ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY

MINES ParisTech

École doctorale n°89 :
Sciences et technologies de l'information et de la communication

Doctorat ParisTech

THÈSE

pour obtenir le grade de docteur délivré par

l'École Nationale Supérieure des Mines de Paris

Spécialité « Informatique temps-réel, robotique et automatique »

présentée et soutenue publiquement par

Gabriela GALLEGOS GARRIDO

le 17 juin 2011

Développement d'un Capteur Composite Vision/Laser à Couplage Serré pour le SLAM d'Intérieur

---

Development of a Tightly-Coupled Composite Vision/Laser Sensor for Indoor SLAM

Directeur de thèse : **Patrick RIVES**

THÈSE

Jury

M. Jean-Paul MARMORAT, Directeur de recherche, CMA, MINES ParisTech	Président
M. El Mustapha MOUADDIB, Professeur, Laboratoire MIS, UPJV	Rapporteur
M. Philippe MARTINET, Professeur, LASMEA, Université Blaise Pascal	Rapporteur
M. Alessandro-Corrêa VICTORINO, Professeur, HEUDIASYC, UTC	Examinateur
M. Cédric DEMONCEAUX, Professeur, UMR CNRS, Université de Bourgogne	Examinateur
M. Patrick RIVES, Directeur de Recherche, INRIA Sophia Antipolis-Mediterranée	Directeur de Thèse

MINES ParisTech
Centre de Mathématiques Appliquées
Rue Claude Daunesse B.P. 207, 06904 Sophia Antipolis Cedex, France

To my parents

Acknowledgements

The research described in this thesis was carried out from October 2006 to June 2011 at INRIA-Sophia Antipolis Mediterranée in the AROBAS (before ICARE) project directed by Patrick Rives. I would like to thank to all the people who in a way or another helped me in the elaboration of this thesis work.

First of all I would like to thank the members of the jury for having accepted to assess my thesis:

- Jean-Paul MARMORAT, Directeur de recherche, CMA, MINES ParisTech for his support being the link between the ENSMP and INRIA along the thesis and to have honored me by presiding the jury.
- El Mustapha MOUADDIB, Professeur, Laboratoire MIS, UPJV for his particularly rigorous reading, constructive criticism and time spent in writing the report.
- Philippe MARTINET, Professeur, LASMEA, Université Blaise Pascal for his encouragement and time spent in reading, analyzing and writing the report.
- Alessandro-Corrêa VICTORINO, Professeur, HEUDIASYC, UTC for his advices and for having accepted to be a member of the jury.
- Cédric DEMONCEAUX, Professeur, UMR CNRS, Université de Bourgogne for his interest in my work and for having accepted to be a member of the jury.

I would particularly like to express my deepest gratitude to my thesis supervisor Patrick RIVES for having confidence in me to offer this incredible research topic, for his support and availability throughout my thesis. Especially, I want to thank him for his patience also in the difficult moments.

I thank the Mexican Council of Science and Technology (CONACYT) for giving me the scholarship (number of registry: 172914) to support my doctoral studies.

Furthermore, I thank all the members and ex-members of the AROBAS project. Ezio Malis, Claude Samson and Pascal Morin for all his advices. I can not forget all my colleagues and friends, Christopher, Hicham, Andrew, Geraldo, Mauro, Mélaine, Benoît, Tiago, Thomas, Maxime, Minh-Duc, Stefan and Cyril.

Similarly, thanks to all the hispanic community at INRIA: Santiago, Tamara, Marcela, Alina, Cristian, Leo, Maria-José, JC, Tavo, Jorge, Dany, Guido, Ezequiel, Emilien, Sapna and Julien, for always share the good and bad moments.

Last but not least, I would like to make a special thanks to my parents Maria de Lourdes and Victor Armando, as well as my sisters Alma, Pato y Mimi for have always believe in me and never letting me down.

Abstract

Autonomous navigation in unknown environments has been the focus of attention in the mobile robotics community for the last three decades. When neither the location of the robot nor a map of the region are known, localization and mapping are two tasks that are highly inter-dependent and must be performed concurrently. This problem, is known as *Simultaneous Localization and Mapping* (**SLAM**).

In order to gather accurate information about the environment, mobile robots are equipped with a variety of sensors (e.g. laser, vision, sonar, odometer, GPS), that together form a *perception system*, that allows accurate localization and reconstruction of reliable and consistent representations of the environment. Vision sensors give mobile robots relatively cheap means of obtaining rich 3D information on their environment, but lack the depth information that laser range finders can provide. We believe that a perception system composed of the odometry of the robot, an omnidirectional camera and a 2D laser range finder provide enough information to solve the SLAM problem robustly.

Nowadays, laser range finders have replaced sonars when possible because of its superior efficacy in estimating distances accurately and their better signal to noise ratio. Many techniques have been developed to make the most of this type of sensor for solving the SLAM problem. Since a laser scan directly provides metric information of the scene, the localization problem can be stated in terms of an odometry-based method where the incremental displacement is found by computing the best rigid transformation that matches two successive scans. To match two scans it is necessary to link the individual measurements in one scan with the corresponding measurements in the other scan.

It is a well known fact that geometrical structures such as lines or planes characterize well a human-made environment. When using an omnidirectional camera, vertical lines in the scene (e.g. walls, facades, doors, windows) project as quasi-radial lines onto the image. A Hough transform was used to detect prominent lines from a binary edge image. Since the camera and laser are calibrated, the image center (i.e, where all radial lines intersect) is available and it is possible to project the laser trace onto the omnidirectional image. Subsequently, at each intersection point between the laser trace and a radial line, a depth measurement can be determined which then fully characterizes the vertical lines in the 3D scene. Furthermore, un-

der the planarity assumption, the laser scan can be shifted along the vertical lines to predict where a virtual laser trace –corresponding to the floor– should project in the omnidirectional image. Due to calibration errors, the predicted trace does not exactly match the real boundary of the floor observed in the image. In practice, the neighborhood of the predicted trace is searched for the closest element of contour detected in the image

In this context we propose an *appearance-based* approach to solve the SLAM problem and reconstruct a reliable 3D representation of the environment. This approach relies on a tightly-coupled laser/omnidirectional sensor in order to tackle the drawbacks of both sensors.

Firstly, a novel generic robot-centered representation that is well adapted to the appearance-based SLAM is proposed. Central omnidirectional cameras can be modeled using two consecutive projections: a *spherical* projection followed by a *perspective* one. An omnidirectional image can thus be mapped onto a sphere by means of an inverse projection. Therefore, the augmented spherical view is constructed using the depth information from the laser range finder and the floor plane, together with lines extracted from the omnidirectional image. In other words, each pixel of the spherical view is associated with a brightness function and is augmented with the depth of the associated 3D point (when data is available).

Secondly, our appearance-based localization method minimizes a non-linear cost function directly built from the augmented spherical view described before. The minimization uses a robust M-estimator in order to reject the outliers due to illumination changes, moving objects or occlusions in the scene. However, iterative methods suffer from convergence problems when initialized far from the solution. This is also true for our method where an initialization sufficiently close to the solution is needed to ensure rapid convergence and reduce computational cost. A Enhanced Polar Scan Matching algorithm is used to obtain this initial guess of the position of the robot to initialize the algorithm.

Résumé

Depuis trois décennies, la navigation autonome en environnement inconnu est une des thématiques principales de recherche de la communauté robotique mobile. Lorsque ni la localisation du robot, ni la cartographie ne sont connues, ces deux taches deviennent extrêmement interdépendantes et doivent être accomplies simultanément. Ce problème est connu sous le nom de **SLAM** (*Simultaneous Localization And Mapping*).

Pour obtenir des informations précises sur leur environnement, les robots mobiles sont équipés d'un ensemble de capteurs (laser, vision, sonar, odomètre, GPS,...). Cette combinaison de capteurs appelé *système de perception* leur permet d'effectuer une localisation précise et une reconstruction fiable et cohérente de leur environnement. Les capteurs de visions permettent l'acquisition d'informations 3D riches à un coût réduit, mais ils leurs manquent les informations de profondeur qu'un télémètre laser apporte. Nous pensons qu'un système de perception composé de l'odométrie du robot, d'une camera omnidirectionnelle et d'un télémètre laser 2D est suffisant pour résoudre de manière robuste les problèmes de SLAM.

Désormais, les télémètres lasers ont remplacé lorsque c'était possible les sonars car ils fournissent des mesures de distances plus précises et un meilleur rapport signal/bruit. De nombreuses solutions ont été mises en place pour résoudre les problèmes de SLAM avec ce type de capteur. Puisque le balayage fournit directement les données métriques de la scène, le problème de localisation peut être reformulé en un problème d'odométrie où le déplacement est déterminé en estimant la meilleure transformation rigide de mise en correspondance de 2 balayages successifs. La correspondance entre deux balayages est nécessaire pour lier les mesures indépendantes d'un balayage avec les mesures associés dans le balayage précédent.

Il est admis que les formes géométriques simples telles que les lignes ou les plans permettent de bien représenter les environnements construits par l'homme. Sur une caméra omnidirectionnelle, les lignes verticales de la scène (les murs, les façades, les portes, le fenêtres,...) sont projetés en ligne quasi-radiale sur l'image. Une transformation de Hough est utilisée pour détecter les lignes principales à partir de l'image binaire. Comme nous utilisons une caméra et un laser calibrés, le centre de l'image (où toute les lignes radiales se croisent) est connu et il est possible de projeter les données du laser sur l'image omnidirectionnelle. Dans une seconde phase, à chaque intersection entre une coupe laser et une ligne radiale, une mesure de la profondeur

peut être effectuée pour définir complètement la représentation des lignes dans la scène 3D.

Si on fait l'hypothèse d'environnement planaires par morceaux, on peut effectuer une translation des données du balayage laser le long des lignes verticales pour déterminer où les traces (virtuelles) du laser correspondant au sol doivent se projeter dans l'image omnidirectionnelle. Cette prédiction de trace ne correspond pas tout à fait aux limites réelles du sol à cause des erreurs de calibration. En pratique, on corrige cette erreur par une recherche de l'élément de contours de l'image le plus proche de l'estimation de trace laser.

Dans ce contexte, nous proposons une approche *appearance-based* pour résoudre les problèmes de SLAM et effectuer une reconstruction 3D fiable de l'environnement. Cette approche repose sur un couplage serré entre les capteurs laser et omnidirectionnel qui permet de compenser leurs imprécisions.

D'une part, nous proposons une représentation originale et générique de l'espace pour les robots bien adapté aux méthodes de type *appearance-based* pour le SLAM. Le centre des caméras omnidirectionnelles peut être modélisé à partir de deux projections consécutives: une projection sphérique suivie d'une projection perspective. L'image omnidirectionnelle peut ainsi être projeté (mise en correspondance) sur une sphère grâce à une projection inverse. Ainsi, la vue augmenté sphérique est construite en utilisant les mesures de profondeur du télémètre laser et la position du sol, associé aux lignes extraient de l'image omnidirectionnelle. En d'autres termes, chaque pixel de la vue sphérique est associé à une fonction de luminosité et est augmenté en utilisant la profondeur du point 3D associé (quand il est disponible).

D'autre part, notre méthode de localisation de type *appearance-based* minimise une fonction de coût non-linéaire directement construite à partir de la vue sphérique augmentée décrite précédemment. Cette minimisation utilise un M-estimateur robuste permettant de rejeter les points aberrants à cause des changements d'illumination, des objets mobiles ou des occlusions qui peuvent survenir dans la scène. Cependant ces méthodes itératives souffrent de problèmes de convergence quand l'initialisation est loin de la solution. Ce problème est aussi présent dans notre méthode où une initialisation suffisamment proche de la solution est nécessaire pour s'assurer une convergence rapide et pour réduire les coûts de calcul. Pour cela, on utilise un algorithme de PSM amélioré pour prédire la position initiale du robot.

Notations and Acronyms

General

$\mathbf{M}$	:	matrix $\mathbf{M}$
$\mathbf{v}$	:	vector $\mathbf{v}$
$\mathbf{R}$	:	rotation matrix
$\mathbf{t}$	:	translation vector
$\mathbf{T}$	:	Euclidean transformation
$\mathbf{M}^\top$	:	the transpose of the matrix $\mathbf{M}$
$\lVert\mathbf{x}\rVert$	:	L_2 norm of $\mathbf{x}$
$\mathcal{F}$	:	a reference frame
$\mathbf{x} \times \mathbf{y}$	:	cross product between $\mathbf{x}$ and $\mathbf{y}$
$[\mathbf{x}]_\times$	:	skew-symmetric matrix associated to $\mathbf{x}$, $[\mathbf{x}]_\times \mathbf{y} = \mathbf{x} \times \mathbf{y}$
$\mathbf{M}^+$	:	pseudo-inverse of $\mathbf{M}$, $\mathbf{M}^+ = (\mathbf{M}^\top\mathbf{M})^{-1}\mathbf{M}^\top$
$\mathbf{0}_{m\times n}$	:	a matrix with m lines and n columns with zero values
$\mathbf{I}_n$	:	the identity matrix of size $n \times n$

Projective geometry

$\boldsymbol{\mathcal{X}} = (X, Y, Z)$	:	a 3D point
$\boldsymbol{\mathcal{X}}_s = (X_s, Y_s, Z_s)$	:	a point belonging to the unit sphere ($\lVert\boldsymbol{\mathcal{X}}_s\rVert$=1)
$\mathbf{p} = (u, v)$	:	coordinate of a point in the image
$\mathbf{m} = (x, y)$	:	coordinate of a point on the normalised plane
Π	:	function that projects a 3D point to the image plane
$\mathbf{K}$	:	camera projection matrix
$\hbar$	:	function projecting a 3D point to the normalized plane for an omnidirectional sensor
$\boldsymbol{\mathcal{L}}$	:	2D or 3D line
$\boldsymbol{\mathcal{I}}$	:	an image

Acronyms

FOV	:	Field of View
SLAM	:	Simultaneous Localization and Mapping
CML	:	Concurrent Map-building and Localization
GPS	:	Global Positioning System
INS	:	Inertial Navigation System
GNSS	:	Global Navigation Satellite System
PL	:	Pseudolite
IMU	:	Inertial Measurement Unit
AVL	:	Absolute Visual Localization
VO	:	Visual Odometry
LSM	:	Laser Scan Matching
PSM	:	Polar Scan Matching
EPSM	:	Enhanced Polar Scan Matching
ToF	:	Time of Flight
SIFT	:	Scale-Invariant Feature Transform
RANSAC	:	RANdom SAmple Concensus
ICP	:	Iterative Closest Point
IMRP	:	Iterative Matching Range Point
SSD	:	Sum of Squared Differences
IDC	:	Iterative Dual Correspondence
EKF	:	Extended Kalman Filter
UKF	:	Unscented Kalman Filter
KF	:	Kalman Filter
EIF	:	Extended Information Filter
IF	:	Information Filter
JCT	:	Joint Compatibility Test
RJC	:	Randomized Joint Compatibility
UPM	:	Unified Projection Model
ESM	:	Efficient Second-order Minimization
D&C	:	Divide and Conquer
NNG	:	Nearest Neighbour Gating
PF	:	Particle Filters
EM	:	Expectation Maximization
DOF	:	Degree Of Freedom

Contents

List of Figures

List of Tables

List of Algorithms

"A man who moves mountains starts by carrying away small stones."

Confucio

1 Introduction

Just a few decades ago, when thinking about robots, C-3P0 and R2-D2 as well as Assimov stories surely were the pictures that came to the mind of the common people. Nowadays scientific reality, in the form of the well known mars rovers [Stone (1996)], assembly line robotic arms or Honda showroom robots, is for sure inside those same minds. It can be expected that, in the near future, robots will be as common to our lives as personal computers are right now.

Researchers have been intensely focused on creating machines that are capable of performing –in an autonomous way– tedious or dangerous tasks that used to be done by humans. Indeed, since the first developments in the 60's, mobile robots represent one of the major challenges for researchers: to design and create an integrated robotic system able to move and act, safely and independently in an *a priori* unknown dynamic environment of large dimension.

Mobile robots can be classified based on the environment in which they travel and based on the device that they use to move. Based on the environment in which they move there are the mobile robots oriented to human-made indoor environments and the mobile robots oriented to unstructured outdoor environments. The last one can include flying-oriented robots, space-oriented robots and underwater robots. Based on the device they use to move we can find the legged robots, wheeled-based robots and the ones with tracks.

1.1 Motivations

A key issue in mobile robotics is to provide robots the ability to navigate in an autonomous way in unknown environments based only on their perception. Thus, a mobile robot must be equipped with a perception system capable of providing accurate information of its current location and its surroundings, so that the robot is able to reconstruct a reliable and consistent representation of the environment. There are two interdependent tasks that any mobile robot has to solve: localization and mapping. When neither the location of the robot nor the map are known, both tasks must be performed concurrently. This problem, known as Simultaneous Localization and Mapping (SLAM), has been largely studied since the seminal work of Smith and Cheeseman (1986) and Smith et al. (1986), and is closely related to the development of sensor technologies.

Understanding the environment from sensor readings is a fundamental task in mobile robots. The sensors embedded on the mobile robot can be classified as proprioceptive sensors and exteroceptive sensors. These sensors provide different and complementary information about the environment, which is why nowadays, mobile robots are equipped with several sensor systems to avoid the limitations when only one sensor is used to reconstruct the environment. Information from different sensors measuring the same feature, can be fused to obtain a more reliable estimate, reducing the final uncertainty on the measurement. Sensor fusion can be done at different levels: loose integration or tight integration. The term *integration*, can be defined as the fusion of two separate entities, resulting in a new entity. In loose integration –also called *loose coupling*– the state estimations provided by each independent sensor are fusioned. On the other hand, tight integration –also called *tight coupling*– consist in directly fuse the outputs (raw data) of each sensor. Loose and tight integration have been widely studied for many years mostly using Inertial Navigation Systems (INS) and Global Navigation Satellite Systems (GNSS) for efficient autonomous navigation purposes. Greenspan (1996) in his work on INS/GPS describes the loose and tight integration architectures.

It is well known that localization methods based only on proprioceptive sensors give bad results due to modeling approximations (e.g., rolling without slippage,...) which are not satisfied and dead reckoning drift. Various techniques to solve the SLAM problem using laser range finders have been extensively studied. The information provided by laser range finders can be used not only to obtain a more accurate position estimate, but also to measure the distance to nearby objects. Localization schemes based on laser scan matching involve computing the most likely alignment between two sets of slightly displaced laser scans, requiring an initial estimate of the pose that can be obtained from the robot odometry. While laser-based schemes perform reasonably well in practice, the use of 2D laser alone limits SLAM to planar motion estimation and does not provide sufficiently rich information to reliably identify previously explored regions. Vision sensors are a natural alternative to laser range finders because they provide richer perceptual information and enable 6 degrees of freedom motion estimation.

