the robot position estimation error.

6.2. CONTRIBUTIONS

Considering the results condensed in this thesis, the goal of using the FSH model for appearance-based simultaneous mapping and localization in long-term navigation has been accomplished. In the development of this goal, the following research contributions are listed below:

- A detailed extrinsic calibration between a 2D LRF with invisible trace and an omnidirectional camera. When this research work was started, there were few works on this topic with the sufficient detail to be replicated. In [Bacca et al., 2010] two approaches were studied simultaneous and non-simultaneous parameter estimation, as well as a method to obtain a ground truth using the calibration information of the omnidirectional camera to validate the results obtained.
- The FSH model, an innovative feature management strategy based on the human memory model [Atkinson & Shiffrin, 1968], and modified in order to avoid the extremely linear representation of the original memory process. In [Bacca et al., 2010] and [Bacca et al., 2011] the FSH model proposed takes into account levels of memory (STM and LTM) represented in the features strength.
- A sensor model based on a range-augmented omnidirectional vision sensor and used to solve the SLAM problem. The sensor model presented is based on the extrinsic calibration of a LRF and an omnidirectional camera. Using this calibration, 3D position of vertical edges were extracted and considered as observations into the FastSLAM algorithm [Bacca Cortes et al., 2011].
- An indoor dataset collected at different time of the day and between seasons including LRF readings, the corresponding omnidirectional images and the robot odometry. The datasets available online contains one run of different places which is not useful for long-term mapping and localization. Since, it requires data recorded under various illumination conditions, over several days and presence of dynamic obstacles. Recently, the COLD database [Pronobis & Caputo, 2009] includes data recorded in the conditions described above.
- The approach presented in this document was validated using real-world experiments over a long period of time, and within different environmental conditions. The FSH model was also tested using different kind of mapping and localization algorithms namely topological visual-based and the particle filter based FastSLAM algorithm. In this way, it showed that the FSH model can be adapted to different types of SLAM solutions.

6.3. FUTURE WORK

During the research work conducted over these years, new interesting problems and research topics have been found. The following list consolidates some of them:

- **Sensor Model**
  - The vertical edge features suffer the same disadvantages of the edge detection methods; they highly depend on the illumination conditions and the performance of the edge detector. The author believes that combining the conic projection of the LRF segments and the vertical edges that intersect it, plane detection on the omnidirectional image can be implemented simplifying the plan detection proposed in [Mei, 2007].
  - The sensor model proposed in this work requires a second step to detect the vertical edges discarding the scene non-vertical edges. It

    - would very interesting find a way to avoid complete line detection of the scene in order to reduce the computation time.
    - The current configuration of the environment is based on the LTM laser scans. It would be interesting to implement a LTM laser segment model as the configuration of the environment. Since, in this way a better metric than the nearest-neighbor could be implemented to rate the laser features.
    - It would very useful to perform the extrinsic calibration between a 3D LRF and an omnidirectional camera, and then applying the FSH model on the textured planes extracted. This enables the robot position estimation in 6-DOF instead of the 3-DOF which actually is implemented.
- **The FSH model**
    - The approach presented in this work based the landmark rating either in the feature uncertainty or the matching distance. However, the feature uncertainty suffers from the well known disadvantage of obtaining an overconfident uncertainty as the map is continuously updated; and the matching distance is a very local measure. Therefore, other landmark rating techniques are required. For instance, it would be interesting to fuse the feature visual appearance and its metric information.
    - Another interesting point arises from the LTM/STM classification method and the STM feature removal. Here, artificial intelligence techniques as Support Vector Machines (SVM), fuzzy logic or neural networks are interesting options to explore.
- **SLAM solutions**
    - It would be interesting to integrate graph simplification techniques as that proposed in [Kretzschmar et al., 2010] with the FSH model as a measure of what observations to discard.
    - It would be a good idea to integrate the FSH model into more SLAM solutions such as: Divide and Conquer [Paz et al., 2008], Conditionally independent maps [Pinies & Tardos, 2008], Hierarchical SLAM [Estrada et al., 2005], Selective map joining [Aulinas et al., 2010], GraphSLAM [Thrun & Montemerlo, 2005], ESEIF [Walter et al., 2007] and Occupancy Grid Mapping [Thrun et al., 2005].
    - It would very useful to conduct a real-time experiment on a vehicle with an on-board computer and using an optimized code in C++.
- **New scenarios**
    - Conducting experiments in outdoors scenarios and without the 3-DOF constraint would be an interesting option to explore. However, the sensing devices have to be improved: a camera with faster white-balance and gain adjust, and a LRF with better performance in terms of range and sensitivity.

6.4. PUBLICATIONS AND SCIENTIFIC COLLABORATIONS

6.4.1. Journal Papers

1. Bladimir Bacca, Joaquim Salvi, Xavier Cufi. “Vertical Edge-Based Mapping Using Range-Augmented Omnidirectional Vision Sensor”. *IET-Computer Vision,* on evaluation, 2012.

2. Bladimir Bacca, Joaquim Salvi, Xavier Cufi. "Appearance-Based Mapping and Localization for Mobile Robots using a Feature Stability Histogram". *Robotics and Autonomous Systems* 59(10), pp 840–857, 2011.
3. B. Bacca, J. Salvi, J. Batlle, X. Cufi. "Appearance-Based Mapping and Localization Using Feature Stability Histograms". *Electronic Letters* 46(16), 1120-1121, 2010.

6.4.2. *Conference Papers*

1. Bladimir Bacca, Joaquim Salvi, Xavier Cufi. "Indoor SLAM using a Range-Augmented Omnidirectional Vision Sensor". **OMNIVIS 2011**, *The 11th Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras, in conjunction with **ICCV 2011** 13th International Conference on Computer Vision*, Barcelona (Spain), November 6-13, 2011.

2. Bladimir Bacca, Joaquim Salvi, Xavier Cufi. "Probabilistic Appearance-based Mapping and Localization using the Feature Stability Histogram". Fourteenth International Conference of the Catalan Association of Artificial Intelligence **CCIA2011**. *Frontiers in Artificial Intelligence and Applications*, Lleida (Spain), October 26-28, 2011.
3. Bacca, E.B. and Mouaddib, E. and Cuffi, X., "Embedding Range Information on Omnidirectional Images through Laser Range Finder", IEEE/RSJ *Int. Conf. on Intelligent Robots and Systems, **IROS'10***, Taipei, Taiwan, October, 2010.
4. B. Bacca, J. Salvi, X. Cufi. "Mapping and Localization for Mobile Robots through Environment Appearance Update". Proceeding of the 2010 conference on Artificial Intelligence Research and Development: Proceedings of the 13th International Conference of the Catalan Association for Artificial Intelligence **CCIA2010**. *Frontiers in Artificial Intelligence and Applications*, Volume 220, Pages 291-300, L'Espluga de Francolí (Spain), Octuber 20-21 2010.
5. Bacca, E.B. and Mouaddib, E. and Cuffi, X., "Range information in omnidirectional images through laser range finder", *The 10th Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras*, **OMNIVIS 2010**, in conjunction with **RSS2010**, Zaragoza, Spain, June, 2010.
6. B. Bacca, J. Salvi, X. Cufi. "Appearance-based SLAM for mobile robots". Proceeding of the 2009 conference on Artificial Intelligence Research and Development: Proceedings of the 12th International Conference of the Catalan Association for Artificial Intelligence **CCIA2009**. *Frontiers in Artificial Intelligence and Applications*; Vol. 202, pp 55-64, 2009.

6.4.3. *Scientific Collaborations*

Most part of this thesis was made at VICOROB at the University of Girona (2008-2012). However, a total of three months were spent at the Université de Picardie Jules Verne (Amiens, France), in the Laboratoire MIS (Modélisation, Information et Systèmes) for a research stay. The research work performed there was supervised by Prof. Dr. El Mustapha Mouaddib.

APPENDIX

7. APPENDIX A – JACOBIANS TO ESTIMATE THE CORNER UNCERTAINTY ON THE IMAGE PLANE

$$\boldsymbol{J_p} = \begin{bmatrix} \cos(\theta_{ci}) & -\sin(\theta_{ci}) & 0 \\ \sin(\theta_{ci}) & \cos(\theta_{ci}) & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \text{A.1}$$

$$\boldsymbol{J_R} = \begin{bmatrix} (r_7 + r_9)y + (r_8 - r_3 s\theta)z & -s\theta c\psi x + c\theta r_4 y + s\phi r_6 z & -r_1 x - (r_2 + r_3)y + (r_4 - r_5)z \\ (-r_4 + r_5)y - (r_2 + r_3)z & -s\theta s\psi x + r_1 s\phi y + r_1 c\phi z & r_6 x + (-r_8 + r_4 s\theta)y + (r_7 + r_9)z \\ c\phi c\theta y - s\phi c\theta z & -c\theta x - s\theta s\phi y - c\phi s\theta z & 0 \end{bmatrix}$$

A.2

$r_1 = c\theta s\psi$

$r_2 = c\phi c\psi$

$r_3 = s\psi s\phi s\theta$

$r_4 = s\phi c\psi$

$r_5 = c\phi s\theta s\psi$

$r_6 = c\theta c\psi$

$r_7 = s\phi s\psi$

$r_8 = c\phi s\psi$

$r_9 = c\phi s\theta c\psi$ *A.3*

$$\boldsymbol{J_S} = \frac{1}{r_p(z_s + \xi r_p)^2} \begin{bmatrix} r_p z_s + \xi(y_s^2 + z_s^2) & -\xi x_s y_s & -x_s(r_p + \xi z_s) \\ -\xi x_s y_s & r_p z_s + \xi(x_s^2 + z_s^2) & -y_s(r_p + \xi z_s) \end{bmatrix} \qquad \text{A.4}$$

$$\boldsymbol{J_D} = \begin{bmatrix} J_{D1} & L \\ L & J_{D2} \end{bmatrix} \qquad \text{A.5}$$

$$J_{D1} = 1 + k_1(r_u + 2x_u^2) + k_2 r_u(r_u + 4x_u^2) + 2yk_3 + 6k_4 x + k_5 r_u^2(r_u + 6x_u^2) \qquad \text{A.6}$$

$$J_{D2} = 1 + k_1(r_u + 2y_u^2) + k_2 r_u(r_u + 4y_u^2) + 6yk_3 + 2k_4 x + k_5 r_u^2(r_u + 6y_u^2) \qquad \text{A.7}$$

$$L = 2x_u y_u k_1 + 4x_u y_u k_2 r_u + 2x_u k_3 + 2y_u k_4 + 6x_u y_u k_5 r_u^2 \qquad \text{A.8}$$

$$\boldsymbol{J_K} = \begin{bmatrix} \gamma_1 & s\gamma_1 \\ 0 & \gamma_2 \end{bmatrix} \qquad \text{A.9}$$

Where, in Equation A.1, $\boldsymbol{J_P}$ is the Jacobian of the polar to Cartesian coordinate transformation, and the corner point orientation is defined by θ_{Ci}. In Equation A.2, $\boldsymbol{J_R}$ is the Jacobian of the transformation to the camera frame and the roll, pitch and yaw angles are defined by ϕ, θ and ψ; *c* and *s* refer to *cos(.)* and *sin(.)* functions. In Equation A.4, $\boldsymbol{J_S}$ is the Jacobian of the projection function to the image plane at infinity, ζ is the mirror parameter, $[x_S\ y_S\ z_S]^T$ is the point on the sphere, and r_P its norm. In Equation A.5, $\boldsymbol{J_D}$ is the Jacobian of the distortion function which was taken from the calibration toolbox presented in [Mei, 2006], and $[x_u\ y_u]^T$ is the point in the image plane at infinity. In Equation A.9, $\boldsymbol{J_K}$ is the Jacobian of the camera projection function. Using Equation 3.20 and Equations A.1 to A.9, the corners uncertainties on the image plane can be defined as depicted in Equation 3.26.