On the one hand, standard cameras only have a small field of view (typically between 40° and 50°) and are easily affected by occlusion. On the other hand, omnidirectional cameras provide full 360° field of view, which makes easier to recognize previously observed places whatever the orientation of the robot is. Furthermore, in order to avoid the limitations due to planar projections, images captured by these cameras can be uniquely mapped to spherical images [Geyer and Daniilidis (2000)]. Nevertheless, vision alone does not provide the depth information that a laser range finder does, which is crucial for solving the localization problem.

For the foregoing reasons, we believe that a perception system composed of the odometry of the robot, an omnidirectional camera and a 2D laser range finder provide enough information to solve the SLAM problem robustly. We also believe that using directly the data as it come out from the sensors have a lot of advantages, which is why we choose a tight coupled approach for our experiments. In fact, complementary information from cameras (panoramic or omnidirectional) with the depth information acquired by a laser range finder have gained increasing attention in the last decade.

Clerentin et al. (2000) described a localization method using the combination of a low-cost laser range finder and an omnidirectional vision system called SYCLOP. However, this vision system is composed by a conic mirror and a CDD camera, therefore not satisfying the single viewpoint constraint. The low-cost laser range is limited to 5-6 meters and in order to achieve a correct localization of the robot, a theoretical map of the environment is given. In the work of Cobzas et al. (2003), the perception system is composed of a laser range finder and a CDD camera mounted in a pan-tilt unit to rotate the camera in order to build a cylindrical or panoramic image model. The proposed localization algorithm uses lines as features, which have to be selected manually. Even more, an initial position and the height difference between the model location and the robot have to be estimated at the beginning by manually selecting corresponding feature points. On the other hand, Biber et al. (2004) are more interested on bulding visually realistic maps using a perception system compose by a laser scanner and a CDD camera with an omnidirectional lens attachment. His method consist of manual, semi-automatic and automatic parts. Recording the data and calibration is done manually by teleoperation, and extraction of the walls is done semi-automatically with an user interface. The rest of the processing is fully automatic.

In contrast to known methods, the schema proposed in this thesis is a fully automatic process to be discussed in detail in the following section.

1.2 Scope of the Thesis

Goal and Methodology

Based on the discussion presented on the previous section, many methods for localization and mapping have been proposed in the last decades. However, it is not an easy task to correctly obtain a reliable estimation of the current location of the robot, while at the same time, obtain an accurate enough map of the navigation area. Besides, fully automated processes permitting a higher level 3D representation of the environment are very rare. These two requirements have motivated our research in the problems of localization and the automatic construction of a global map of the navigation area of the robot.

The research work presented in this thesis focuses in wheeled-based mobile robots that navigate in human-made indoor environments. More precisely, we are interested on a multi-sensor perception, localization and mapping. The perception system for this work is constituted by odometry as proprioceptive sensor; an omnidirectional camera and a 360° 2D laser range finder as exteroceptive sensors. The aim of this thesis is to contribute towards gaining a better understanding of the conception and use of hybrid sensors to solve the SLAM problem. It also proposes concrete solutions for improving 2D laser-based SLAM and perceptually rich 3D textured map representation.

In this context we face the simultaneous mobile robot localization and map building problem reconsidering the global approach proposed by Biber et al. (2004). The major difference is that the process we describe is fully automated and does not require manual post-processing by an operator. The methodology used in this work is shown in Figure 1.1.

It is important to remark some assumptions that are considered throughout this thesis:

- It is assumed that the robot navigates on a horizontal, even surface, which is referred to as the motion plane.
- It is assumed that the robot navigates in an static environment, which can be accurately represented by a set of vertical planes perpendicular to the motion plane. However, thanks to the robustness of our method, dynamic objects in the scene can be detected and rejected as perturbations.
- It is assumed that the sensors –laser and omnidirectional camera– are correctly calibrated.
- It is assumed that the distance between the sensors and the floor is approximately known (which requires the plane to be horizontal).

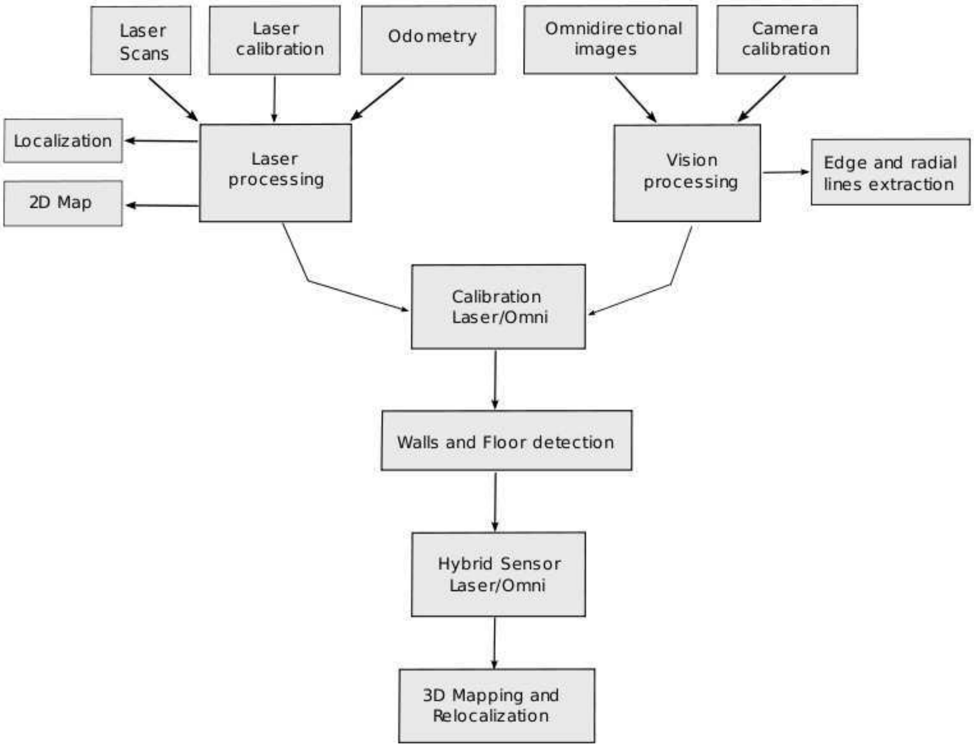


Figure 1.1: Overview of our methodology.

Main Contributions

The main contributions of this thesis are:

1. *The generalization of the Polar Scan Matching technique* proposed by Diosi and Kleeman (2005). Our implementation is parametrized so that it can deal with lasers with arbitrary angular resolution and bearing range. In addition, instead of just returning the pose estimate at the moment the algorithm stops, our implementation keeps record of the estimate with the minimum error and returns it as a result. This generalization will be called the *Enhanced Polar Scan Matching* (EPSM).

2. *The introduction of an original composite sensor approach* that takes advantage of the information given by an omnidirectional camera and a laser rangefinder to efficiently solve the SLAM problem. We developed a procedure to extract vertical lines from omnidirectional images, while at the same time estimating their 3D positions using laser information. This lines were used to build a 3D wired representation of the environment. This approach was published in the IEEE International Conference on Robotics and Automation (ICRA 2010) [Gallegos and Rives (2010)]. An extended version is presented in chapter 3.

3. *The improvement of the composite sensor mentioned above into a hybrid laser/vision*

appearance-based approach in order to obtain a more reliable 3D odometry robust to illumination changes. Furthermore, a complete set of 3D points can be easily mapped to reconstruct a dense and consistent representation of the environment. This approach was published in the IEEE International Conference on Intelligent Robots and Systems (IROS 2010) [Gallegos et al. (2010)]. An extended version is presented in chapter 4.

All approaches introduced in this work have been validated with two mobile platforms for indoor environments: Anis and Hannibal. These two platforms are described in more detail in the next section. The indoor environments used in this work are part of our institute building. They include scenarios found in typical office-like environments such as places that look the same (e.g. corridors) and people moving around.

1.3 The Platforms

For the work reported here, two mobile platforms were used: Anis (see Figure 1.2) and Hannibal (see figure 1.3) for office-like indoor environments. Both robots are part of the experimental testbeds at AROBAS project in INRIA Sophia Antipolis. The relevant characteristics of each platform are mentioned in the following.

1.3.1 Anis

Anis is the first robot platform that we used in our experiments. Anis is equipped with three types of sensors: an AccuRange 4000 2D laser range finder, a catadioptric camera and proximity sensors. Using Anis, sequences of odometry, laser and vision data were taken. The laser with which Anis is equipped, is composed of a laser telemeter with a rotating mirror that allows measurements of points on $360°$, except for an occlusion cone of approximately $30°$ caused by the assembly of the mirror. The telemeter computes distances using an intermediate technology between frequency modulation and amplitude modulation. The range finder reaches a maximal frequency of 50Hz and is capable of acquiring 2000 data points in 40ms, which is more than enough for real-time applications. The perspective camera is a progressive-scan CCD camera (Marlin F-131B) equipped with a telecentric lens and a parabolic mirror (S80 from Remote Reality).

1.3.2 Hannibal

The most recent robot acquire by AROBAS project is Hannibal, from Neobotix mobile platform (MP-S500). Hannibal is equipped with a Sick LD-LRS1000 laser, capable of collecting full $360°$ data. The laser head can revolve with a variable frequency ranging from 5Hz to 10Hz and the angular resolution can be adjusted up to $1.5°$ at multiples of $0.125°$. To perform a $360°$ scan with a resolution of $0.25°$, for example, it is necessary

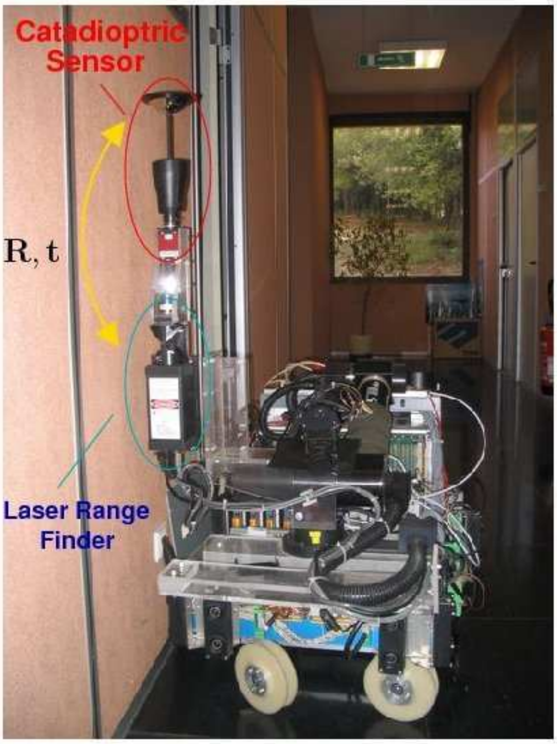


Figure 1.2: Anis robot experimental testbed.

to reduce the frequency of the rotor to 5Hz. This allow us to obtain 1,400 data points per scan. The perspective camera is a progressive-scan CCD camera (Marlin F-131B) equipped with a hyperbolic mirror HM-N15 from Accowle (Seiwapro) with a black needle at the apex of the mirror to avoid internal reflections of the glass cylinder. In Hannibal, odometry data arrives at a frequency of 50Hz, omnidirectional images at 15Hz and laser measurements at 5Hz. Since data from the different sensors that it uses arrive at different frequencies, we implemented a function to synchronize the data as it comes out from the robot.

Remark: *Careful calibration of the laser and the camera is required for merging image and laser data. We used the Matlab Omnidirectional Calibration Toolbox*[1] *developed by Mei to estimate the intrinsic parameters of the camera and the parameters of the hyperbolic/parabolic mirror [Mei and Rives (2006b)].* ■

1.4 Detailed plan

This section outlines the structure of the thesis and summarizes the content of each of the chapters.

[1] http://www.robots.ox.ac.uk/~cmei/Toolbox.html

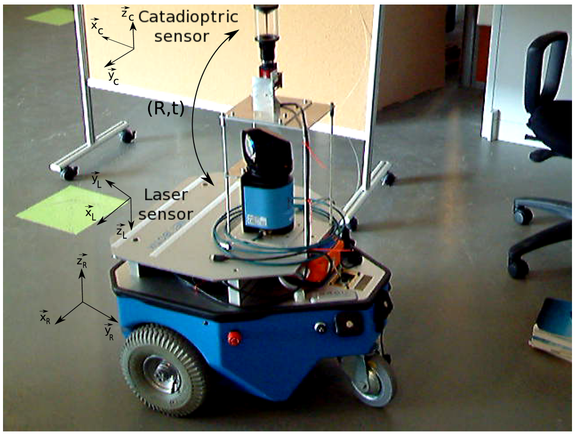


Figure 1.3: Hannibal robot experimental testbed.

Chapter 2. SLAM: State of the Art. This chapter states the Simultaneous Localization and Mapping (SLAM) problem and presents a survey of the work done during the last decades related to it. The most commonly used estimation methods and map representations by the robotics community for SLAM are also described. Since the SLAM problem is closely related to the development of sensor technologies, a complete description of the sensors used in this work is presented in the last sections.

Chapter 3. 2D Laser Based SLAM. In this chapter, the generalities of the Laser Scan Matching (LSM) algorithm are presented. Then we focus on the Polar Scan Matching (PSM) algorithm and the generalizations that we made in order to improve it and make it robust enough to deal with lasers with arbitrary angular resolution and bearing range. Then, we described how to build local maps using the Enhanced Polar Scan Matching (EPSM). Finally, the SLAM framework to reconstruct 2D global map from which it is possible to recover the pose of the robot at each instant is introduced.

Chapter 4. Tightly Coupled Sensors Fusion. In this chapter, we introduce the main aspects of omnidirectional vision. Different methods to acquire large field of views will be described and we will explain the advantages of central catadioptric sensors for robotics, as well as the models used. The second part of the chapter will be consecrated to describe how to link images obtained by an omnidirectional camera with a laser range finder in order to build a composite laser/omnidirectional sensor that will enhance both, localization and map representation of the robot's environment. The developed procedure to extract vertical lines from omnidirectional images and to estimate their 3D positions using information from the laser range finder will be explained. This lines will allow to build a 3D wired representation of the environment.

Chapter 5. Appearance-Based SLAM. In this chapter, a brief introduction to visual SLAM and the different approaches found in the computer vision community is given. The core part of this chapter is the introduction of a novel and efficient hybrid laser/vision appeareance-based SLAM approach, in order to provide the mobile robot with rich 3D information about the environment. Our approach consists on initialize the tracking algorithm with the EPSM in order to ensure rapid convergence and reduce computational cost.

Chapter 6. Conclusion and Future work. Based on a review of our results, we present our general conclusions and propose potential avenues for further research.

"Tell me and I'll forget. Teach me and I may remember. Involve me and I'll learn."

Chinese Proverb

2
SLAM: State of the Art

Overview

Simultaneous Localization and Mapping (SLAM) is a method that allows a mobile robot placed in an unknown environment in an unknown location to build a map of its surroundings while at the same time determining and keeping track of its current location within the environment. Solving the SLAM problem provides the means to make a mobile robot truly autonomous, which is the reason why the problem has drawn a lot of attention from researchers during the last two decades.

In order to gather accurate information about the environment, mobile robots are equipped with a variety of sensors (e.g. laser, vision, sonar, odometer, GPS), that together form a perception system allowing accurate localization and reconstruction of reliable and consistent representations of the environment.

In this chapter we stated the SLAM problem, its notation and some methods for solving the SLAM problem. We review recent work on the subject, with a focus on the kind of problems that we address in the remainder of this work. In particular, we overview various ways of representing maps and the most typical kind of sensors used in mobile robots, with an emphasis on the sensors used in this work.

Keywords: Simultaneous Localization and Mapping, Kalman Filter, Bundle Adjustment, Extended Kalman Filter, Particle Filter, metric and topological maps, data fusion.

Organization of the chapter

This chapter is organized as follows:

Section 2.1 presents a survey into the field of robotic localization and mapping with a focus on indoor environments. The use of different sensors –namely ultrasonic, laser, and vision sensors– used over the years for solving the SLAM problem, as well as their advantages and disadvantages are discussed. It is shown that the tendency to merge or fuse the information of two different sensors, (e.g. laser and vision) helps to overcome the drawbacks of using only one sensor.

Section 2.2 states mathematically the general SLAM problem and the notation used in the reminder of the thesis.

Sections 2.3 and 2.4 describe the two major formulations to solve the SLAM problem: The probabilistic approach and the optimization approach. A brief state of the art of the most common methods used in the robotics community for each one of the approaches is presented.

Section 2.5 aims to present a summary of the environment modeling techniques to solve the mapping problem. The main representations of maps, which are classified into three big categories, namely metric, topological and hybrid maps, are described in detail.

Sections 2.6 and 2.7 explain the two different sensors embedded in our mobile robot: the proprioceptive and exteroceptive sensors. It is shown that this sensors provide different and complementary information about the environment. Each one of the sensors used in this thesis are classified and described in detail. Particularly for vision sensors, the principal camera models are described. It will be explained how to acquire large fields of views, as well as the advantages that it has in robotics applications.

Section 2.8 explains the choice of using a tight integration to fuse the data from a laser range finder and an omnidirectional camera.

Section 2.9 gives the conclusion of the chapter.

2.1 A Survey of SLAM

Localization methods based only on proprioceptive sensors give bad results due to modeling errors (rolling without slippage) and sensor drift (IMU's). For these reasons, map integrity cannot be sustained only using this type of sensors. Exteroceptive sensors, e.g. laser range finders, omnidirectional cameras, ultrasonic sensors, provides supplementary and valuable information. Thus, the use of multi-sensor platforms is becoming the norm in contemporaneous mobile robot design.

Ultrasonic sensors have been largely used mainly because of its low cost, operation simplicity and fast acquisition of the environment model. They are based on a Time-of-Flight (ToF) principle using an ultrasonic wave. Considerable research effort has been done to produce sonar maps for localization and navigation in indoor environments. In Elfes (1987) range measurements from multiple points of view taken from multiple sensors are integrated to build the sonar map. Wei et al. (1996) merges ultrasonic and vision sensors in order to produce an occupancy grid representation of the environment. Other authors like Nakamura et al. (1996), Chong and Kleeman (1999), Kleeman (2001, 2003) and Tardós et al. (2002) have done interesting work using sonar data.