8. *BIBLIOGRAPHY*

Abrate, F. et al., 2010. Map updating in dynamic environments. *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*, pp.1 -8.

AdeptTechnology, 2012. *Intelligent Mobile Robot Platforms for Service Robots, Research and Rapid Prototyping*. [Online] Available at: http://mobilerobots.com/Mobile_Robots.aspx [Accessed 2012].

Aly, M., Munich, M. & Perona, P., 2011. Indexing in large scale image collections: Scaling properties and benchmark., 2011. IEEE Computer Society.

Andrade-Cetto, J. & Sanfeliu, A., 2002. Concurrent Map Building and Localization on Indoor Dynamic Environments. *IJPRAI*, 16, pp.361-74.

Andreasson, H., Duckett, T. & Lilienthal, A.J., 2008. A Minimalistic Approach to Appearance-Based Visual SLAM. *Robotics, IEEE Transactions on*, 24, pp.991 -1001.

Andreasson, H., Treptow, A. & Duckett, T., 2007. Self-localization in non-stationary environments using omni-directional vision. *Robot. Auton. Syst.*, 55(7), pp.541--551.

Angeli, A., Doncieux, S., Meyer, J. & Filliat, D., 2008. Incremental vision-based topological SLAM. In *Intelligent Robots and Systems, IEEE/RSJ International Conference on*., 2008.

Antone, M. & Friedman, Y., 2007. Fully Automated Laser Range Calibration. In *In Proceedings of British Machine Vision Conference*., 2007.

Arican, Z. & Frossard, P., 2010. OmniSIFT: Scale invariant features in omnidirectional images. In *Image Processing (ICIP), IEEE International Conference on*., 2010.

Arras, K.O. & Siegwart, R., 1997. Feature Extraction and Scene Interpretation for Map-Based Navigation and Map Building. In *Proc. of SPIE, Mobile Robotics XII*., 1997.

Atkinson, R.C. & Shiffrin, R.M., 1968. Human memory: a proposed system and its control processes. In *The Psychology of Learning and Motivation*. New York: Academic Press. p.89–195.

Aulinas, J., LladoÌ□ and, X., Salvi, J. & Petillot, Y.R., 2010. Selective Submap Joining for underwater large scale 6-DOF SLAM. In *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*., 2010.

Bacca Cortes, B., Cufi Sole, X. & Salvi, J., 2011. Indoor SLAM using a range-augmented omnidirectional vision. In *Computer Vision Workshops (ICCV Workshops), IEEE International Conference on*., 2011.

Bacca, E.B., Mouaddib, E. & Cufi, X., 2010. Embedding range information in omnidirectional images through laser range finder. In *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*., 2010.

Bacca, B., Salvi, J., Batlle, J. & CufiÌ□ and, X., 2010. Appearance-based mapping and localisation using feature stability histograms. *Electronics Letters*, 46, pp.1120 -1121.

Bacca, B., Salvi, J. & CufÃƒÂ, X., 2011. Appearance-based mapping and localization for mobile robots using a feature stability histogram. *Robotics and Autonomous Systems*, 59, pp.840 - 857.

Bacca, B., Salvi, J. & CufÃ, X., 2009. Appearance-Based SLAM for Mobile Robots. In *Proceedings of the 12th International Conference of the Catalan Association for Artificial Intelligence*., 2009. IOS Press.

Baddeley, A., 2003. WORKING MEMORY: LOOKING BACK AND LOOKING FORWARD. *Nat Rev Neurosci*, 4, pp.829--839.

Bailey, T. & Durrant-Whyte, H., 2006. Simultaneous localization and mapping (SLAM): part II. *Robotics Automation Magazine, IEEE*, 13, pp.108 -117.

Barber Castaño, R., 2000. *Desarrollo de una arquitectura para robots móviles autónomos : aplicación a un sistema de navegación topológica*. Madrid: Universidad Carlos III de Madrid. Departamento de Ingeniería Eléctrica, Electrónica y Automática.

Barreto, J.P. & Araujo, H., 2001. Issues on the geometry of central catadioptric image formation. In *Computer Vision and Pattern Recognition, CVPR. Proceedings of the IEEE Computer Society Conference on*., 2001.

Barreto, J.P. & Araujo, H., 2005. Geometric properties of central catadioptric line images and their application in calibration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27, pp.1327 -1333.

Bay, H., Ess, A., Tuytelaars, T. & Van Gool, L., 2008. Speeded-Up Robust Features (SURF). *Comput. Vis. Image Underst.*, 110(3), pp.346--359.

Bazeille, S. & Filliat, D., 2010. Combining Odometry and Visual Loop-Closure Detection for Consistent Topo-Metrical Mapping. *RAIRO - Operations Research*, 44, pp.365-77.

Bazin, J., Demonceaux, C. & Vasseur, P., 2007. Fast Central Catadioptric Line Extraction. In *Pattern Recognition and Image Analysis, Lecture Notes in Computer Science*. Springer Berlin / Heidelberg. pp.25-32.

Bazin, J., Kweon, I., Demonceaux, C. & Vasseur, P., 2007. Rectangle Extraction in Catadioptric Images. In *Computer Vision, ICCV IEEE 11th International Conference on*., 2007.

Besl, P.J. & McKay, N.D., 1992. A Method for Registration of 3-D Shapes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2), pp.239--256.

Biber, P. & Duckett, T., 2009. Experimental Analysis of Sample-Based Maps for Long-Term SLAM. *The International Journal of Robotics Research*, 28, pp.20-33.

Botterill, T., Mills, S. & Green, R., 2011. Bag-of-words-driven, single-camera simultaneous localization and mapping. *Journal of Field Robotics*, 28, pp.204--226.

Bouraoui, L. et al., 2011. An on-demand personal automated transport system: The CityMobil demonstration in La Rochelle., 2011.

Bradski, G., 2000. The OpenCV Library. *Dr. Dobbs Journal of Software Tools*.

Briggs, A.J. et al., 2006. Matching scale-space features in 1D panoramas. *Computer Vision and Image Understanding*, 103, pp.184 - 195.

Burgard, W., Stachniss, C. & Haehnel, D., 2007. Mobile Robot Map Learning from Range Data in Dynamic Environments. In *Autonomous Navigation in Dynamic Environments*. Springer Verlag. pp.3-28.

C&D, S.R., 2011. *C&D, Skilled Robotics*. [Online] Available at: http://www.cdrobot.com/.

CARE, E., 2010. *An Executive Summary of the Strategic Research Agenda for Robotics in Europe*. Augsburg, Germany: European Comission, Coordination Action for Robotics in Europe (CARE).

Carnegie Mellon University, C.S.a.R., 2003. *NurseBot Project*. [Online] Available at: http://www.cs.cmu.edu/~nursebot/.

Castellanos, J.A. & D., T.J., 1999. *Mobile Robot Localization and Map Building: A Multisensor Fusion Approach*. Boston, MA: Kluwer Academic Publishers.

Center, A.H.M.a.C.T.R., 2011. *Pavement Crack Sealing*. [Online] Available at: http://ahmct.ucdavis.edu/.

Centro Tecnológico CARTIF, P., 2011. *Sistema de Transporte basado en Robots Móviles para Dispensación de Medicamentos en Recintos Hospitalarios*. Madrid: Centro Tecnológico CARTIF, Proingesa.

Computing Community Consortium, C.R.A., 2009. *A Roadmap for US Robotics from Internet to Robotics*. Georgia, USA.

Cummins, M. & Newman, .P., 2009. Highly Scalable Appearance-Only SLAM - FAB-MAP 2.0. In *Proceedings of Robotics: Science and Systems*., 2009.

David Schneider, I.S., 2009. *Robin Murphy: Roboticist to the Rescue*. [Online] Available at: http://spectrum.ieee.org/robotics/artificial-intelligence/robin-murphy-roboticist-to-the-rescue/0 [Accessed 2011].

Dayoub, F., Cielniak, G. & Duckett, T., 2011. Long-term experiments with an adaptive spherical view representation for navigation in changing environments. *Robotics and Autonomous Systems*, 59, pp.285 - 295.

Dayoub, F. & Duckett, T., 2008. An adaptive appearance-based map for long-term topological localization of mobile robots. In *Intelligent Robots and Systems, IROS IEEE/RSJ International Conference on*., 2008.

Dissanayake, G., Durrant-Whyte, H.F. & Bailey, T., 2000. A Computationally Efficient Solution to the Simultaneous Localisation and Map Building (SLAM) Problem. In *Robotics and Automation, Proceedings. ICRA IEEE International Conference on*., 2000.

Dorit Borrmann, J.E.S.S.R.a.A.N., 2010. Lifelong 3D Mapping – Monitoring with a 3D Scanner. In *Proceedings of the IEEE/RSJ IROS Workshop on Robotics for Environmental Monitoring*. Taipei, Taiwan, 2010.

Durrant-Whyte, H. & Bailey, T., 2006. Simultaneous Localisation and Mapping (SLAM): Part I The Essential Algorithms. *IEEE ROBOTICS AND AUTOMATION MAGAZINE*, 2, p.2006.

Eggert, D.W., Lorusso, A. & Fisher, R.B., 1997. Estimating 3-D rigid body transformations: a comparison of four major algorithms. *Mach. Vision Appl.*, 9(5-6), pp.272--290.

Einsele, T., 2001. *Localization in indoor environments using a panoramic laser range*. Ph.D. dissertation Technical University of München ed. München, Germany.

Erico Guizzo, I.S., 2010. *Tech in Sight*. [Online] Available at: http://spectrum.ieee.org/green-tech/mass-transit/tech-in-sight/0 [Accessed 2011].

Estrada, C., Neira, J. & Tardos, J.D., 2005. Hierarchical SLAM: Real-Time Accurate Mapping of Large Environments. *Robotics, IEEE Transactions on*, 21, pp.588 - 596.

EURON, 2008. *Research Roadmap V.4.1*.

EUROP, E.R.T.P., 2009. *Robotic Visions to 2020 and Beyond - The Strategic Research Agenda for Robotics in Europe*. Augsburg, Germany.

Fischler, M.A. & Bolles, R.C., 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6), pp.381--395.

ForssÃ©n, P.-E. & Lowe, D., 2007. Shape Descriptors for Maximally Stable Extremal Regions., 2007. IEEE Computer Society.

Forsyth, D.A..&.P.J., 2003. *Computer vision: a modern approach*. New York, U.S.A.: Prentice Hall.

Gallegos, G., Meilland, M., Rives, P. & Comport, A.I., 2010. Appearance-based SLAM relying on a hybrid laser/omnidirectional sensor. In *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*., 2010.

Gallegos, G. & Rives, P., 2010. Indoor SLAM based on composite sensor mixing laser scans and omnidirectional images. In *Robotics and Automation (ICRA), IEEE International Conference on*., 2010.