However sonar readings are prone to several measuring errors due to various phenomena (e.g., multiple reflections, wide radiation cone, low angular resolution). Nowadays, laser range finders have replaced sonars when possible because of its superior efficacy in estimating distances accurately and their better signal to noise ratio. Many techniques have been developed to make the most of this type of sensor for solving the SLAM problem. Since a laser scan directly provides metric information of the scene, the localization problem can be stated in terms of an odometry-based method where the incremental displacement is found by computing the best rigid transformation that matches two successive scans. This is called the *scan matching* algorithm, in which, to match two scans it is necessary to link the individual measurements in one scan with the corresponding measurements in the other scan. This association can be done either using an intermediate representation of the laser data (e.g. a polygonal approximation as in Charbonnier and Strauss (1995)) or directly, by exploiting the raw data [Nieto et al. (2007)].

SLAM based on scan matching is not new, early work on the alignment of range-laser scans for localization and map building was done by Lu and Milios (1997a,b) and more recently by Nieto et al. (2005, 2007). Most of them are based on the Iterative Closest Point (ICP) algorithm developed by Besl and McKay (1992) for point to point scan matching, and in the Cox's pairwise scan matching algorithm proposed by Cox (1991) that matches points to line segments. The ICP algorithm is probably the most widely used matching algorithm and many extensions have been developed from it (more details in chapter 3).

Despite all the work that has been done to improve techniques to use lasers to solve the SLAM problem, the use of 2D lasers alone limits SLAM to planar motion estimation and does not provide sufficiently rich information to reliably identify previously explored regions. Vision sensors are a natural alternative to 2D laser range finders because they provide richer perceptual information and enable 6 degrees of freedom motion estimation.

One of the first attempts to solve the SLAM problem using monocular vision was the work by Broida et al. (1990). Since then faster computers and ways of selecting sparse but distinct features have allowed new approaches to emerge. Davison (2003) proposes a real-time approach that attempts to minimize drift by detecting and mapping only long-term landmarks during the SLAM process. This approach, however, is not appropriate for long displacements because of algorithmic complexity and growing uncertainty. Another interesting approach better suited for outdoor environments and large displacements was proposed by Mouragnon et al. (2006).

More recently, many other researchers have pursued research on vision-based SLAM with stereo and monocular approaches [Lemaire et al. (2007)], either relying on feature-based representations [Miro et al. (2005)] or on direct approaches [Silveira et al. (2007)]. Some vision-based solutions to the localization problem assume a piecewise planar scene and are based on a homography model as in Benhimane and Malis (2004) or Mei et al. (2008). Others methods use a stereo approach involving a quadrifocal warping function to close a nonlinear iterative estimation loop directly using images, leading to robust and precise localization [Comport et al. (2007, 2010)].

However, cameras are used less often than range scanners in solving the SLAM problem because extracting relevant information from images can be a computational intensive task, less attractive for real-time applications. Additionally, standard cameras only have a small field of view (typically between 40° and 50°) and can be easily affected by occlusion. It was shown in Davison et al. (2004) that a larger field of view (e.g. using fish-eye lens) makes it easier to find and follow salient landmarks. In contrast, using a catadioptric camera [Nayar (1997); Baker and Nayar (1998)] allow us to obtain a full 360° view of the environment. Image acquisition with these omnidirectional cameras has many advantages: it can be done in real time, it is easier to recognize previously observed places whatever the orientation of the robot is and it is also less likely that the robot gets *stuck* when facing a wall or an obstacle. Furthermore, in order to avoid the limitations due to planar projections, images captured by these cameras can be uniquely mapped to spherical images [Geyer and Daniilidis (2000)]. Thus, vision sensors provide dense and rich 3D information about the environment. Nevertheless, vision alone does not provide the depth information that a laser range finder does, which is crucial for solving the localization problem.

Many authors avoid the problems of monocular algorithms (i.e. scale factor, initialization, observability) by using multi-view constraints (see Hartley and Zisserman (2004) and references therein). Others try to complement, merge or fusion the information of different sensors. Cobzas et al. (2003) and Clerentin et al. (2000) complement visual information from an omnidirectional camera with depth information acquired by a laser range finder. Fu et al. (2007) extract environmental features from the monocular vision data and laser range finders in order to build metric maps, then fusion them using Kalman Filter and grid map building simultaneously.

2.2 Problem Statement and Notations

Referring to the notation proposed by Durrant-Whyte and Bailey (2006) and adapted by Mei (2007), let us consider a mobile robot moving through an environment taking relative observations of a number of unknown landmarks as shown in figure 2.1. Thus the following notations can be defined:

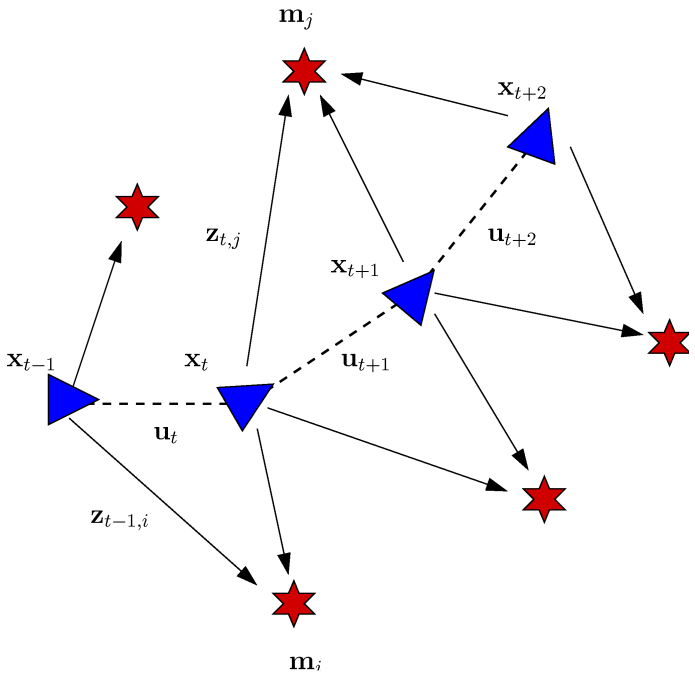


Figure 2.1: Notations for the SLAM problem [Mei (2007)]

- a discrete time index $t = 1, 2, ...,$
- x_t the true location of the robot at a time t,
- $\mathbf{u}_t$ a control vector applied at time $t-1$ to drive the robot from x_{t-1} to x_t at time t,
- m_i the position of the i^{th} feature or landmark,
- $\mathrm{z}_{t,i}$ an observation or measure of the i^{th} feature made in x_t at time t,
- z_t a generic observation of all the landmarks at time t.

The following sets can be defined as well:

- a history of past states: $\mathbf{X}_{0:t} = \{\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_t\} = \{\mathbf{X}_{0:t-1}, \mathbf{x}_t\}$
- a history of control inputs: $\mathbf{U}_{0:t} = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_t\} = \{\mathbf{U}_{0:t-1}, \mathbf{u}_t\}$

- the set of all landmarks: $\mathbf{m} = \{\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_M\}$
- the history of the landmark observations: $\mathbf{Z}_{0:t} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_t\} = \{\mathbf{Z}_{0:t-1}, \mathbf{z}_t\}$

In order to solve the SLAM problem, we assume that:

- the landmarks are static,
- no prior information is available on the features $\mathbf{m}$ that constitute the map,
- the initial position or origin $\mathbf{x}_0$ is known,
- the control sequence $\mathbf{U}_{0:t}$ is also known.

Hence, the problem is to build incrementally and simultaneously the map $\mathbf{m}$ and the set of positions $\mathbf{X}_{0:t}$ thanks to the observations or measures acquired by the robot. To illustrate this, we can refer to the *bearing-only* SLAM problem, where the relative angles between the robot and the landmarks are the only information available (see figure 2.2). The parametrization of the vectors $\mathbf{x}_t$ (robot state), $\mathbf{u}_t$ (control input), $\mathbf{z}_t$ (sensing observations) will be then:

$$\mathbf{x}_t = \begin{bmatrix} x_t & y_t & \theta_t \end{bmatrix}^T$$

$$\mathbf{m}_i = \begin{bmatrix} x_i & y_i & \theta_i \end{bmatrix}^T$$

$$\mathbf{z}_{t,i} = \begin{bmatrix} \alpha_{t,i} & \beta_{t,i} \end{bmatrix}^T$$

$$\mathbf{u}_{t,i} = \begin{bmatrix} V_t & \omega_t \end{bmatrix}^T$$

where V_t is the velocity vector expressed in the robot frame and ω_t is angular velocity at time t.

In the robotics community, there are two major formulations to the SLAM problem. The following sections will be devoted to state these two approaches: ***Probabilistic Approach*** and ***Optimization Approach***.

2.3 Probabilistic Approach

Since the seminal work of Smith and Cheeseman (1986) and Smith et al. (1986) stochastic mapping has become the dominant approach to SLAM, because SLAM can be formally best described in terms of probability, where the probability distribution[1]:

[1] For a brief review on fundamentals of probability theory see Appendix B

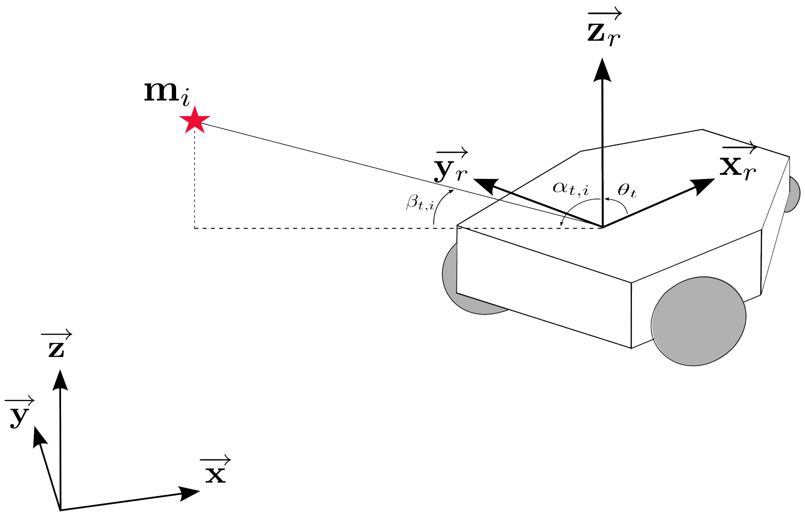

$\overrightarrow{\mathbf{z}_r}$
$\mathbf{m}_i$
$\overrightarrow{\mathbf{y}_r}$
$\overrightarrow{\mathbf{x}_r}$
$\alpha_{t,i}$
θ_t
$\beta_{t,i}$
$\overrightarrow{\mathbf{z}}$
$\overrightarrow{\mathbf{y}}$
$\overrightarrow{\mathbf{x}}$

Figure 2.2: Bearing-Only SLAM

$$P(\mathbf{x}_t, \mathbf{m}|\mathbf{Z}_{0:t}, \mathbf{U}_{0:t}, \mathbf{x}_0) \tag{2.1}$$

describes the joint posterior density of the vehicle state and landmarks knowing all the observations and control inputs given to the robot. This probability distribution need to be computed for all times t in order to solve the SLAM problem. Written in this way it is also known as the *full SLAM problem*.

2.3.1 Recursive Formulation

In order to state a recursive solution to the SLAM problem some assumptions have to be done. Considering a Markovian process, i.e., the present state x_t depends only in the immediately precedent state x_{t-1} and assuming that the pose is independent of the observation and the map, we can write:

$$P(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t) = P(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t, \mathbf{X}_{0:t-2}, \mathbf{U}_{0:t-1}, \mathbf{m}) \tag{2.2}$$

Equation 2.2 is called the *motion model*. Under the above assumptions and applying the Bayes' rule the *time update equation* is defined as:

$$\begin{aligned}
P(\mathbf{x}_t, \mathbf{m}|\mathbf{Z}_{0:t-1},\mathbf{U}_{0:t}, \mathbf{x}_0) &= \int P(\mathbf{x}_t, \mathbf{x}_{t-1}, \mathbf{m}|\mathbf{Z}_{0:t-1}, \mathbf{U}_{0:t}, \mathbf{x}_0) d\mathbf{x}_{t-1} \\
&= \int P(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{m}, \mathbf{Z}_{0:t-1}, \mathbf{U}_{0:t}, \mathbf{x}_0) P(\mathbf{x}_{t-1}, \mathbf{m}|\mathbf{Z}_{0:t-1}, \mathbf{U}_{0:t-1}, \mathbf{x}_0) d\mathbf{x}_{t-1} \\
&= \int P(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{u}_t) P(\mathbf{x}_{t-1}, \mathbf{m}|\mathbf{Z}_{0:t-1}, \mathbf{U}_{0:t-1}, \mathbf{x}_0) d\mathbf{x}_{t-1} \qquad (2.3)
\end{aligned}$$

In practice, it is reasonable to assume that the measurements are conditionally independent, that is, sensor noise is uncorrelated over time, which can be expressed as:

$$P(\mathbf{Z}_{0:t}|\mathbf{X}_{0:t},\mathbf{m}) = \Pi_{i=1}^{t} P(\mathbf{z}_i|\mathbf{X}_{0:t},\mathbf{m})$$

Applying Bayes' rule to expand the joint distribution in terms of the state and then in terms of the landmark observations gives the following two equalities:

$$\begin{aligned} P(\mathbf{x}_t,\mathbf{m},\mathbf{z}_t|\mathbf{Z}_{0:t-1},\mathbf{U}_{0:t},\mathbf{x}_0) &= P(\mathbf{x}_t,\mathbf{m}|\mathbf{Z}_{0:\mathbf{t}},\mathbf{U}_{0:t},\mathbf{x}_0)P(\mathbf{z}_t|\mathbf{Z}_{0:t-1},\mathbf{U}_{0:t},\mathbf{x}_0) \\ P(\mathbf{x}_t,\mathbf{m},\mathbf{z}_t|\mathbf{Z}_{0:t-1},\mathbf{U}_{0:t},\mathbf{x}_0) &= P(\mathbf{z}_t|\mathbf{x}_t,\mathbf{m})P(\mathbf{x}_t,\mathbf{m}|\mathbf{Z}_{0:t-1},\mathbf{U}_{0:t},\mathbf{x}_0) \end{aligned}$$

where $P(\mathbf{z}_t|\mathbf{x}_t,\mathbf{m})$ is called the *observation model*. By combining these equations, we obtain the *measurement update equation* as:

$$P(\mathbf{x}_t,\mathbf{m}|\mathbf{Z}_{0:t},\mathbf{U}_{0:t},\mathbf{x}_0) = \frac{P(\mathbf{z}_t|\mathbf{x}_t,\mathbf{m})P(\mathbf{x}_t,\mathbf{m}|\mathbf{Z}_{0:t-1},\mathbf{U}_{0:t},\mathbf{x}_0)}{P(\mathbf{z}_t|\mathbf{Z}_{0:t-1},\mathbf{U}_{0:t})} \tag{2.4}$$

Finally, from equations 2.3 and 2.4, the recursive formulation to the SLAM problem,

$$P(\mathbf{x}_t,\mathbf{m}|\mathbf{Z}_{0:t},\mathbf{U}_{0:t},\mathbf{x}_0) = \eta P(\mathbf{z}_t|\mathbf{x}_t,\mathbf{m}) \int P(\mathbf{x}_t|\mathbf{x}_{t-1},\mathbf{u}_t)P(\mathbf{x}_{t-1},\mathbf{m}|\mathbf{Z}_{0:t-1},\mathbf{U}_{0:t-1},\mathbf{x}_0)d\mathbf{x}_{t-1}$$

with η a normalizing constant, is a function of the motion model and the observation model.

Probabilistic Solutions to the SLAM Problem

There are many different methods to attack the probabilistic SLAM problem. Recursive solutions, where the current pose of the robot is computed processing one data item at a time, are the most used ones. In the literature, such algorithms are called *filters*. The next part of this section will describe the Kalman Filter and the Particle Filter for been the most popular ones in the robotics community. In the same way some works are presented to illustrate the basic concepts.

2.3.2 Kalman Filter

In Kalman (1960) famous paper, a recursive solution to the discrete data linear filtering was proposed, refined later by Kalman and Bucy (1961). It has opened new routes for research and applications in mobile robot autonomous navigation. The simple and robust nature of this recursive algorithm has made it particularly appealing to solve the SLAM problem.

Kalman filter's first approach (see figure 2.3) requires a linear model of the system evolution over time, a linear relationship between the state and the measurements and zero mean noise (White Gaussian Noise) to ensure optimality. If these conditions are met, the Kalman filter provides the provably optimal method –in a least square sense– for fusing data. An extension of the Kalman Filter to cope with non-linearity is the so called *Extended Kalman Filter (EKF)* which simply linearizes all nonlinear models so that the linear Kalman filter can be applied. It is not frequently mentioned

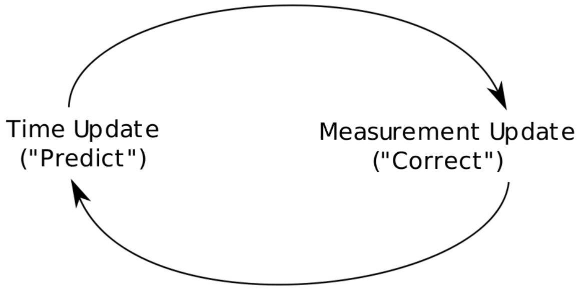


Figure 2.3: Kalman Filter Cycle

in literature, but the Extended Kalman filter was implemented by Stanley Schmidt. The former name of this filter was Kalman-Schmidt filter.

The Extended Kalman Filter, as a solution for the SLAM problem, was introduced in the papers by Smith and Cheeseman (1986); Smith et al. (1986) and some of the first implementations were done by Moutarlier and Chatila (1989, 1990) and Leonard and Durrant-Whyte (1991). Indeed, EKF SLAM is perhaps the most used SLAM algorithm in the literature.

Kalman Filters can be viewed as a variant of a Bayesian Filter, thus, the state $\mathbf{x}_t$ of the system at time t can be considered a random variable where the uncertainty about this state is represented by a probability distribution as in equation 2.1. In Appendix A, the equations of the discrete Kalman filter and the EKF are described in detail.

Even though it is rare that the optimal conditions exist in real-life applications, some assumptions can be relaxed, yielding a qualified optimal filter. The results in practice are quite satisfactory; some positive results of the performance of the EKF to solve the SLAM problem are depicted in [Dissanayake et al. (2001)]. They proved three important convergency properties of the EKF solution to SLAM, namely that:

1. The determinant of any submatrix of the map covariance matrix decreases monotonically as observations are successively made.

2. In the limit as the number of observations increases, the landmark estimates become fully correlated.

3. In the limit, the covariance associated with any single landmark location estimate reaches a lower bound determined only by the initial covariance in the vehicle location estimate at the time of the first sighting of the first landmark.

In other words, they show that the entire structure of the SLAM problem depends mostly on maintaining complete knowledge of the cross correlation between landmark estimates, and in fact, the EKF maintains a complete covariance matrix and mean vector for all of the features, which is important for the data association problem (recognizing when two observations belong to the same feature).