Gaspar, J.A., Winters, N., Grossmann, E. & Santos-Victor, J., 2007. Toward Robot Perception through Omnidirectional Vision. In *Studies in Computational Intelligence (SCI)*. Springer. pp.223-70.

Geyer, C. & Daniilidis, K., 2000. A Unifying Theory for Central Panoramic Systems and Practical Implications. In *Proceedings of the 6th European Conference on Computer Vision-Part II*., 2000. Springer Berlin / Heidelberg.

Glover, A.J., Maddern, W.P., Milford, M.J. & Wyeth, G.F., 2010. FAB-MAP + RatSLAM: Appearance-based SLAM for multiple times of day. In *Robotics and Automation (ICRA), IEEE International Conference on , vol., n*., 2010.

Goedemé, T., Nuttin, M., Tuytelaars, T. & Van Gool, L., 2007. Omnidirectional Vision Based Topological Navigation. *International Journal of Computer Vision*, 74(3), pp.219-36.

Grisetti, G. et al., 2008. Online constraint network optimization for efficient maximum likelihood map learning. In *Robotics and Automation, ICRA. IEEE International Conference on*., 2008.

Gross, H., Koenig, A. & Mueller, S., 2005. Omniview-based concurrent map building and localization using adaptive appearance maps. In *Systems, Man and Cybernetics, IEEE International Conference on*., 2005.

Guerrero, L.P.a.J.J., 2011. Scale Space for Central Catadioptric Systems. Towards a generic camera feature extractor, 13th International Conference on Computer Vision. In *ICCV*. Barcelona, 2011.

Gutmann, J. & Konolige, K., 1999. Incremental mapping of large cyclic environments. In *Computational Intelligence in Robotics and Automation, CIRA. Proceedings. IEEE International Symposium on*., 1999.

Hahnel, D., Triebel, R., Burgard, W. & Thrun, S., 2003. Map building with mobile robots in dynamic environments. In *Robotics and Automation, Proceedings. ICRA IEEE International Conference on , vol.2, no*., 2003.

Hansen, P., Corke, P. & Boles, W., 2010. Wide-angle Visual Feature Matching for Outdoor Localization. *Int. J. Rob. Res.*, 29(2-3), pp.267--297.

Hansen, P., Corke, P., Boles, W. & Daniilidis, K., 2007. Scale-Invariant Features on the Sphere. In *Computer Vision, ICCV IEEE 11th International Conference on*., 2007.

Harris, C. & Stephens, M., 1988. A combined corner and edge detector. In *Alvey Vision Conference*. Manchester, UK. pp.147-51.

Hochdorfer, S., Lutz, M. & Schlegel, C., 2009. Lifelong Localization of a Mobile Service-Robot in Everyday Indoor Environments Using Omnidirectional Vision. *2009 IEEE International Conference on Technologies for Practical Robot Applications*, pp.161--166.

Hochdorfer, S., Lutz, M. & Schlegel, C., 2009. Lifelong localization of a mobile service-robot in everyday indoor environments using omnidirectional vision. In *Technologies for Practical Robot Applications, TePRA IEEE International Conference on*., 2009.

Hochdorfer, S. & Schlegel, C., 2009. Landmark rating and selection according to localization coverage: Addressing the challenge of lifelong operation of SLAM in service robots. In *ntelligent Robots and Systems, IROS IEEE/RSJ International Conference on*., 2009.

Hokuyo, Automatic Co., 2009. *Scanning range finder URG-04LX*. [Online] Available at: http://www.hokuyo-aut.jp/02sensor/07scanner/urg_04lx.html [Accessed 2011].

Ho, K.L. & Newman, P., 2007. Detecting Loop Closure with Scene Sequences. *Int. J. Comput. Vision*, 74(3), pp.261--286.

J. Bermúdez, L.P.a.J.J.G., 2010. Line extraction in central hyper-catadioptric systems. In *OMNIVIS - 10th Workshop on Omnidirectional Vision, Camera Networks and Non-classical Cameras*. Zaragoza, 2010.

Julier, S.J., 1997. New extension of the Kalman filter to nonlinear systems. *Proceedings of SPIE*, 3068, pp.182--193.

Juretzki, B., 2011. Urban Service Robotics in the 7th Framework Programe and Beyond. Tallinn, 2011. The 15th International Conference in Advanced Robotics.

Koenig, A., Kessler, J. & Gross, H.-M., 2009. Improvements for an Appearance-based SLAM-Approach for Large-scale Environments. In *Proceedings of the 4th European Conference on Mobile Robots*., 2009.

Konolige, K. & Bowman, J., 2009. Towards lifelong visual maps. In *Intelligent Robots and Systems, IROS IEEE/RSJ International Conference on*., 2009.

Konolige, K. et al., 2009. View-based maps. In *Proceedings of Robotics: Science and Systems*., 2009.

Kretzschmar, H., Grisetti, G. & Stachniss, C., 2010. Life-long Map Learning for Graph-based SLAM Approaches in Static Environments. *KI KÃ¼nstliche Intelligenz*, 24, pp.1--7.

Kummerle, R. et al., 2011. g2o: A General Framework for Graph Optimization. In *Robotics and Automation (ICRA), IEEE International Conference on*., 2011.

Labbe, M. & Michaud, F., 2011. Memory management for real-time appearance-based loop closure detection. In *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*., 2011.

Lacey, G. & Rodriguez-Losada, D., 2008. The Evolution of Guido. *Robotics Automation Magazine, IEEE*, 15, pp.75 -83.

Lidoris, G., Rohrmuller, F., Wollherr, D. & Buss, M., 2009. The Autonomous City Explorer (ACE) project #x2014; mobile robot navigation in highly populated urban environments., 2009.

Linaker, F. & Ishikawa, M., 2006. Real-time appearance-based Monte Carlo localization. *Robotics and Autonomous Systems*, 54, pp.205 - 220.

Llinas, R.R., 2002. *I of the Vortex: From Neurons to Self*. The MIT Press.

Lloyd, S., 1982. Least squares quantization in PCM. *Information Theory, IEEE Transactions on*, 28, pp.129 - 137.

Lourenco, M., Barreto, J.P. & Malti, A., 2010. Feature detection and matching in images with radial distortion. In *Robotics and Automation (ICRA), IEEE International Conference on*., 2010.

Lowe, D.G., 2004. Distinctive Image Features from Scale-Invariant Keypoints. *Int. J. Comput. Vision*, 60(2), pp.91--110.

Madsen, K., Nielsen, H. & Tingleff, O., 2004. *Methods for Non-Linear Least Squares Problems*. Inf.and Math. Model. Technical University of Denmark.

Magazine, R., 2011. *Behind the scenes at TheRobotReport.com*. [Online] Available at: http://find.botmag.com/031121 [Accessed 2011].

Magnusson, M., Andreasson, H., Nuchter, A. & Lilienthal, A.J., 2009. Appearance-based loop detection from 3D laser data using the normal distributions transform. In *Robotics and Automation, ICRA IEEE International Conference on*., 2009.

Mardia, K.V., Kent, J.T. & Bibby, J.M., 1980. *Multivariate Analysis (Probability and Mathematical Statistics)*. Academic Press.

Matsumoto, Y., Ikeda, K., Inaba, M. & Inoue, H., 1999. Visual navigation using omnidirectional view sequence. In *Intelligent Robots and Systems, IROS Proceedings. IEEE/RSJ International Conference on*., 1999.

Mei, C., 2006. *Calibration Toolbox for omnidirectional cameras*. [Online] Available at: http://www.robots.ox.ac.uk/~cmei/Toolbox.html [Accessed 2011].

Mei, C., 2007. *Laser-Augmented Omnidirectional Vision for 3D Localisation and Mapping*. Paris: Ecole des mines de Paris, Inria Sophia Antipolis.

Mei, C. & Malis, E., 2006. Fast central catadioptric line extraction, estimation, tracking and structure from motion. In *Intelligent Robots and Systems, IEEE/RSJ International Conference on*., 2006.

Mei, C. & Rives, P., 2006. Calibration between a central catadioptric camera and a laser range finder for robotic applications. In *Robotics and Automation, ICRA Proceedings IEEE International Conference on*., 2006.

Mei, C. & Rives, P., 2007. Single View Point Omnidirectional Camera Calibration from Planar Grids. In *Robotics and Automation, IEEE International Conference on*., 2007.

Meltzer, J. & Soatto, S., 2008. Edge descriptors for robust wide-baseline correspondence. In *Computer Vision and Pattern Recognition, CVPR IEEE Conference on*., 2008.

Meyer-Delius, D., Hess, J., Grisetti, G. & Burgard, W., 2010. Temporary Maps for Robust Localization in Semi-static Environments. In *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*., 2010.

Milford, M. & Wyeth, G., 2009. Persistent Navigation and Mapping using a Biologically Inspired SLAM System. *The International Journal of Robotics Research*.

Milford, M. & Wyeth, G., 2010. Persistent Navigation and Mapping using a Biologically Inspired SLAM System. *Int. J. Rob. Res.*, 29(9), pp.1131--1153.

Ministerio de Ciencia e Innovación, G.d.E., 2011. *Hisparob*. [Online] Available at: http://www.hisparob.es/ [Accessed 2011].

Montemerlo, M. & Thrun, S., 2003. Simultaneous localization and mapping with unknown data association using FastSLAM. In *Robotics and Automation, Proceedings. ICRA IEEE International Conference on*., 2003.

Montemerlo, M., Thrun, S., Koller, D. & Wegbreit, B., 2003. FastSLAM 2.0: An Improved Particle Filtering Algorithm for Simultaneous Localization and Mapping that Provably Converges. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*., 2003.

Montesano, L., Minguez, J. & Montano, L., 2005. Modeling the Static and the Dynamic Parts of the Environment to Improve Sensor-based Navigation. In *Robotics and Automation, ICRA Proceedings of the IEEE International Conference on*., 2005.

Montijano, E. & Sagues, C., 2009. Topological maps based on graphs of planar regions. In *Intelligent Robots and Systems, IROS IEEE/RSJ International Conference on*., 2009.

Mozos, O.M. et al., 2006. Semantic Labeling of Places using Information Extracted from Laser and Vision Sensor Data. In *IEEE/RSJ IROS Workshop: From Sensors to Human Spatial Concepts*., 2006.

Murillo, A.C. et al., 2007. From omnidirectional images to hierarchical localization. *Robot. Auton. Syst.*, 55(5), pp.372--382.

Nachter, A. & Hertzberg, J., 2008. Towards semantic maps for mobile robots. *Robot. Auton. Syst.*, 56(11), pp.915--926.

Neira, J. & Tardos, J.D., 2001. Data association in stochastic mapping using the joint compatibility test. *Robotics and Automation, IEEE Transactions on*, 17, pp.890 -897.

Newman, P., Cole, D. & Ho, K., 2006. Outdoor SLAM using visual appearance and laser ranging. In *Robotics and Automation, ICRA Proceedings IEEE International Conference on*., 2006.

Nguyen, V., Martinelli, A., Tomatis, N. & Siegwart, R., 2005. A comparison of line extraction algorithms using 2D laser rangefinder for indoor mobile robotics. In *Intelligent Robots and Systems, (IROS). IEEE/RSJ International Conference on*., 2005.

Nieto, J., Bailey, T. & Nebot, E., 2007. Recursive scan-matching SLAM. *Robot. Auton. Syst.*, 55(1), pp.39--49.

Nilsback, M. & Zisserman, A., 2006. A Visual Vocabulary for Flower Classification. In *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*., 2006.