Nevertheless, there are also some important issues when using EKF for SLAM:

- The linearization used in the EKF leads to inconsistencies in the solution because there is no guarantee that the computed covariances will match the actual estimation errors, which is the true SLAM consistency issue. Convergence and consistency of the filter have only been shown in the linear case. [Bailey et al. (2006a,b)]

- A basic implementation of this filter is quadratic ($O(n^2)$ where n is the number of features) in time and memory usage, which means considerable computational effort. The observation update equation requires an update of the landmark poses and joint covariance. In addition, for each new landmark, the correlation with all the values of the state vector must be saved. This limits the application of EKF SLAM to small scale environments with a few hundred features.

- Data association problem, which measurement observation correspond to which landmark? This difficulty is also enhanced by the inconsistencies introduced by the linearization and some processes in SLAM like loop closing are crippled. Data association can be solved by Nearest Neighbor Gating (NNG) or by joint compatibility test (JCT) as shown by Neira and Tardos (2001) or by tree search as in Arras et al. (2002).

These problems have been thoroughly studied over the last decades, generating different new versions of the EKF in order to improve its performance and the results. Among the most popular variants we have the *Unscented Kalman filter* (UKF) which avoids linearization via parametrization of means and covariances through selected points to which the nonlinear transformation is applied [Julier and Uhlmann (1997)]. This method gives better results than the standard EKF but the problems due to linearization remains. Unscented SLAM improved consistency properties as shown by Martinez-Cantin and Castellanos (2005), but ignore the computational complexity problem.

It is a well known fact that to enable real-time mapping of large environments, the problem of computational complexity have to be taken into consideration. The methods used to reduce computational complexity requires a re-formulation of the time and observation update equations. Most of the authors exploit the sparsity in the dependencies between the local robot position and distant landmarks to build local maps. The ATLAS framework for large scale SLAM proposed by Bosse et al. (2003), achieves real-time performance in large indoor structured environments but does not compute the state estimate in a global reference frame. Leonard and Feder (2000) propose the Decoupled Stochastic Mapping (DSM) which reduces the computational complexity by dividing the environment into multiple overlapping submap regions (each one with its own stochastic map) achieving constant-time updates, but the solution does not guarantee consistency. Later, Leonard and Newman (2003) propose a consistent and constant-time SLAM approach which also ensures convergence, but considers data association known, which is a major assumption.

Julier and Uhlmann (2001) and Bailey et al. (2006a) have studied and proved the causes of the significant inconsistency of the EKF-SLAM formulation; they agreed that inconsistency can be prevented if the jacobians for the process and observation models are always linearized about the true state. It is important to remark that actually, convergence and consistency have only been proved for the linear case.

Another difficult problem to solve in EKF-SLAM frameworks is the ambiguity in data association. The problem is that a single incorrect data association can induce considerable drift on the localization estimation and therefore divergence into the map estimate. Tardós et al. (2002) propose a solution using Hough transform obtaining remarkable results on sonar data. More recently, Paz et al. (2007) propose the Divide and Conquer (D&C) SLAM algorithm which reduces the computational complexity of the EKF-SLAM; furthermore, to limit the computational cost of data association they developed the Randomized Joint Compatibility (RJC) which is a variant of the Joint Compatibility Test proposed earlier by Neira and Tardos (2001).

Other authors combine different data association techniques with EKF-SLAM to reduce the ambiguity of local associations for large outdoor environments. Bosse and Zlot (2008) use a robust iterative scan matching technique in order to build local maps. On the other hand, Nebot et al. (2003) used particle filters combined with EKF-SLAM to resolve the data association problem that is presented when returning to a known location after a large exploration task. This problem is generally referred as *loop closing* in the literature.

2.3.3 Particle Filter

As we have stated before, the Kalman Filter approach comes with a number of limitations. An alternative approach is to obtain an approximate estimate of the posterior Probability Density Function (PDF, see Appendix B.2) using samples. The technique of drawing (randomly) state samples from the prior distribution and using these samples (in conjunction with state transition and observation information) to approximate the posterior is known as *particle filtering*.

Particle Filters (PF), also known as Sequential Monte Carlo methods, are non-parametric implementations of the Bayes' filter and are frequently used to estimate the state of a dynamic system. They were firstly introduced by Handschin and Mayne (1969) and later a similar approach were presented in Akashi and Kumamoto (1977). Since then, several PF algorithms started to appear in the literature under many different names such as Sequential Importance Sampling (SIS) and Sequential Importance Resampling (SIR) Filters [Rubin (1988)] which is the same as SIS, but with a resampling step; Monte Carlo filters [Kitagawa (1996)]; condensation algorithm [Isard and Blake (1996)]; Bootstrap filters [Gordon et al. (1993)]; Dynamic mixture models [West (1993)]; survival of the fittest [Kanazawa et al. (1995)]; etc.

The idea of particle filters with SLAM was presented by Montemerlo (2003) and it is called Fast-SLAM. This algorithm uses Rao-Blackwellised particle filter [Doucet et al. (2000)] to solve the SLAM problem efficiently. Using Fast-SLAM algorithm, the posterior estimation will be over the robot's pose and landmarks locations. Thus, if we assume that a single measurement is obtained at a given time step (which does not affect the generality of the approach) we can rewrite equation 2.1 as:

$$P(\mathbf{x}_t, \mathbf{m}|\mathbf{Z}_{0:t}, \mathbf{U}_{0:t}, \mathbf{C}_{0:t}, \mathbf{x}_0) \tag{2.5}$$

where $\mathbf{C}_{0:t}$ correspond to the set of associations,

$$\mathbf{C}_{0:t} = \{\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_t\}$$

If the full trajectory (all the poses $\mathbf{X}_{0:t}$ are known), we have a simple mapping problem with conditionally independent landmarks and equation 2.5 can be written as:

$$P(\mathbf{X}_{0:t}, \mathbf{m}|\mathbf{Z}_{0:t}, \mathbf{U}_{0:t}, \mathbf{C}_{0:t}, \mathbf{x}_0) = P(\mathbf{m}|\mathbf{Z}_{0:t}, \mathbf{U}_{0:t}, \mathbf{C}_{0:t}, \mathbf{x}_0)P(\mathbf{X}_{0:t}|\mathbf{Z}_{0:t}, \mathbf{U}_{0:t}, \mathbf{C}_{0:t}, \mathbf{x}_0)$$

The Fast-SLAM algorithm has been implemented successfully over thousands of landmarks in contrast to EKF-SLAM that can only handle a few hundreds. Moreover, FastSLAM achieves an $O(M\log(N))$ complexity, with N the number of features and M the number of particles. If the reader is interested, a detailed comparison between EKF-SLAM and Fast-SLAM can be found in Calonder (2010).

Despite the many advantages offered by the FastSLAM algorithm in terms of complexity, it also has disadvantages in terms of consistency. Bailey et al. (2006b) show that FastSLAM can not ensure long-term consistency. The main problem is that the particle filter is operating in the space of vehicle trajectories and not momentary poses, which means a very high-dimensional space. Thus, the number of particles needed are exponential in the length of the trajectory. If a smaller number of particles are used, the filter underestimates the total uncertainty and eventually becomes inconsistent. Even so, this may still produce good maps but when large loops need to be closed and full uncertainty is required the problems start.

2.4 Optimization Approach

An alternative to the probabilistic formulation for the SLAM problem consists to state the problem in terms of optimization (minimization) of a cost function. The problem is stated as a parametric estimation problem and it requires to define:

- A parametric model of the perception function $z = h(\mathbf{x}, \mathbf{m})$
- A cost function $d(h(\mathbf{x}, \mathbf{m}), \mathbf{z}_{obs})$

- An efficient optimization method for finding $(\widehat{\mathbf{x}}, \widehat{\mathbf{m}})$ such that,

$$min_{x,m}\{d(h(\mathbf{x}, \mathbf{m}), \mathbf{z}_{obs})\}$$

Optimization approaches are simple and robust methods capable to work directly on raw data, moreover, no feature extraction or matching steps are needed. However, the downside is the need of an initialization sufficiently close to the true solution and for the method to work, a model of the perception process also need to be available.

Optimization Solutions to the SLAM problem

Optimization methods require the entire data set, i.e., all the data is processed at the same time. These methods are usually of batch type (generally maximum likelihood approaches) like bundle adjustment and Expected Maximization (EM). The following sections will describe more in detail the most important techniques used to solve the SLAM problem with an optimization approach, namely bundle adjustment, EM, Iterative Closest Point (PIC) and Sum of Squared Differences (SAD).

2.4.1 Bundle Adjustment

Bundle adjustment was originally developed in the field of photogrammetry during fifties for the U.S. Air Force by Brown (1958, 1976). At the time, the main objective was mostly for aerial cartography but recently it has increasingly been used by computer vision researchers because the bundle adjustment algorithm can be used to handle different types of image features (points, lines, curves, surfaces etc), different camera models and autocalibration parameter sets.

Bundle adjustment is an iterative (batch update) method in which one attempts to fit a nonlinear model to the measured data assuming that the data association have been made. Bundle adjustment can be seen as an optimal solution to the SLAM problem if implemented in such a way that the discovered solution is the Maximum Likelihood[1] solution given all measurements over all time. In vision, data can be defined as points projected in several views and matched by correlation, thus, bundle adjustment would consist in minimizing the reprojection error to recover the camera and point positions [Triggs et al. (1999)]. This error is expressed as the sum of squares of a large number of nonlinear, real-valued functions. Therefore, the problem is typically tackled with nonlinear least-squares optimization algorithms like Levenberg-Marquardt or the Gauss-Newton method.

Unfortunately bundle adjustment is an iterative process and cannot ensure convergence to the optimal solution from an arbitrary starting point or for mapping large and cycle environments. In a single coordinate frame, the farther the robot travels from the origin, the larger position uncertainty becomes. Errors at loop closure can

[1] Reminder: Maximum Likelihood estimation wishes to estimate the model parameter(s) for which the observed data are the most likely.

therefore become arbitrarily large, which in turn makes it impossible to compute the maximum likelihood solution in constant time. This is why bundle adjustment is used only as a "local" optimization method or as the last step of any feature-based 3D reconstruction algorithm.

More recently, interesting efforts by Ni et al. (2007) and Mouragnon et al. (2009) were made to use bundle adjustment for large-scale reconstruction. Basically, they decouple the original problem into several submaps that have their own local coordinate systems and can be optimized in parallel.

2.4.2 Expectation Maximization

The Expectation Maximization (EM) algorithm is also an iterative (batch update) method that computes the Maximum Likelihood estimate in the presence of missing or hidden data. Its name was given in a classic paper by Dempster et al. (1977), in which they also explained the process.

In the basic EM algorithm each iteration consist of two processes: The *Expectation Step* (E-step) and the *Maximization Step* (M-step). In the E-step, the missing data are estimated given the observed data and current estimate of the model parameters. In the M-step, the likelihood function is maximized under the assumption that the missing data are known. Convergence is assured since the algorithm is guaranteed to increase the likelihood at each iteration.

Expectation Maximization is not designed for the SLAM framework since the data sets are too large and it becomes impractical due to the requirement that the whole data have to be available at each iteration of the algorithm. Nevertheless, EM can be used as an alternative approach to solve the correspondence problem. Since building the map with a known robot path is relatively simple, the algorithm separates the estimation of the global posterior over the poses and the map by two optimization steps: Firstly, the E-step calculates the posterior over the robot poses for a given map. Secondly, the M-step calculates the most likely pose given the robot poses. The algorithm has proved to give good results [Thrun et al. (1998a)], but as any other nonlinear optimization method, the approach risk to fall in a local maxima and is computationally expensive.

A modified version of EM was presented by Thrun et al. (2004), which is capable of generating online maps, while the robot is in motion assuming the robot pose is available. This approach retains the ability to revise past components based on future data while simultaneously restricting the computation in ways that make it possible to run the algorithm in real-time, regardless of the size of the map.

2.4.3 Iterative Closest Point (ICP)

The ICP algorithm is an iterative alignment algorithm, it has been widely used in many robotics applications, namely, for localization and path planning, to reconstruct

2D or 3D surfaces from scans and for mapping in different scenarios. The general ICP algorithm was proposed by Besl and McKay (1992). His method was proved to compute the closest point to a given point on various geometric representations such as point sets, line segment sets (polylines), implicit curves, parametric curves, triangle sets, implicit surfaces and parametric surfaces, which basically covered most of the applications that would utilize a method to register shapes.

The idea of any pose estimation algorithm that uses ICP is depicted in figure 2.4. As input data, the ICP algorithm requires a set of entities defining the model and a set of data measurements acquired by a sensor device. The first step is to relate the sensor measurements to its corresponding model by a given correspondence search criterion. The obtained correspondences are the input for the minimization step used to compute the pose parameters. The minimization iteratively revises the transformation (translation, rotation) needed to minimize the distance between pairs of features until convergence.

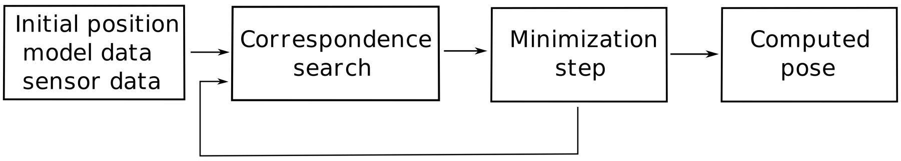


Figure 2.4: ICP algorithm

The algorithm is conceptually simple and is commonly used in real-time. It is important to remark that for the algorithm to converge, the starting position must be "close enough" to the solution (see figure 2.5).

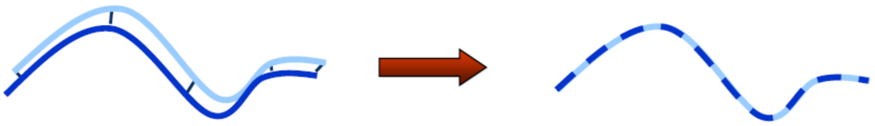


Figure 2.5: ICP alignement

Some variants to improve the ICP algorithm have been proposed, such as point subsets (from one or both point sets), weighting the correspondences, data association, rejecting certain (outlier) point pairs. These variants improve the performance of the algorithm over speed, stability (local minima) and tolerance with respect to noise and outliers. More details in chapter 3.

2.4.4 Sum of Squared Differeces (SSD)

Sum of squared differences (SSD) tracking can be traced back to the work by Lucas and Kanade (1981) and later Shi and Tomasi (1994) with the KLT tracker. Basically, SSD measures the difference in intensity between a portion of the first image reprojected onto the second image. The minimization is based on the image gradient and can be 2D imaged-based (e.g., searching for the translation (t_x, t_y) that gives the smallest reprojection error). It can also be 3D or model-based by reprojecting a 3D object and minimizing the difference in the image over the position (6 degrees of freedom: rotation and translation).

The advantage of this approach is precision and speed. This is why these techniques are particularly well adapted to robotic tasks such as motion estimation and visual servoing. Compared to matching approaches, SSD tracking is generally faster and more precise. The downside is the need for a strong overlap between the reprojected and the real object for the system to converge.

2.5 The Representations of the Environment

So far, we have presented some of the major problems and solutions to the SLAM problem, but for the robot to move autonomously and safely, it is necessary to know the environment where it moves.

In order to solve the SLAM problem, some form of the environment modeling is required. This is called the mapping problem, in which, it is important to define what we want to represent and how. This problem has been extensively studied in the robotics community and many methods of constructing maps have been proposed. A general survey on robotic mapping can be found in [Thrun (2002)]. The aim of this section is to present some of them.

The main representations of maps are classified into three big categories:

- Metric: Metric maps represent geometric properties of the environment in an Euclidean world.
- Topological: Topological maps are usually represented as graphs that describe the connectivity between locations.
- Hybrids: Based in mixed methods with topological, metric and probabilistic characteristics.

2.5.1 Metric Maps

In a metric map, the environment is represented by a set of objects, which positions are associated to a metric space. This type of maps are often related to sensors that can provide a measure of distance between the robot and the objects in the environment such as ultrasonic sensors or laser range finders.

One characteristic of metric maps is the use of information from exteroceptive and proprioceptive sensors (see sections 2.6 and 2.7 for details about sensors). The data provided by proprioceptive sensors is use to estimate the position of the robot defining the metric space, while the data provided by exteroceptive sensors allow to

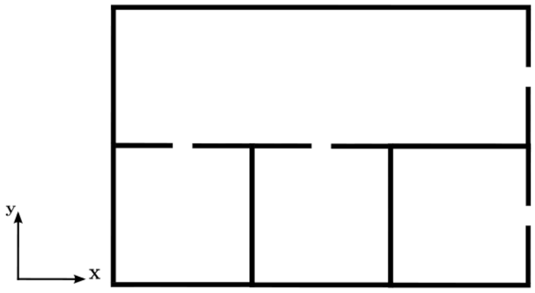


Figure 2.6: Metric map with lines as features

detect objects in the environment and to estimate its relative position to the robot using the metric model of the sensors. However, there are always problems related to sensor inaccuracies and the metric model of the sensors is not always available or easy to obtain, which is a weak point when using metric maps.

Metric maps have many advantages over topological maps. They can robustly map large-scale environments, while topological maps have difficulties to construct and maintain in large-scale environments if sensor information is ambiguous. This permits to estimate accurately and continuously the position of the robot in the environment. Moreover, this representation is not limited to the explored positions, but to all the areas that the robot has been able to sense from the places that it has visited, which allows a more complete map of the environment.

In the literature there are two main approaches to represent metric maps, one based on the representation of geometric primitives (features) and the other based on an occupancy grid representation.

Feature Maps

In metric feature maps (or landmark maps), the environment model is built from the geometry of the elements that compose it. In general, the construction of these maps follows three basic steps: first, the choice of the geometric primitives, which will constitute the basic elements of the representation; secondly, the characterization of these elements in the current observation; and finally the update of the global map (under construction) by incorporating the elements of the current local observation.

Geometric features are often chosen according to the sensor and the environment that should be explored. Some of the geometric representations mostly used for SLAM are:

- *Points*: Vision sensors mostly use points as geometric primitive, allowing to easily characterize indoor or outdoor environments [Davison (2003)].
- *Lines*: There are mostly used to characterize indoor and structured environments. For robots with laser range finders it is a normal choice because they

provide a relatively dense representation of these environments [Chatila and Laumond (1985), Lu and Milios (1997a)].

- *Planes*: Generally used to represent complex structures as they provide a 3D representation of the environment. Thrun et al. (2004) successfully generate a 3D map of an indoor environment from laser rangefinder and vision information. The planes were extracted using a real-time variant of the expectation-maximization algorithm.