Okubo, Y..Y.C.a.B.J., 2009. Characterization of the Hokuyo URG-04LX Laser Rangefinder for Mobile Robot Obstacle Negotiation. In *SPIE Defense, Security + Sensing; Unmanned Systems Technology XI, Conference*. Orlando, 2009.

Orghidan, R., Salvi, J. & Mouaddib, E.M., 2003. Calibration of A Structured Light-Based Stereo Catadioptric Sensor. In *Computer Vision and Pattern Recognition Workshop, CVPRW Conference on*., 2003.

Pandey, G., McBride, J.R. & Eustice, R.M., 2011. Ford campus vision and lidar data set. *International Journal of Robotics Research*.

Pavlidis, T. & Horowitz, S.L., 1974. Segmentation of Plane Curves. *Computers, IEEE Transactions on*, C-23, pp.860 - 870.

Paz, L.M., Tardos, J.D. & Neira, J., 2008. Divide and Conquer: EKF SLAM in O(n). *Robotics, IEEE Transactions on*, 24, pp.1107 -1120.

Pfister, S.T., Roumeliotis, S.I. & Burdick, J.W., 2003. Weighted line fitting algorithms for mobile robot map building and efficient data representation. In *Robotics and Automation, Proceedings. ICRA IEEE International Conference on*., 2003.

Pinies, P. & Tardos, J.D., 2008. Large-Scale SLAM Building Conditionally Independent Local Maps: Application to Monocular Vision. *Robotics, IEEE Transactions on*, 24, pp.1094 -1106.

Pirker, K., Ruther, M. & Bischof, H., 2011. CD SLAM - Continuous localization and mapping in a dynamic world. In *Intelligent Robots and Systems (IROS), IEEE/RSJ International Conference on*., 2011.

Porta, J.M. & Krose, B.J., 2004. Appearance-based concurrent map building and localization using a multi-hypotheses tracker. In *Intelligent Robots and Systems (IROS). Proceedings. IEEE/RSJ International Conference on*., 2004.

Porta, J.M., Verbeek, J.J. & KrÃ¶se, B.J., 2005. Active Appearance-Based Robot Localization Using Stereo Vision. *Auton. Robots*, 18(1), pp.59--80.

Project, I., 2011. *IWard Project*. [Online] Available at: http://www.iward.eu/cms/index.php [Accessed 2011].

Pronobis, A. & Caputo, B., 2009. COLD: COsy Localization Database. *The International Journal of Robotics Research (IJRR)*, 28.

Ranganathan, A. & Dellaert, F., 2011. Online probabilistic topological mapping. *The International Journal of Robotics Research*.

Reina, A. & Gonzales, J., 1997. Characterization of a radial laser scanner for mobile robot navigation. In *Intelligent Robots and Systems, IROS , Proceedings of the IEEE/RSJ International Conference on*., 1997.

Remazeilles, A., Chaumette, F. & Gros, P., 2006. 3D navigation based on a visual memory. In *Robotics and Automation, ICRA Proceedings IEEE International Conference on*., 2006.

Ridao, P., Carreras, M., Ribas, D. & Garcia, R., 2010. Visual inspection of hydroelectric dams using an autonomous underwater vehicle. *Journal of Field Robotics*, 27, pp.759--778.

Rosten, E. & Drummond, T., 2006. Machine learning for high-speed corner detection. In *European Conference on Computer Vision*., 2006.

Rucklidge, W., 1996. *Efficient Visual Recognition Using the Hausdorff Distance*. Springer-Verlag New York, Inc.

Salvini, P., 2011. Urban Service Robotics: Challenges and Opportunities. Tallinn, 2011. The 15th International Conference in Advanced Robotics.

Sant'Anna, S.S., 2006. *DUSTbot Project*. [Online] Available at: http://www.dustbot.org/ [Accessed 2011].

Sant'Anna, S.S., 2008. *HydroNet Project*. [Online] Available at: http://www.hydronet-project.eu/ [Accessed 2011].

Scaramuzza, D., 2008. *Omnidirectional Vision: from Calibration to Robot Motion Estimation*. Zurich: ETH Zurich.

Scaramuzza, D., Fraundorfer, F. & Siegwart, R., 2009. Real-time monocular visual odometry for on-road vehicles with 1-point RANSAC. In *Robotics and Automation, ICRA IEEE International Conference on*., 2009.

Scaramuzza, D., Harati, A. & Siegwart, R., 2007. Extrinsic self calibration of a camera and a 3D laser range finder from natural scenes. In *Intelligent Robots and Systems, IROS IEEE/RSJ International Conference on*., 2007.

Scaramuzza, D., Martinelli, A. & Siegwart, Y., 2009. A Robust Descriptor for Tracking Vertical Lines in Omnidirectional Images and its Use in Mobile Robotics. *International Journal on Robotics Research*, p.000.

Segvic, S., Remazeilles, A., Diosi, A. & Chaumette, F., 2009. A mapping and localization framework for scalable appearance-based navigation. *Comput. Vis. Image Underst.*, 113(2), pp.172--187.

Siggelkow, S., 2002. *Feature Histograms for Content-Based Image Retrieval*. phdthesis.

Silveira, G., Malis, E. & Rives, P., 2008. An Efficient Direct Approach to Visual SLAM. *Robotics, IEEE Transactions on*, 24, pp.969 -979.

Sivic, J. & Zisserman, A., 2003. Video Google: A Text Retrieval Approach to Object Matching in Videos. In *Computer Vision, Proceedings. Ninth IEEE International Conference on*., 2003.

Sooyeong Yi, B.C., 2007. Real-time Omni-directional Distance Measurement with Active Panoramic Vision. *International Journal of Control Automation and System*, 5, pp.184-91.