Chong and Kleeman (1997) propose a feature-based mapping strategy where discrete planar and corner elements gathered by a sonar sensor are merged incrementally to form partial planes to produce a realistic representation of environment. In Castellanos et al. (1998), geometric primitives like straight line, corner and half-plane are extracted from laser range data and vertical segments (lines) are extracted from the images provided by a camera. The vertical segments are mapped to the primitives from the laser, allowing better accuracy in the calculation of parameters of the primitives. Localization and mapping are then carried out by an extended Kalman filter.

In general the maps generated with this approach may be more compact than occupancy grid maps, especially if the environment is structured. Furthermore, if the geometric primitives are chosen correctly the maps are more accurate and often closer to the perception that people have of their environment. The principal problem concerning feature maps is their suitability only to environments where the observed objects can be represented by basic geometric feature models. This is not the case for unstructured environments where the objects might appear as curves rather than distinct point or lines. A alternative to construct feature maps of unstructured environments is to parameterize feature models that depict the observed objects well enough to correctly extract the features.

Occupancy Grid

Occupancy (or evidence) grids developed by Moravec and Elfes (1985), represent the space around the robot as a 2D [Elfes (1990)] or 3D [Moravec (1996)] grid of regular cells. Each cell is associated with a measure modeled by a probability function, describing the potential presence of an object in the corresponding position. As originally proposed by Moravec and Elfes (1985), the occupancy is a binary variable (see figure 2.7): Either the cell is occupied (black cells) or it is free (white cells). However, later some authors used a third variable with the *"unknown"* status as recently shown in Carpin (2008).

The occupancy grid maps has gained great popularity because they are very robust, easy to implement and allows to use raw sensor information. They are particularly suitable for navigation, path planning, detection and obstacle avoidance and tracking of objects. Sonars and laser, are the most used sensors to build occupancy

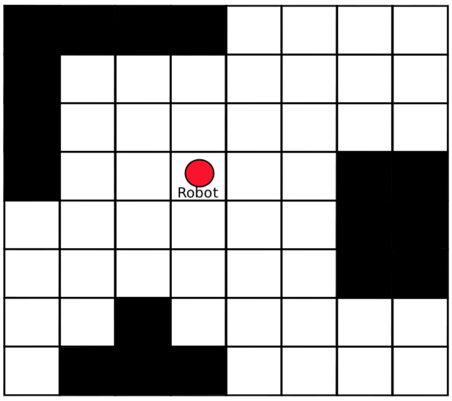


Figure 2.7: Occupancy Grid Map of the robot surroundings

grid maps. Diosi et al. (2005) generate occupancy grid maps by fusing laser and sonar measurements in a Kalman filter based SLAM framework. Later this maps are used for localization.

The major disadvantage with occupancy grid maps is the trade-off between grid resolution (granularity) and computational complexity. If we wish detailed maps, the grid size need to be as small as possible, increasing the number of grid cells and therefore, the computational complexity.

2.5.2 Topological Maps

Topological maps represent the environment without the use of any metric information. They are generally represented as a graph with nodes (or vertices) and edges which in turn are used to represent the topological properties of places as neighborhood, inclusion, connectivity and order. The nodes characterize particular places, (i.e., the positions that the robot can reach) and the edges between nodes define the pathways that allow the robot to move from one place to another and to memorize how to perform this displacement. The displacement between two non-adjacent places is determined by a sequence of transitions between the intermediate nodes.

One of the advantages of using topological maps is that it is possible to use standard graph algorithms for high-level planning operations such as finding the shortest path between non-adjacent nodes. For example, from fig 2.8, the shortest path to go from **a** to **g** requires traveling through the sequence **a-b-c-e-d-g**.

Commonly, specific attributes can be associated to the nodes and edges depending on the navigation task to perform [Kuipers and Byun (1991)]. For example, to localize the robot, the nodes can be provided of geometrical characteristics or color. Likewise, for navigation strategies, the nodes can be attributed with logical states, i.e., if a node have been already visited or if it is new. As the edges provide the necessary information to travel between nodes, they can be attributed with the sense of the path to follow for an example.

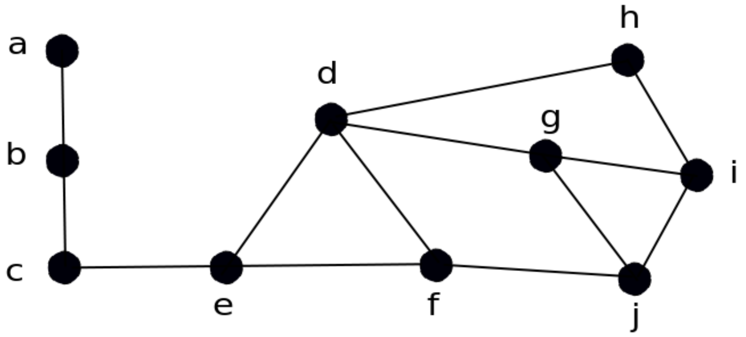


Figure 2.8: Topological Map

Topological maps do not require a metric model of the sensors in order to fusion proprioceptive and exteroceptive data into an unified representation of the environment. However, an exhaustive exploration of the environment and very precise sensors are needed to obtain a good representation of the environment. On the other hand, the principal weakness of topological maps is to ensure reliable navigation between places without the aid of some form of metric measure. Consequently, topological maps show good performance for small and static environments, but fails or gives a false positive for dynamic or more complex and large environments.

These maps are widely used for pattern recognition for its compact and efficient representation as an *a priori* reference. In Choset and Nagatani (2001), the topology of the environment is encoded in a topological map called *generalized Voronoi graph* (GVG), that also encodes some metric information about the robot's environment. The graph is constructed incrementally during the exploration of the environment and the SLAM problem is reduced to a graph matching problem at a topological scale.

The work of Blanc et al. (2005), proposes a navigation framework in which the robot performs paths that are stored as a set of ordered key images acquired by a standard camera. These keys images compose a visual memory of the environment that are later structured as a graph, taking into consideration the environment topology to build a topological map. Their work was later extended by Courbon et al. (2007), to the entire class of central cameras (conventional and catadioptric cameras).

2.5.3 Hybrid Maps

The idea of integrating topological and metric representations was proposed by many researchers as Kuipers (1978) and Chatila and Laumond (1985), because the qualities between metric and topological maps are complementary. The famous framework: first topological then metric, was first propose by Kuipers (1978) and later implemented by Kuipers and Byun (1991). Since then it has been the framework used by many researchers as Thrun et al. (1998b), Victorino et al. (2003a,b) and Victorino and Rives (2004). Other idea was to connect local metric maps by means of a global topological map proposed by Tomatis et al. (2001).

Nieto et al. (2004) propose the Hybrid Metric Maps (HYMMs) framework which is a mapping algorithm that combines feature maps with occupancy grid maps as a solution for the dense SLAM problem obtaining interesting results for outdoor environments. Another interesting hybrid approach was proposed by Thrun (1998), which combines grid-based and topological maps. On the one hand, grid-based maps are learned using artificial neural networks and Bayes rule. On the other hand, topological maps are generated by partitioning the grid-based map into critical regions. He proves practically the efficiency of his method.

In fact, to build a map the robot must posses sensors that enable it to perceive the environment around it. The sensors more commonly used to accomplish this task include cameras, laser range finders, sonars, radars, GPS, and infrared technology. In the following sections a detailed description of the sensors used in this thesis is presented.

2.6 Proprioceptive Sensors

Proprioceptive sensors provides information about the robot's position/movement in space. This information is also called *idiothetic*, and it is crucial for the navigation of the robot. Idiothetic information may come from the measure of the rotation of the wheels or the acceleration's measure from an inertial measurement unit (IMU). An integration process allows then, by collecting this information over time, to estimate the relative position of two frames where the robot moved (dead reckoning). However, because of this integration process the quality of the information with this kind of sensors degrades continuously over time (drift), which makes the position estimation unreliable at long-term. Despite this limitation, idiothetic information has the advantage that it does not depend at all on environmental conditions that strongly disturb the perception of information.

In this thesis, only the wheels encoders are used as proprioceptive sensors. The odometry provided by the robot will be used as an initialization of the SLAM process.

2.6.1 Wheels Encoders (Odometry)

The most used method for mobile robot positioning is odometry because it provides good short-term accuracy, is inexpensive and allows very high sampling rates. The robots used in this work are moved by a differential wheel configuration, using two servo drives mounted in a common axis. Each wheel can independently being driven either forward or backward. The speed difference between both wheels results in a rotation of the vehicle about center of the axle. Hannibal has only one castor (see figure 2.9(a)) and Anis has two castors (see figure 2.9(b)). The castors serve to support the weight of the vehicle and under ideal conditions do not affect the kinematics of the robot.

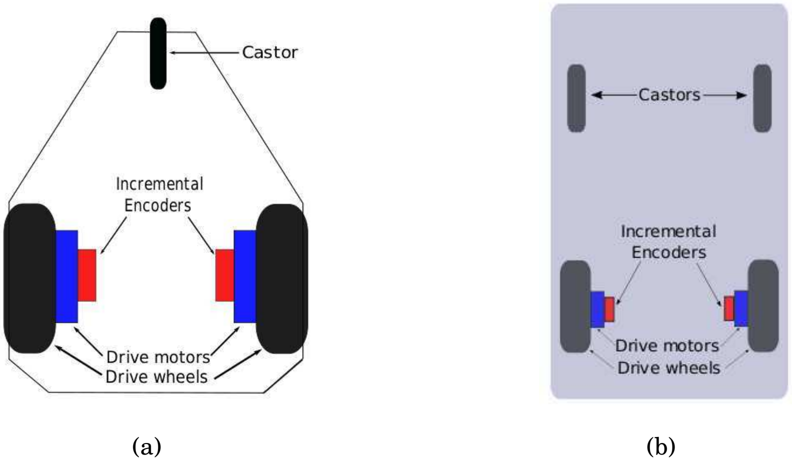


Figure 2.9: Differential drive system. 2.9(a) Hannibal. 2.9(b) Anis.

The position of the robot between two instants k and $k+1$ (i.e. the kinematic model) with the measurement of the displacement of the wheels (d_{rk}, d_{lk}) between those instants, can be described in cartesian coordinates as shown in figure 2.10. Under the assumption that the frequency of acquisition of the encoder wheel is large enough to observe small incremental movements of the robot and that the movement takes place without slipping, then we can write:

$$\begin{aligned} x_{k+1} &= x_k + d_k \cos(\theta_k + \frac{\Delta\theta_k}{2}) \\ y_{k+1} &= y_k + d_k \sin(\theta_k + \frac{\Delta\theta_k}{2}) \\ \theta_{k+1} &= \theta_k + \Delta\theta_k \end{aligned} \tag{2.6}$$

where:

$$\begin{aligned} d_k &= \frac{d_{rk} + d_{lk}}{2} \\ \Delta\theta_k &= \frac{d_{rk} - d_{lk}}{D} \end{aligned}$$

and D is the half distance between the wheels.

In practice, the model (equation 2.6) only gives an estimation of the position of the robot because the assumptions are of limited validity and there are always measurement noise. In order to determine the accuracy of the state estimation based on encoder data, it is important to identify odometry errors. As mentioned earlier, odometry is the integration of incremental motion information over time and, as any other dead-reckoning method, errors will result in a drift of the estimation of the po-

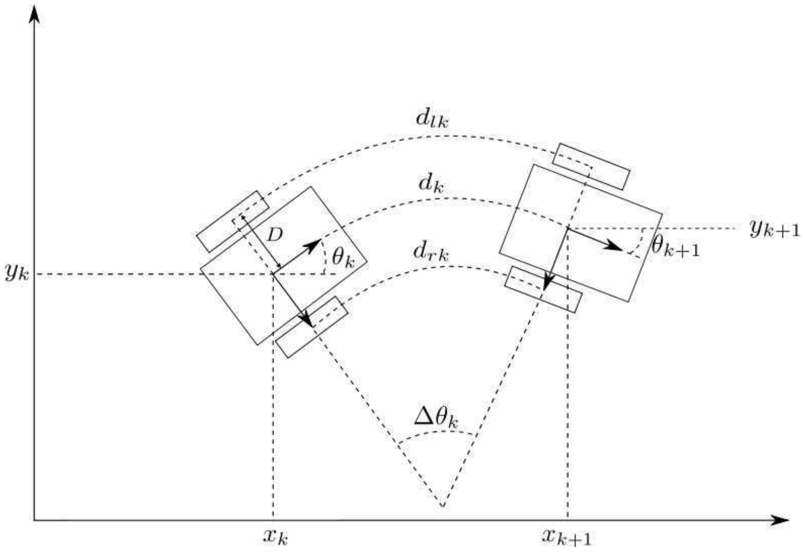


Figure 2.10: Kinematic model.

sition of the robot. Errors in odometry can be either *systematic* or *non-systematic* (see table 2.1).

The defects in the mechanical design of a mobile robot are the principal cause of systematic errors. Consequently, this errors are present in every integration step and accumulates constantly. Nevertheless, this errors can be identified and corrected. Borenstein and Feng (1996) propose a calibration technique called *UMBmark* test, developed to calibrate the systematic errors of a mobile robot equipped with a differential drive system. On the contrary, non-systematic errors may appear unexpectedly because of unpredictable features in the environment causing large position errors. Non-systematic errors are not observable directly from odometry measurements making them almost impossible to compensate without using others measures in the localization process.

It is important to notice that, depending on the environment where the robot moves, one of this sources of errors will be predominant. On most smooth indoor surfaces systematic errors contribute much more to odometry errors than non-systematic errors. However, on rough surfaces with significant irregularities, non-systematic errors are dominant.

2.7 Exteroceptive Sensors

Exteroceptive sensors acquire information about the environment. This information is also called *allothetic* and basically allows a connection between the robot and its environment. The exteroceptive sensors determine the measurements of objects relative to a robot's reference frame and are classified according to their functionality into three main types:

Sources of error	
Systematic	- unequal wheel diameter - misalignment of the wheels - finite encoder resolution - finite encoder sampling rate
Non-Systematic	- travel over uneven floors - travel over unexpected objects on the floor - wheel-slippage due to: - slippery floors - over-acceleration - fast turning (skidding) - external forces (external bodies) - internal forces (castor wheels) - non wheel contact with the floor

Table 2.1: Sources of errors

- Contact sensors: which are typically mechanical switches that send a signal when physical contact is made.
- Range sensors: which are used to measure the distance to nearby objects, to avoid obstacles or to recover the scale factor for monocular vision.
- Vision sensors: which are used to extract features of the environment or to measure color and luminosity.

The exteroceptive sensors that we are interested in this thesis are range sensors and vision sensors because of their complementary information. Thanks to this sensors, the robot can choose perceptions that could be used as reference points. Unlike proprioceptive sensors, these reference points are independent of the movement of the robot, thus, do not generate cumulative errors and make them usable as long-term references. In particularly, we are going to deal with laser range finders and omnidirectional cameras.

2.7.1 Laser Range Finder

The laser range finder is an active sensor, i.e., it emits energy into the environment and measures the properties of the environment based on the response. According to Dudek and Jenkin (2000), based on the methodology used to measure the distance traveled by the laser beam, laser range finders can be of two types:

- Triangulation: This technique is called triangulation because the emitted laser and the reflected laser light form a triangle (see figure 2.11). Basically, the laser beam projects onto the measurement object and the reflected light is collected by a receiving device. The distance to the object can be calculated using geometric relationships between the outgoing beam, the incoming ray and the position of

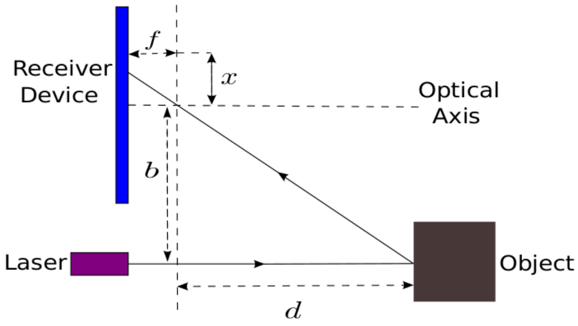


Figure 2.11: Triangulation principle.

the receiver as:

$$d = f\frac{b}{x} \tag{2.7}$$

where b is the distance between the emitter and the optical axis of the receiver, f the focal length and x the position where the reflected light hits the receiver.

This configuration restricts the distance measurement capability to one single point. In more sophisticated triangulation sensors a laser stripe, instead of a single laser dot, is swept across the object to provide 3D shape information of the object.

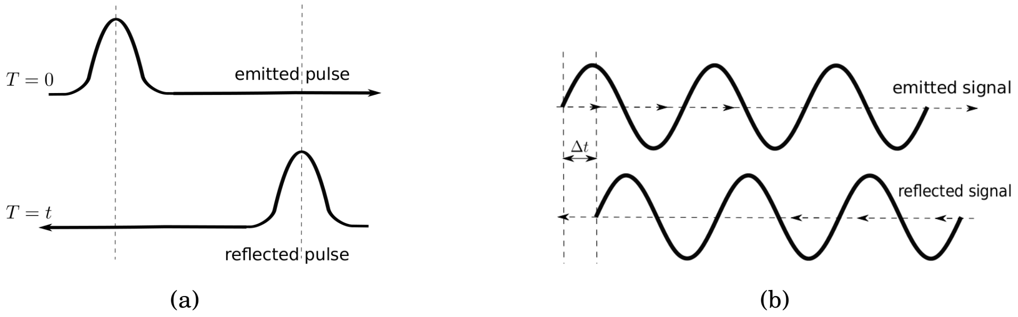
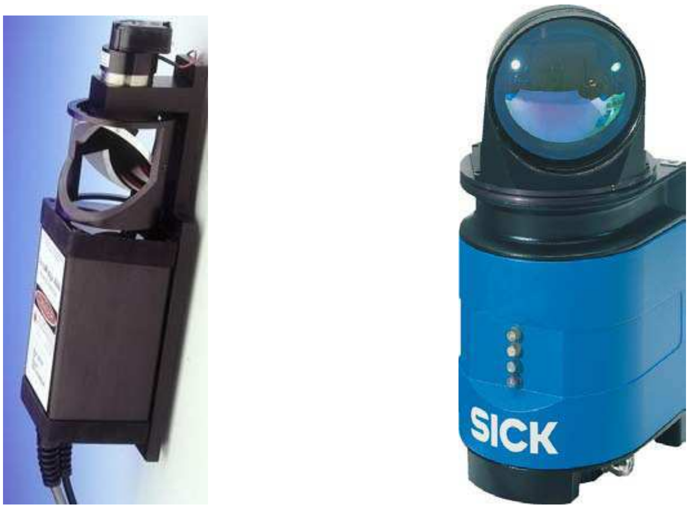


Figure 2.12: Time of flight methods. 2.12(a) Impulse time of flight. 2.12(b) Phase difference.