Strandmark, P., 2010. *SURFmex*. [Online] Available at: http://www.maths.lth.se/matematiklth/personal/petter/surfmex.php [Accessed 2012].

Sujan, V.A., Meggiolaro, M.A. & Belo, F.A., 2006. Information based indoor environment robotic exploration and modeling using 2-D images and graphs. *Auton. Robots*, 21(1), pp.15--28.

Systems, C., 2007. *EUROPA - European Robotic Pedestrian Assistant*. [Online] Available at: http://europa.informatik.uni-freiburg.de/index.html [Accessed 2011].

Systems, C., 2007. *Robots at Home*. [Online] Available at: http://robots-at-home.acin.tuwien.ac.at/ [Accessed 2011].

Systems, C., 2007. *URUS Project*. [Online] Available at: http://www.urus.upc.es/ [Accessed 2011].

Systems, C., 2009. *Ubiquitous Networking Robotics in Urban Settings*. Barcelona: Cognitive Systems.

Tapus, A., Heinzer, S. & Siegwart, R., 2004. Bayesian Programming for Topological Global Localization with Fingerprints. In Automation, R.a., ed. *Proceedings. ICRA '04. 2004 IEEE International Conference on*. New Orleans, 2004.

Technologies, I.S., 2005. *COMETS Project*. [Online] Available at: http://www.comets-uavs.org/.

The Robot Report, E.-, 2011. *Disecting the Controversy about the Robotics Industry*. [Online] Available at: http://www.everything-robotic.com/2011/04/dissecting-controversy-about-robotics.html [Accessed 2011].

Thrun, S., Burgard, W. & Fox, D., 2005. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press.

Thrun, S. et al., 2003. Simultaneous Localization and Mapping with Sparse Extended Information Filters. *The International Journal of Robotics Research*, 23, pp.693--716.

Thrun, S. & Montemerlo, M., 2005. The GraphSLAM Algorithm With Applications to Large-Scale Mapping of Urban Structures. *International Journal on Robotics Research*, 25, pp.403--430.

Thrun, S. et al., 2004. Autonomous exploration and mapping of abandoned mines. *Robotics Automation Magazine, IEEE*, 11, pp.79 - 91.

Tobe, F., 2011. *The Robot Report*. [Online] Available at: http://www.therobotreport.com/.

Ullah, M.M. et al., 2008. Towards robust place recognition for robot localization. In *Robotics and Automation, ICRA IEEE International Conference on*., 2008.

University, S.H., 2010. *GUARDIANS final report*. [Online] Available at: https://shura.shu.ac.uk/2340/ [Accessed 2012].

Unnikrishnan, R. & Hebert, M., 2005. *Fast Extrinsic Calibration of a Laser Rangefinder to a Camera*. techreport.

Walter, M.R., Eustice, R.M. & Leonard, J.J., 2007. Exactly Sparse Extended Information Filters for Feature-based SLAM. *Int. J. Rob. Res.*, 26, pp.335--359.

Wang, C.-C. & Thorpe, C., 2002. Simultaneous localization and mapping with detection and tracking of moving objects. In *Robotics and Automation, Proceedings. ICRA IEEE International Conference on*., 2002.

Wang, C.-C., Thorpe, C. & Thrun, S., 2003. Online simultaneous localization and mapping with detection and tracking of moving objects: theory and results from a ground vehicle in crowded urban areas. In *Robotics and Automation, Proceedings. ICRA IEEE International Conference on*., 2003.

Wasielewski, S. & Strauss, O., 1995. Calibration of a multi-sensor system laser rangefinder/camera. In *Intelligent Vehicles Symposium., Proceedings of the*., 1995.

Weingarten, J., 2006. Feature-based 3D SLAM. In *Intelligent Robots and Systems, IEEE/RSJ International Conference on*., 2006.

Weingarten, J. & Siegwart, R., 2005. EKF-based 3D SLAM for structured environment reconstruction. In *Intelligent Robots and Systems, (IROS). IEEE/RSJ International Conference on*., 2005.

Wongphati, M., Niparnan, N. & Sudsang, A., 2008. Bearing only FastSLAM using vertical line information from an omnidirectional camera. In *Robotics and Biomimetics, ROBIO IEEE International Conference on*., 2008.

Xiao, J. et al., 2011. Fast plane detection for SLAM from noisy range images in both structured and unstructured environments. In *Mechatronics and Automation (ICMA), International Conference on*., 2011.

Xu, L. & Oja, E., 1993. Randomized Hough Transform (RHT): Basic Mechanisms, Algorithms, and Computational Complexities. *CVGIP: Image Understanding*, 57, pp.131 - 154.

Ye, C. & Borenstein, J., 2002. Characterization of a 2D laser scanner for mobile robot obstacle negotiation. In *Robotics and Automation, Proceedings. ICRA IEEE International Conference on*., 2002.

Zhang, Z., 1997. Parameter estimation techniques: a tutorial with application to conic fitting. *Image Vision Comput.*, 15, pp.59-76.

Zhang, Q. & Pless, R., 2004. Constraints for Heterogeneous Sensor Auto-Calibration. In *Computer Vision and Pattern Recognition Workshop, CVPRW Conference on*., 2004.

Zhang, Q. & Pless, R., 2004. Extrinsic calibration of a camera and laser range finder (improves camera calibration). In *Intelligent Robots and Systems, (IROS). Proceedings. IEEE/RSJ International Conference on*., 2004.

Zhou, C., Wei, Y. & Tan, T., 2003. Mobile robot self-localization based on global visual appearance features., 2003.

Zivkovic, Z., Bakker, B. & Krose, B., 2006. Hierarchical map building and planning based on graph partitioning., 2006.

Zivkovic, Z., Booij, O. & Krose, B., 2007. From images to rooms. *Robot. Auton. Syst.*, 55(5), pp.411--418.