- Time-of-flight: This technique uses the time that the laser beam needs to reach the target and return back to measure the distance to the object. For calculating this time different methods can be used. The most common principles are the *impulse time of flight method* and the *phase difference method*. In the impulse time of flight method (see figure 2.12(a)), the elapsed time t is directly measured from the emission of a short impulse until receiving its reflection. Since the speed of light c is known, the object distance can be calculated by:

$$d = \frac{tc}{2} \tag{2.8}$$


(a)

(b)

Figure 2.13: Laser range finders mounted on Anis and Hannibal respectively. 2.13(a) Accurange 4000. 2.13(b) Sick LD-LRS1000.

Clearly, the accuracy depends on how precisely the time t is measured. In the second one, the phase based method (see figure 2.12(b)), the time t is calculated by measuring the phase difference between an emitted modulation signal and its reflection. Phase measurement is limited in accuracy by the frequency of modulation and the ability to resolve the phase difference between the signals.

Remark: *The laser only detects the distance of one point in its direction of view. Thus, either rotating the laser itself or by using a rotation mirror system, the view direction of the laser can be changed.* ■

The laser with which Anis is equipped (*AccuRange 4000* figure 2.13(a)), is composed of a laser telemeter with a rotating mirror that allows measurements of points on $360°$, except for an occlusion cone of approximately $30°$ caused by the assembly of the mirror. The telemeter computes distances using time of flight measurement principles, combining an intermediate technology between frequency modulation and amplitude modulation. A more detailed description of the system is explained in Victorino (2002). The range finder reaches a maximal frequency of 50Hz and is capable of acquiring 2000 data points in 40ms, which is more than enough for real time applications.

Hannibal is equipped with a Sick LD-LRS1000 laser (figure 2.13(b)), capable of collecting full $360°$ data. The distance to an object is measured using the impulse time of flight method. The laser head can revolve with a variable frequency ranging from 5Hz to 10Hz and the angular resolution can be adjusted up to $1.5°$ at multiples

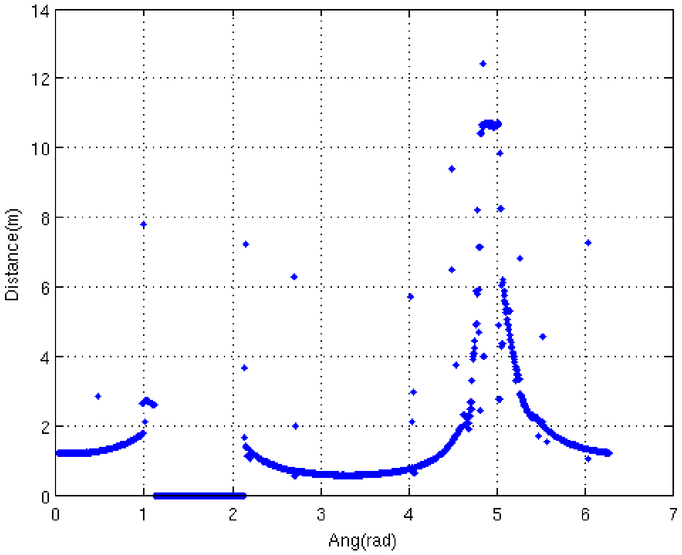


Figure 2.14: Laser scan in polar coordinates.

of $0.125°$. The laser maximum range is approximately 250m under ideal conditions (e.g. light surfaces). To perform a $360°$ scan with a resolution of $0.25°$, it is necessary to reduce the frequency of the rotor to 5Hz, thus obtaining 1,400 data points per scan.

It is important to recall that the scanning range of a laser depends on the reflectance of the object and the transmission of the scanner. The better a surface reflects the beam, the greater the scanning range is. The light reflected will vary depending on the nature and texture of the reflecting surface as well as the angle at which the light hits the surface. To mention some examples, light surfaces reflect the laser beam better than dark surfaces. In rough surfaces, part of the energy is lost due to shading. If the laser beam is incident perpendicularly to the surface the beam is correctly reflected if not, scanning range loss is caused. Even though laser range finders are know for their accuracy, they are unable to measure on transparent objects such as glass. This is a limitation in indoor environments applications, where the robot may pass nearby a window generating bad measurements.

The lasers measure their surroundings in two-dimensional polar coordinates. If a measuring beam is incident on an object, the position is determined as a distance and direction. The signal delivered from the laser range finder during a scan of the environment have the following form:

$$S = ((r, \phi_0), ..., (r, \phi_i), ..., (r, (\phi_{2\pi})) \tag{2.9}$$

where $r_i = r(\phi_i)$ is the distance measured between the origin of the laser coordinate system and the nearest object in the angular direction ϕ_i.

A representation of the output signal obtained by the laser after a horizontal scanning of the environment is shown in figure 2.14. It is clear that random noise sources are present including the timer counting the time-of-flight, the acquisition speed or the beam sweeping frequency. For robotics applications, it is common to pre-process the data, e.g. applying filtering algorithms, in order to get rid of some of the outliers.

In this thesis, laser range finders are used as a source of information to correct odometry errors, for cartography purposes, and to recover the third dimension of a monocular vision sensor. Each problem will be discussed in the following chapters.

2.7.2 Vision Sensors

Vision sensors (i.e. cameras) are a rich source of perceptual information about the environment of the robot. Alas, they require hard processing; e.g. extraction, characterization, and interpretation of data from the captured images in order to identify or describe objects in the given environment.

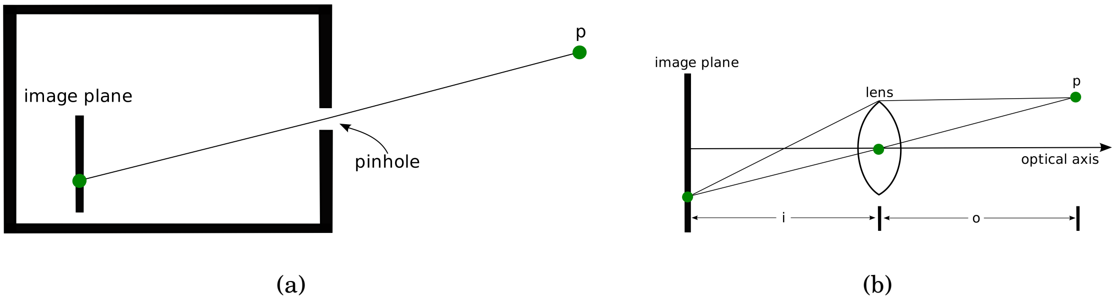


Figure 2.15: Basic Models. 2.15(a) Pinhole camera model, 2.15(b) Thin lens model.

Basically, cameras are used to map 3D (world) points onto a 2D surface (image plane). The pinhole camera was the first camera model. In a *pinhole camera model* the light from a point travels along a single straight path through a pinhole onto the image plane (see figure 2.15(a)). The object is imaged upside-down on the image plane. More detailed information about the pinhole camera model can be found in Faugeras (1993).

The pinhole camera have however, some limitations. If the aperture is to big, the image is blurry; if it is too small, it requires long exposure or high intensity.

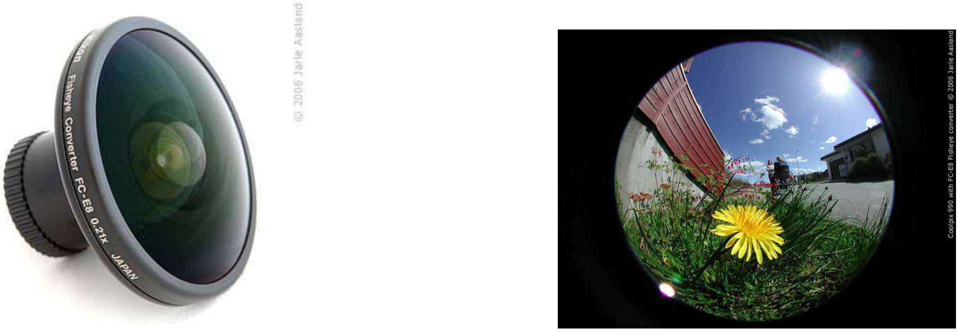
Nowadays, cameras use lens to focus light onto the image plane. Lens models can be quite complex, been perhaps the *thin lens model* the simplest one. In the thin lens model, rays of light emitted from a point travel along paths through the lens (see figure 2.15(b)), converging at a single point behind the lens. As a result the aperture can be bigger. In fact, a pinhole camera is an idealization of the thin lens as aperture shrinks to zero. A common Gaussian form of the lens equation is:

$$\frac{1}{i} + \frac{1}{o} = \frac{1}{f}$$

where i is the image distance, o is the object distance and f the focal length.

For our experiments we use a CCD camera, which means that the image plane contains a charge-coupled device array for the image formation. CCDs are widely used in professional, medical, and scientific applications where high-quality image data is required. The images obtained with this camera are *grey level* images.

Standard cameras have limited field of view –typically between 50° and 60°– which makes them useless for some applications in computer vision; e.g. place recognition or motion estimation can be easily affected by occlusion. Panoramic cameras on the other hand, provide wide field of view in a single image and can solve some of these issues. The wide-angle field of view gave more discriminate results and more robustness to changes in the environment. This is why they are becoming more and more popular in the robotics community.


Fisheye Converter FC-E8 0.21x JAPAN
© 2006 Jarle Aasland
Coolpix 990 with FC-E8 Fisheye converter © 2006 Jarle Aasland

(a) (b)

Figure 2.16: Fish-eye lens. 2.16(a) Fish-eye converter Nikon FC-E8. 2.16(b) Image Fish-eye [Source: http://www.nikonweb.com/fisheye/]

Wide Angle views

There are three main techniques for increasing the field of view of a camera. The first technique is based on the use of lenses to widen the field of view of a conventional camera. However, lenses are mostly bulky, complex and expensive to design. An example of this kind of system are the fish-eye lenses (see figure 2.16), which can acquire up to 180° images.

The second technique –called *blending* or *stitching*– is based on the generation of a panorama from a series of images of one or more conventional cameras. It is also possible to reconstruct the panorama using a rotating camera around a given axis, which is perpendicular to the optical axis. The main advantage of this technique is that it is possible to obtain panoramic images with high resolution. Nevertheless, the acquisition and the data association is computationally expensive and rarely real-time.

The third technique employ lenses and convex mirrors. This systems are called *catadioptric*, from *dioptric* relating to the refraction of light by lenses and *catoptric* relating to reflection of light using mirrors. Catadioptric systems are usually used to provide a far wider field of view (360°) than using lenses or mirrors alone. Therefore, in this work, a catadioptric system is adopted so as to obtain an omnidirectional image. The mirror is mounted on top of the camera lens (figure 2.17), thus providing an omnidirectional view of the robot's surroundings.

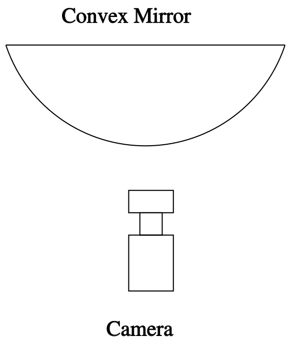


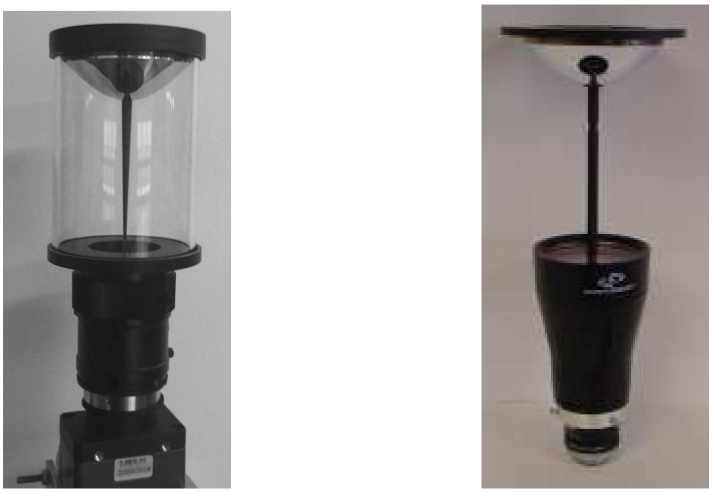
Figure 2.17: Construction of a Catadioptric System

There exist several types of mirrors designed to build omni-images: spherical mirrors, ellipsoidal mirrors, hyperbolical mirrors and parabolic mirrors are just some examples (see chapter 4 for catadioptric projection models). In this thesis, two different mirrors were used to construct two different catadioptric systems. Firstly, a progressive-scan CCD camera (Marlin F-131B) equipped with a telecentric lens and a parabolic mirror S80 from Remote Reality (see figure 2.18(b)). The second system uses the same CCD camera with a hyperbolic mirror HM-N15 from Accowle (Seiwapro) with a black needle at the apex of the mirror to avoid internal reflections of the glass cylinder (see figure 2.18(a)).

2.8 Multi-sensor Perception

Understanding the environment from sensor readings is a fundamental task in mobile robots. It is a well known fact that proprioceptive sensors and exteroceptive sensors provide different and complementary information about the environment, which is why nowadays, mobile robots are equipped with several sensor systems to avoid the limitations when only one sensor is used to reconstruct the environment.

Information from different sensors measuring the same feature, can be fused to obtain a more reliable estimate, reducing the final uncertainty on the measurement. Sensor fusion can be done at different levels: loose integration or tight integration. The term *integration*, can be defined as the fusion of two separate entities, resulting in a new entity. In loose integration –also called *loose coupling*– the state estimations provided by each independent sensor are fused. On the other hand, tight integration –also called *tight coupling*– consist in directly fuse the outputs (raw data) of each sensor. Loose and tight integration have been widely studied for many years mostly



(a)

(b)

Figure 2.18: Convex Mirrors 2.18(a) Hyperbolic Mirror. 2.18(b) Parabolic Mirror.

using Inertial Navigation Systems (INS) and Global Navigation Satellite Systems (GNSS) for efficient autonomous navigation purposes. Greenspan (1996) in his work on INS/GPS describes the loose and tight integration architectures.

More recently, researchers are using different sensors as Li et al. (2006), who perform a tight integration of a Global Positioning System (GPS), a Pseudolite (PL) and an INS. Soloviev (2008) has developed a multi-sensor tight integrated solution that combines the complementary features of the GPS, laser scanner feature-based navigation, and INS for urban scenarios. The work of Mourikis et al. (2009) called VISINAV describes a tight integration between the Absolute Visual Localization (AVL) method based on persistent features, and a Visual Odometry (VO) method using opportunistic features with an Inertial Measurement Unit (IMU) for spacecraft landing. Brevi et al. (2009) analyze both: loose and tight information coupling of a laser range finder and a WIFI signal to coordinate a team of mobile robots. Clearly, the choose of the sensors will depend on application and/or task to be performed by the robot.

Using raw data as it comes out from the sensors have many advantages, that is why we chose a tight integration to fuse the data from a laser range finder and an omnidirectional camera.

2.9 Conclusion

We have presented throughout the chapter the general framework of the SLAM problem. We have shown the solutions provided in the literature to solve it, as well as the importance of choosing a correct methodology depending on the information provided by the sensors and the application or task to be performed by the mobile robot. It was also explained the choice and the advantages of using a tight integration to fuse the raw data from a laser range finder and an omnidirectional camera in order to solve the SLAM problem. Our method will be explain in more detail in the next chapters.

"It doesn't matter how beautiful your theory is, it doesn't matter how smart you are. If it doesn't agree with experiment, it's wrong."

Richard Feynman

3 2D Laser-Based SLAM

Overview

In mobile robots, the primary objective is to make the robot able to act in an autonomous way in an unknown environment thanks to the sensors with which it is equipped. Using laser scan matching is one possible way to help solving this problem.

The purpose of scan matching when used for robot localization is to find the relative distance and rotation between a reference position and the current position of the robot by comparing one scan taken at the reference position and one scan taken at the robot current position. It is assumed that the current position is known approximately from, e.g. dead reckoning (odometry), which limits the search space of the scan matching algorithm. The scan matching algorithm then translates and rotates the actual scan to make the best overlap of the reference scan. The translation and rotation is, if the match is done correctly, the relative distance and rotation between the reference position and the current position. The relative distance and rotation are then used to update the position of the robot. One important step before the match is done, is the filtering of the scans, which smooths the scan points and remove outliers to avoid mismatches.

The aim of this chapter is to describe the generalities of the Laser Scan Matching algorithm, as well as the different approaches that there exist in the robotics community. Then we will focus on the Polar Scan Matching (PSM) algorithm and the generalizations that were made in order to improve it and make it robust enough to deal with lasers with arbitrary angular resolution and bearing range and in environments that does not contain predefined structures or features.

Keywords: Laser scan matching, pose estimation, ICP, PSM, 2D SLAM.

Organization of this chapter:

This chapter is organized as follows:

Section 3.1 introduces the Laser Scan Matching algorithm as a technique to solve the SLAM problem and gives a brief state of the art on the subject. It describe the most common algorithms used for scan matching, namely the ICP and its variants.

Section 3.2 describes a method for 2D laser scan matching called "Polar Scan Matching". Each one of the phases that constitute the method are described in detail. A generalization of the method, called Enhanced Polar Scan Matching is proposed.

Section 3.3 explains how to build local maps using the Enhanced Polar Scan Matching implementation described in section 3.2. It is also described how this local maps are used in the localization and mapping process, while a 2D global map is built thanks to the SLAM approach, from which it is possible to recover the pose of the robot at each instant.

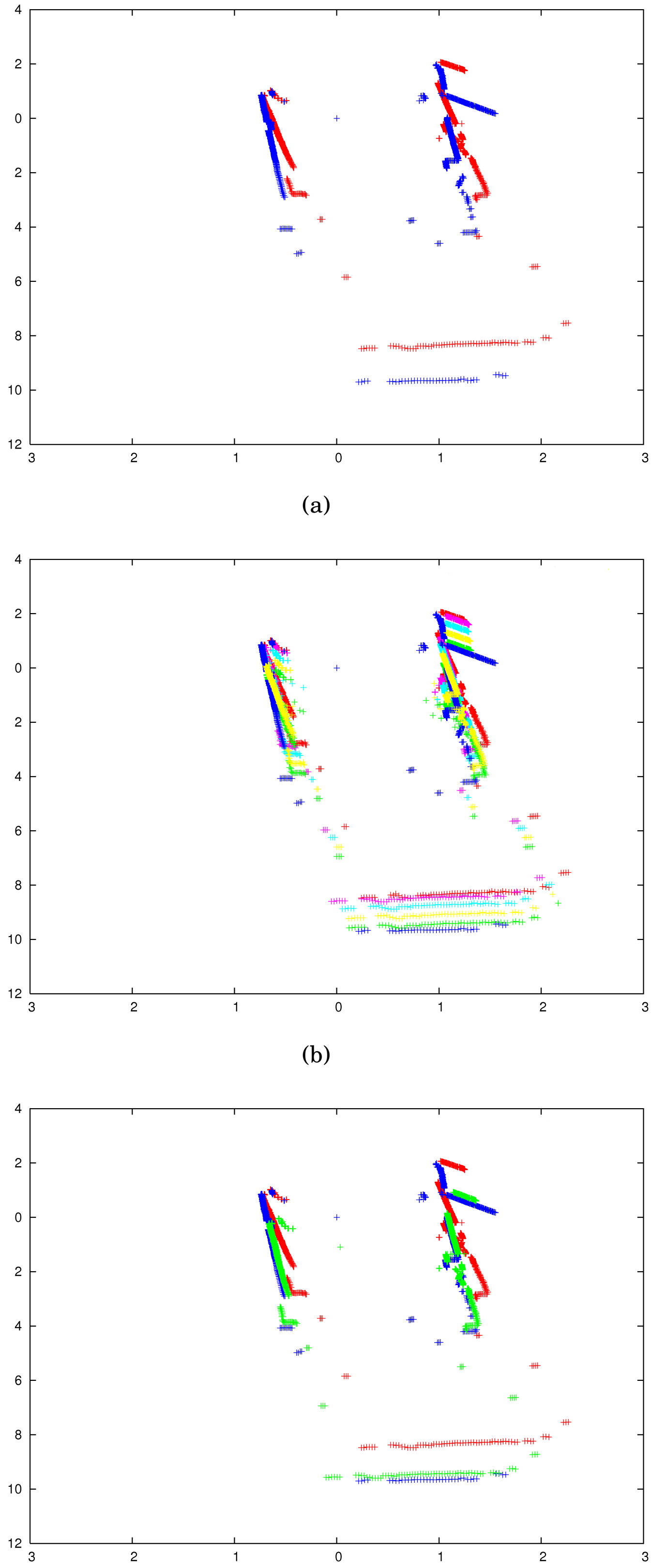
Section 3.4 show the results of our implementation of the EPSM in synthetic and real data, and depicted the obtained 2D global maps in two different indoor office-like environments.

Section 3.5 concludes the chapter.

3.1 Laser Scan Matching

Laser Scan Matching has been widely used for robot localization and mapping because of its simplicity. The goal of the laser scan matching algorithm is to find the position and orientation (i.e., the pose) of a *current* scan with respect to a *reference* scan by adjusting the pose of the current scan until the best overlap with the reference scan is achieved (see figure 3.1).

Several methods can be found in the literature for 2D and 3D scan matching. These methods are often categorized based on their association rule such as feature to feature or point to point matching. In the feature-based approach [Gutmann et al. (2000); Ramos et al. (2007)], features such as line segments and corners are extracted from laser scans and then matched against each other. More recently, Nieto et al. (2008), propose a scan matching approach that allows the use of arbitrary shapes. However, such approaches requires the identification of appropriate features in the



(a)

(b)

(c)

Figure 3.1: Laser Scan Matching algorithm. 3.1(a) Reference (blue) and current scan (red). 3.1(b) Scan matching algorithm iterations. 3.1(c) Current scan aligned (green) to the reference scan.

environment. On the other hand, point to point matching does not require the environment to be structured or contain any predefined features, which is why we will focus our attention in this approach.

The Iterative Closest Point [ICP by Besl and McKay (1992)] algorithm is perhaps the most widely used point to point scan matching method that works with range sensors. ICP uses a nearest neighbor association rule to match points, and least squares optimization to compute the best transformation between two scans. Two enhanced methods based on ICP were proposed by Lu and Milios (1997b): the Iterative Matching Range Point (IMRP) and the Iterative Dual Correspondence (IDC) method. Although ICP and its extensions [Rusinkiewicz and Levoy (2001)] are fast and in general produce good results, they are only guaranteed to converge towards a local minimum and may not always find the correct transformation. Furthermore, these algorithms suffer from computational complexity problems when dealing with large-scale environments because the point to point association rule they use, result in a $O(n \log(n))$ complexity in the best case (where n is the number of points in a scan).

To overpass these constraints, Diosi and Kleeman (2005) proposed the Polar Scan Matching method which avoids searching for point associations by simply matching points with the same bearing. Another approach using normal distributions has been proposed by Biber and Straßer (2003) to represent laser scans in order to avoid correspondences between primitives. Weiß and Puttkamer (1995) and Bosse and Zlot (2008) used angular histograms to recover the rotation between two poses. Then both histograms, which were calculated after finding the most common direction were used to recover the translation.

The next section will focus on the Polar Scan Matching approach proposed by Diosi and Kleeman (2005), while at the same time, our extensions to generalize their algorithm –so it can deal with arbitrary angular resolution and bearing range– will be described. In the first instance, our *enhanced* PSM implementation will be used to build 2D local maps of the environment. These local maps will be used both, in the localization and mapping process. Later, these maps will be used as a 2D laser-based SLAM to reconstruct the 2D global map from which it is possible to recover the pose of the robot at each instant.

3.2 Polar Scan Matching

Polar Scan Matching (PSM) is a point to point laser scan matching method that exploits the natural representation of laser scans in a polar coordinate system to cut the complexity of the matching process. As other scan matching approaches, like the Iterative Closest Point (ICP) method, the PSM method finds the pose of a laser scan with respect to a reference scan by performing a gradient descent search for the transformation that minimizes the square error between corresponding points.

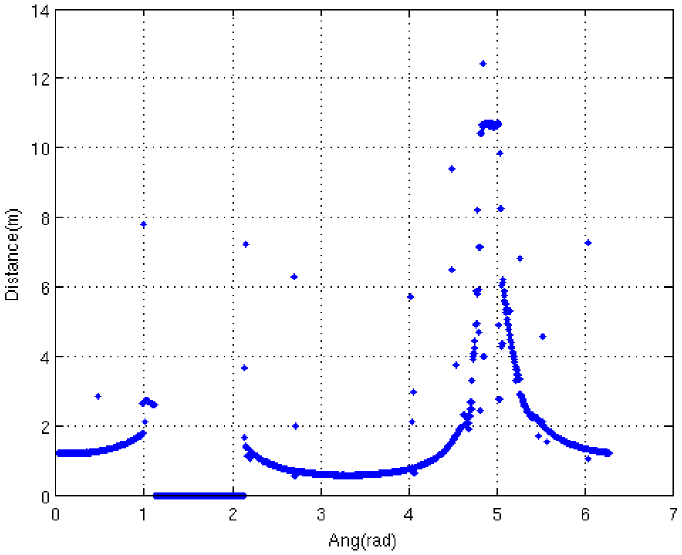


Figure 3.2: Raw polar laser scan.

In contrast to other matching methods, PSM avoids an expensive search for corresponding points by matching points with the same bearing. The method assumes the reference and current scans are given as sequences of range and bearing measurements of the form $\{r_{ri}, \phi_{ri}\}_{i=1}^{n}$ and $\{r_{ci}, \phi_{ci}\}_{i=1}^{n}$, respectively, and requires an initial estimate (x_c, y_c, θ_c) for the pose (position and orientation) of the current scan in the reference scan coordinate frame.

The method may be best described by detailing each of its phases:

- Pre-processing
- Scan projection
- Translation and orientation estimation

Basically after pre-processing the scans, scan projection followed by a translation estimation or orientation estimation are iterated.

3.2.1 Scan Pre-processing

Before the matching and in order to have a good couple of laser scans to be matched a pre-processing of the scans is required. Figure 3.2 show the raw data provided by the laser. It can be seen that laser raw data have a lot of noise and bad measurements (outliers) due to several reasons, for example, a bad reflection of the laser beam on metallic or dull surfaces. The pre-processing step helps to remove outliers to increase the accuracy and robustness of scan matching.

The pre-processing is applied to the data in three basic steps:

1. First we have the *filtering step*, which help us to remove some undesirable measurements from the laser scan. It was shown in Gutmann (2000) that using the median filter it was possible to replace outliers with suitable measurements. The results obtained were quite satisfactory. However, in order to improve the filtering step, instead of the median filter, the algorithm for rejection of local artifacts proposed by Victorino (2002) was applied to remove the outliers from the laser range measurements. With this algorithm, a sliding window of three points is applied to the laser distance measurements. If the relative distance between points is greater than a predefined threshold, either we are in the presence of an outlier or a new set of continuous points. It produces n point sets with at least three close points. This algorithm attains much better results than the median filter.

2. Secondly, the *tagging step* will help us in the segmentation step. In this step all the points further than a threshold (MAX_RANGE) are tagged. This points will not be used in the scan matching as they will be removed, like this we avoid the problem of interpolating neighboring points of two different objects which can be source of an error. The choice of this threshold is based on the maximum range and angular resolution of the sensor.

3. Finally, after tagging long range measurements a *segmentation* algorithm is applied. Then once having defined the segments, the interpolation between two different objects can be avoided. The segmentation is done according to two simple criteria:

   - If two consecutive range readings points are no further than a threshold, they belong to the same segment.
   - If three consecutive range readings points lie approximately on the same polar line, they are assigned to the same segment.

   Segments consisting of a single point are discarded (most mixed pixels). To aid the segmentation process, the maximum range is limited so that two consecutive readings belonging to the same segment cannot be too far apart. Also, one more step was added to the segmentation algorithm, which we called the *segment filter* (see algorithm 1). This step will remove all the segments smaller than five points, so, only ”long” segments are kept.

After the pre-process step, we have now a new filtered laser scan to use in the scan matching algorithm.

3.2.2 Scan Projection

There is an intermedium step before applying the scan matching algorithm called *scan projection*. The aim is to find out how the current scan would look like if it

Algorithm 1 Segment Filter

```
for $each\_segment$ do
  if $length(segment) < MIN\_SEGMENT$ then
    Remove(segment)
  end if
end for
```

where taken from the reference position. This will help to calculate the error in the estimation of the pose of the current scan.

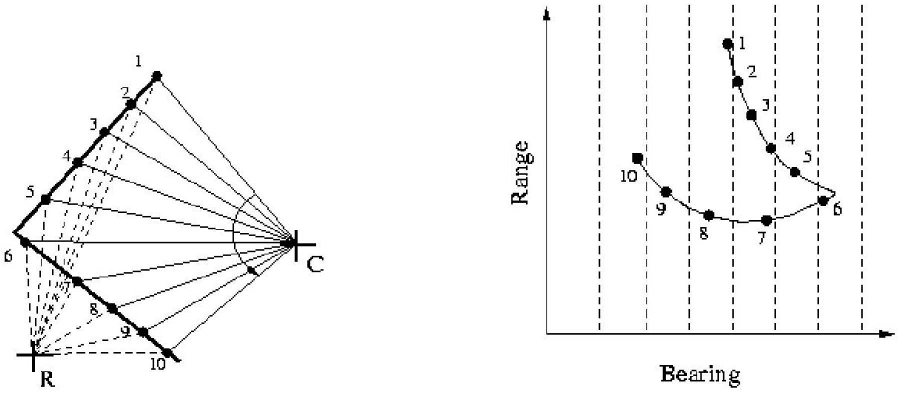
The projection of the current scan into the reference scan coordinate frame is a sequence of measurements $(r'_{ci}, \phi'_{ci})_{i=1}^{n}$ computed as follows:

$$r'_{ci} = \sqrt{(r_{ci}\cos(\theta_c + \phi_{ci}) + x_c)^2 + (r_{ci}\sin(\theta_c + \phi_{ci}) + y_c)^2} \tag{3.1}$$

$$\phi'_{ci} = \text{atan2}(r_{ci}\sin(\theta_c + \phi_{ci}) + y_c, r_{ci}\cos(\theta_c + \phi_{ci}) + x_c) \tag{3.2}$$

where atan2 is the four quadrant version of arctan, and (x_c, y_c, θ_c) is the pose (position and orientation) of the current scan in the reference scan coordinate frame.

The association rule for this algorithm is to match bearings of points. However, the bearings of the above sequence do not necessarily coincide with bearings where the laser would have sampled a reading. A range measurement r''_{ci} is computed for each sample bearing by linear interpolation among points belonging to a same segment. Points that would have been occluded are not taken into account, only the smallest range measurement for a bearing is kept.



(a)

(b)

Figure 3.3: Scan Projection. 3.3(a) Projection of measured points taken at C to location R. 3.3(b) Points projected to R shown in polar coordinates. [Source: Diosi and Kleeman (2005)]

As an illustration of the scan projection process, figure 3.3 show the current scan taken at location C and the reference scan taken at position R. The range and bearings of the points from point R are calculated with equations 3.1 and 3.2 respectively. Dashed vertical lines in figure 3.3(b), represent sampling bearings ϕ_{ri} of the laser at position R in figure 3.3(a).

3.2.3 Translation and Orientation Estimation

The method alternates between translation and orientation estimation. After making a correction to the pose estimate, the projection phase is repeated with the corrected estimate. The process stops when the magnitude of the last position and orientation correction is smaller than a given threshold, hopefully indicating that a minimum has been reached. Translation is estimated using a standard weighted least squares method. A correction $(\Delta x_c, \Delta y_c)$ to the position estimate is found by minimizing the weighted sum of the square range residuals $\sum_{i=1}^{n} w_i(r_{ri} - r''_{ci})^2$ while leaving orientation unchanged. The weights are computed as recommended by Dudek and Jenkin (2000),

$$w_i = \frac{c^2}{(r_{ri} - r''_{ci})^2 + c^2} \tag{3.3}$$

Orientation is estimated by computing the average range residual for 1° shifts of the current scan in a $\pm 20^\circ$ window. The new orientation estimate is found by fitting a parabola to the shift with the minimum average error and its left and right neighbors.

The implementation of the PSM method provided by Diosi is tailored to a laser with 1° angular resolution and 180° bearing range. These assumptions are used when transforming sample bearings from radians to indexes into arrays and back. We generalized Diosi's implementation to lift these assumptions. Our implementation is parametrized so that it can deal with lasers with arbitrary angular resolution and bearing range. In addition, instead of just returning the pose estimate at the moment the algorithm stops, our implementation keeps record of the estimate with the minimum error and returns it as a result.

3.3 Local and global maps with SLAM

As stated before, 2D local maps of the environment are built using the enhanced PSM implementation described in the previous section. Local maps will be used both, in the localization process and for mapping the environment. Finally, these maps will be used in SLAM to reconstruct a 2D global map from which it is possible to recover the pose of the robot at each instant.

Let T_L be the rigid transformation between the laser coordinate frame and the robot coordinate frame. The global coordinate frame is fixed to be the coordinate frame of the odometry data. Let (x, y, θ) be the current position of the laser scan coordinate frame. The affine transformation matrix from the laser coordinate frame to the global coordinate frame is given by the procedure shown in algorithm 2.

Algorithm 2 Affine transformation for a translation (x, y) and a counterclockwise rotation around the origin by an angle θ.

```
AFFINE-TRANSFORMATION$(x, y, \theta)$
    return $\begin{bmatrix} \cos\theta & -\sin\theta & 0 & x \\ \sin\theta & \cos\theta & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
```

The procedure shown in algorithm 3 is used to build the global map and reconstruct the path of the robot from a sequence of laser range scans with associated odometry data. We begin by taking the first scan in the sequence as the reference scan S_R. Initially, the map consists only of the points in the scan S_R represented in the global coordinate frame, but it will be incrementally enriched at each iteration of the loop. At every moment, a transformation matrix T_3, from the coordinate frame of the laser in the reference scan frame to the global coordinate frame is kept. At the beginning of each iteration the next scan in the sequence is taken to update the current scan S_C.

Then the odometry data is used to obtain an initial estimate for the pose of the laser in the current scan with respect to the reference scan coordinate frame. This estimate is feed to the PSM procedure described in the previous section, and get as a result a new estimate of the pose. Using this new estimate, the T_3 matrix is updated, the points in the current scan are transformed into the global coordinate frame, and added to the global map. The current scan becomes then the reference scan and the whole process is repeated again.

Because the short-term odometry of the robot when traveling on a flat surface is relatively accurate, in practice we do not need to use scan matching to compute the pose of the robot in every scan. Instead, we only use scan matching to get a better estimate of the pose of the robot when it has traveled a certain distance or rotated a certain angle, or when a certain lapse of time has passed since the last time scan matching was used.

Using the results obtained by the PSM algorithm, the odometry data of the whole sequence can be recomputed. It suffices to multiply after each iteration matrix T_3 by the transformation matrix T_L^{-1}, which gives the transformation matrix from the robot's (not the laser) coordinate frame of the current scan to the global frame. The pose (x, y, θ) can be readily extracted from this last matrix.

Algorithm 3 Pseudocode of the procedure used to incrementally build a global map from a sequence of laser range scans with odometry information.

```
GLOBAL-MAP($scan[N]$)
$S_R \leftarrow scan[1]$
$T_1 \leftarrow$ AFFINE-TRANSFORMATION($S_R.x, S_R.y, S_R.\theta$)
$T_3 \leftarrow T_1 \times T_L$
$Map \leftarrow$ APPLY-TRANSFORMATION($T_3, S_R$)
for $i \leftarrow 2$ to $N$
    $S_C \leftarrow scan[i]$
    $T_2 \leftarrow$ AFFINE-TRANSFORMATION($S_C.x, S_C.y, S_C.\theta$)
    $T \leftarrow T_L^{-1} \times T_1^{-1} \times T_2 \times T_L$
    $(x, y, \theta) \leftarrow (T_{(1,4)}, T_{(2,4)}, \mathrm{atan2}(T_{(2,1)}, T_{(2,2)}))$
    $(x, y, \theta) \leftarrow$ PSM($S_R, S_C, x, y, \theta$)
    $T_3' \leftarrow$ AFFINE-TRANSFORMATION($x, y, \theta$)
    $T_3 \leftarrow T_3 \times T_3'$
    $Map \leftarrow Map \cup$ APPLY-TRANSFORMATION($T_3, S_C$)
    $S_R \leftarrow S_C$
    $T_1 \leftarrow T_2$
return $Map$
```

3.4 Results of Implementation

The enhanced PSM algorithm was first tested with synthetic data of a simulated room which covers the whole room (360°). Figure 3.4 show in red the reference scan –in this case the simulated room– and in green the current scan prior to matching. Both scans are identical, but the pose and orientation of the current scan was altered.

Figure 3.5 shows the convergence sequence –iterations– of the EPSM algorithm. Finally, the result of the EPSM with synthetic data of the simulated room is shown in figure 3.6, where the current matched scan is in green and the reference scan is in red. It can be seen that the current scan converge accurately to the reference scan.

Later, our algorithm was tested with a pair of real scans taken after a small displacement of our robot. In this particular experiment, we chose a pair of scans relatively difficult to match, in order to evaluate the convergence criterion of the algorithm. The reference and current scan prior to matching are shown in figure 3.7. It can be seen from the convergence sequence in figure 3.8, that even though the scans are quite different, they have some similarities that the algorithm uses to correctly match the scans as shown in figure 3.9.

The first laser-based SLAM experiment used Anis robot in an indoor office-like environment (Borel building). The sequence was obtained by manually commanding the robot to explore the ground floor in a closed loop in a corridor. The original odometry is in red, and the estimated trajectory after scan matching is in green.

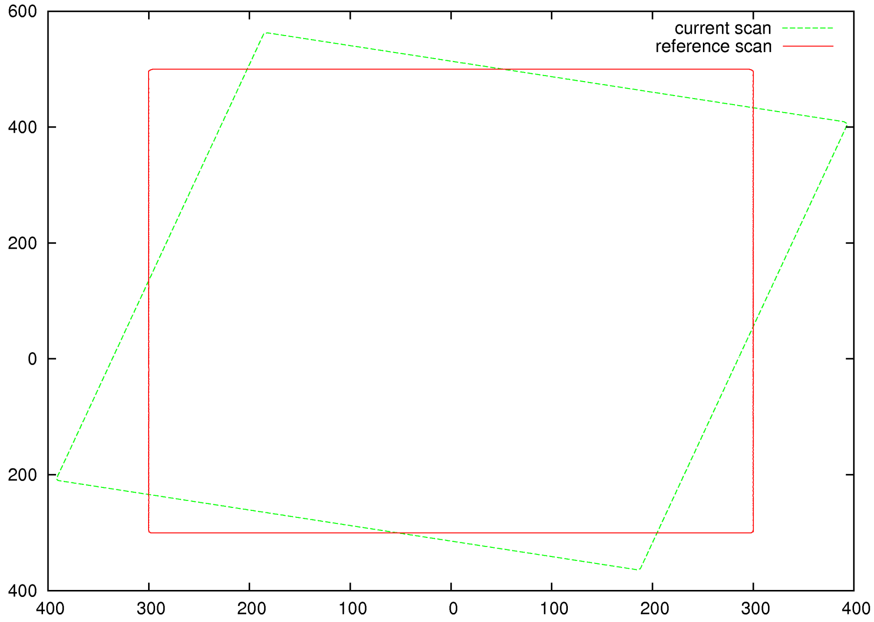


Figure 3.4: Synthetic data: current scan (green) and reference scan (red) prior to matching

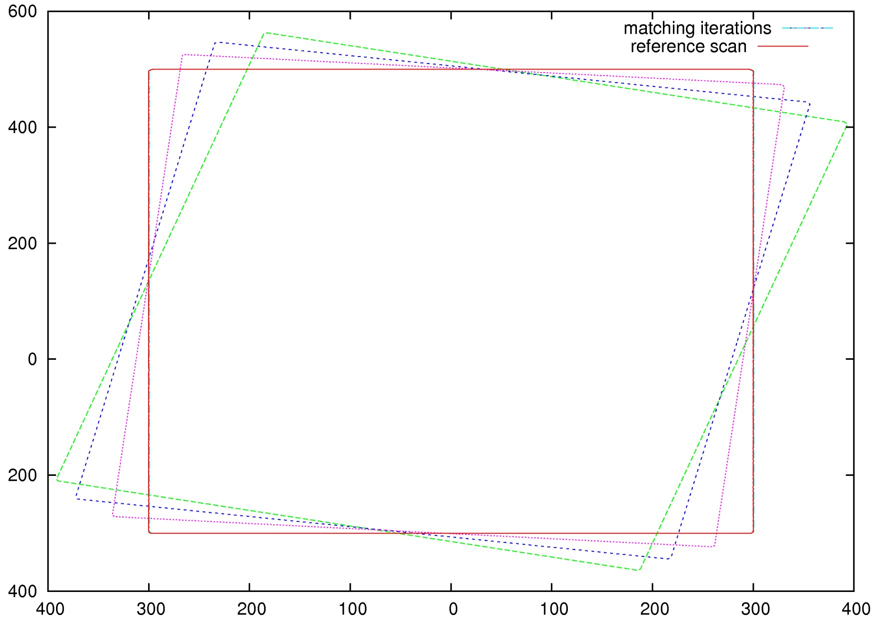


Figure 3.5: Convergence sequence of a simulated room

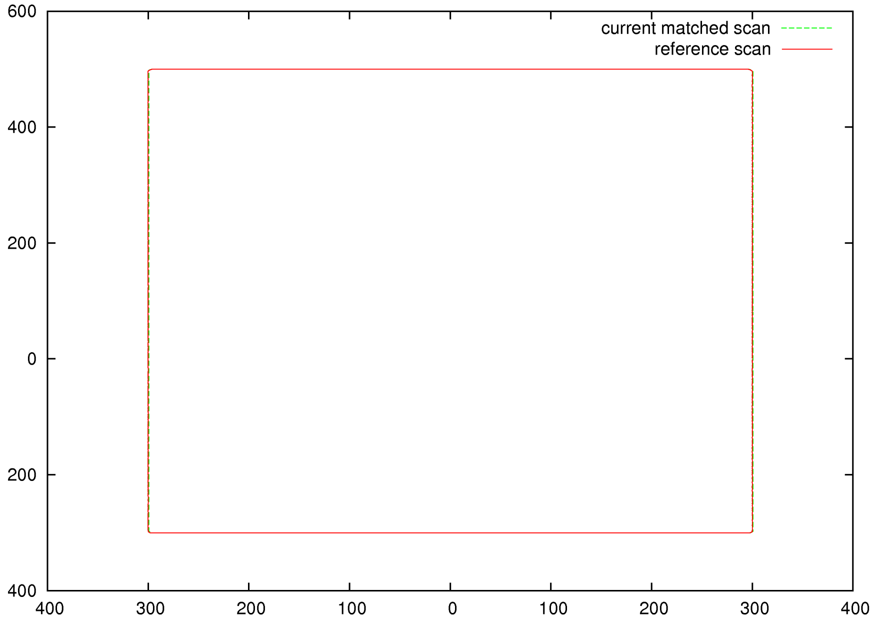


Figure 3.6: EPSM result: Matched current scan (green) and reference scan (red)

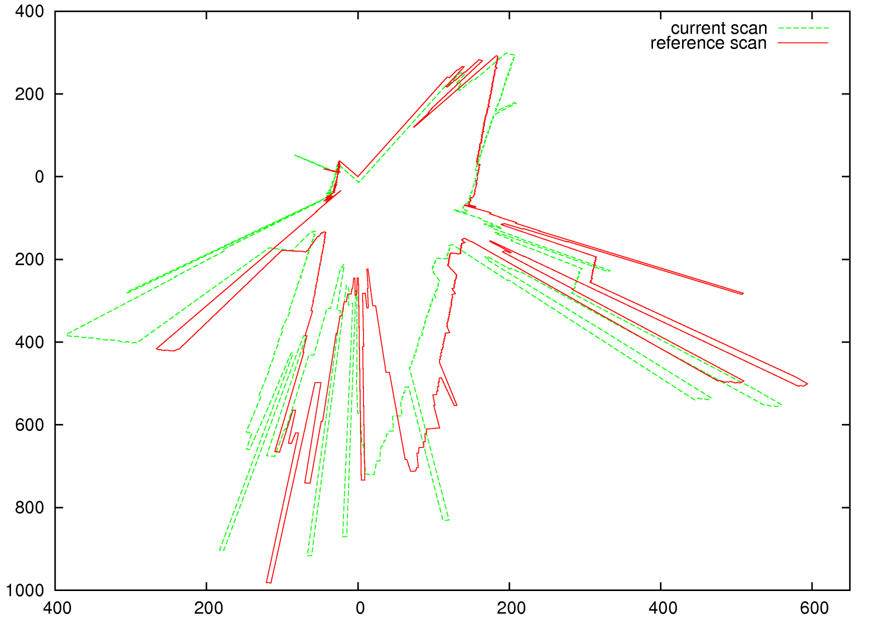


Figure 3.7: Real scans: current scan (green) and reference scan (red) prior to matching

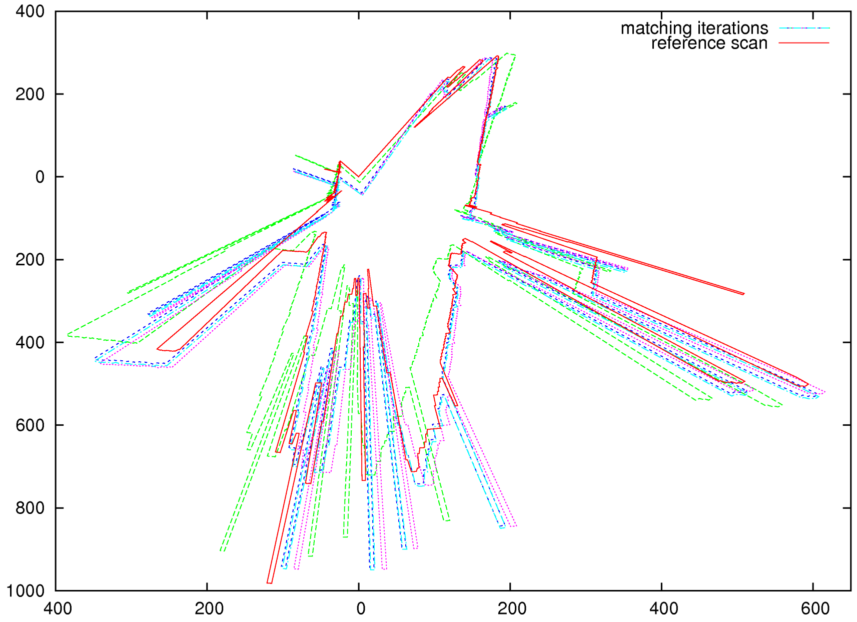


Figure 3.8: Convergence sequence of real data scans

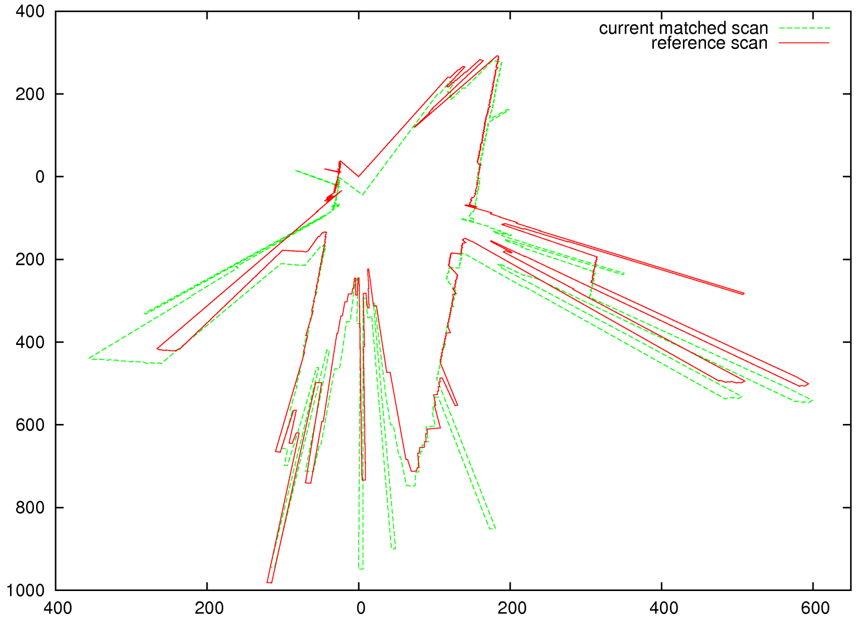


Figure 3.9: EPSM result: Matched current scan (green) and reference scan (red)

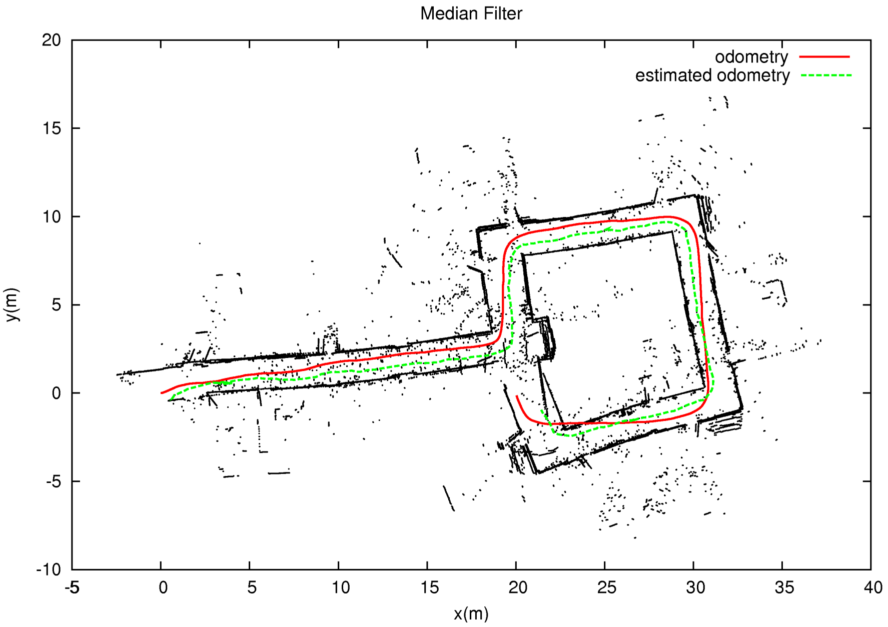


Figure 3.10: Scan Matching with median filter

In a first instance, the implementation used the median filter. The result is shown in figure 3.10. It is important to notice that the median filter is not powerful enough to eliminate many of the outliers, which causes wrong associations. As a result, the map is not straight because of the considerable drift at the end.

A second experiment using the algorithm for rejection of local artifacts instead of the median filter is shown in figure 3.11. Obviously, as expected, much more outliers were removed and a better result was obtained. However, there is still a small drift at the end of the sequence.

Finally, with the implementation of the proposed segment filter we obtained the map shown in figure 3.12. The result is a straight map without drifts at the end of the sequence. It is important to notice here, that even when we are not using any algorithm to close the loop, the EPSM achieves good convergence.

As we stated before in chapter 1, the robots are capable of taking measurements from different sensors, namely, odometry, laser scans and omnidirectional images. In Joly (2010), this same sequence was used to perform visual-based SLAM using a SAM algorithm. In order to compare and evaluate the validity of our results, we show on figure 3.13 the results obtained with the same sequence using the odometry and omnidirectional images.

The second experiment to evaluate our laser-based SLAM algorithm, uses our more recent robot Hannibal. As stated before, the sequence was obtained by manually commanding the robot to explore the ground floor. In this case, the environment is not a corridor, but an indoor environment of the robotic hall (kahn building). The robot performed always a sequence in a closed loop. Figure 3.14 shows the position of

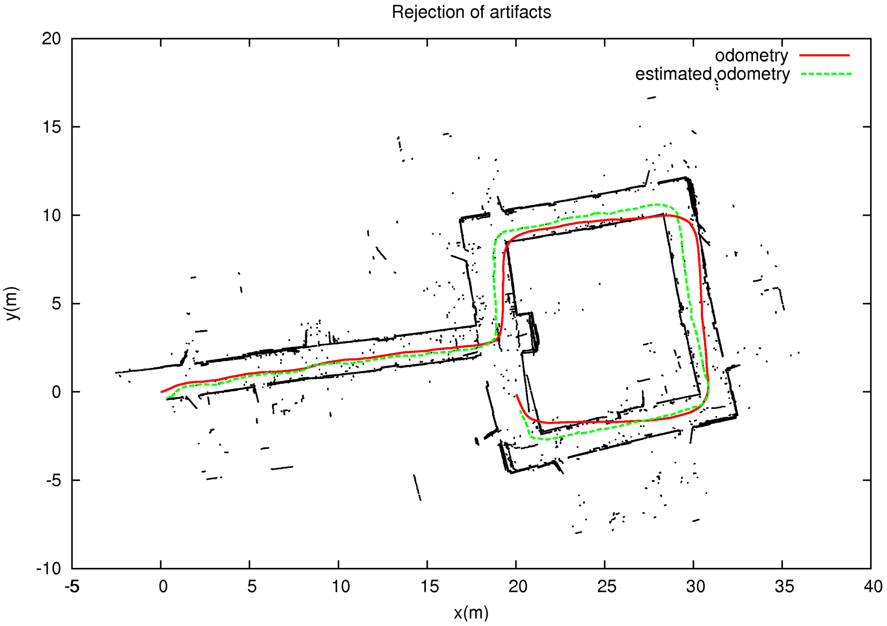


Figure 3.11: Scan Matching with rejection of local artifacts algorithm

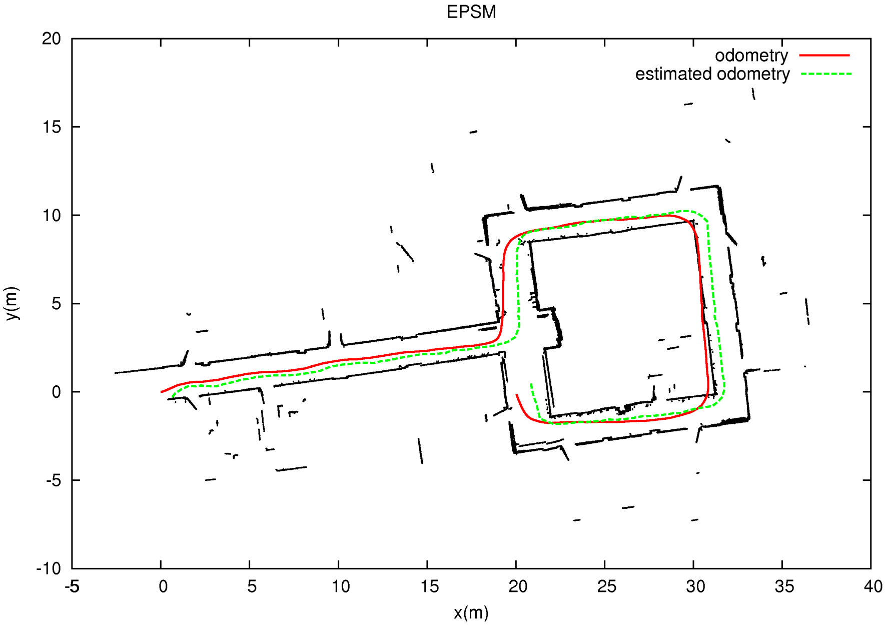


Figure 3.12: Global map obtained by EPSM-SLAM algorithm

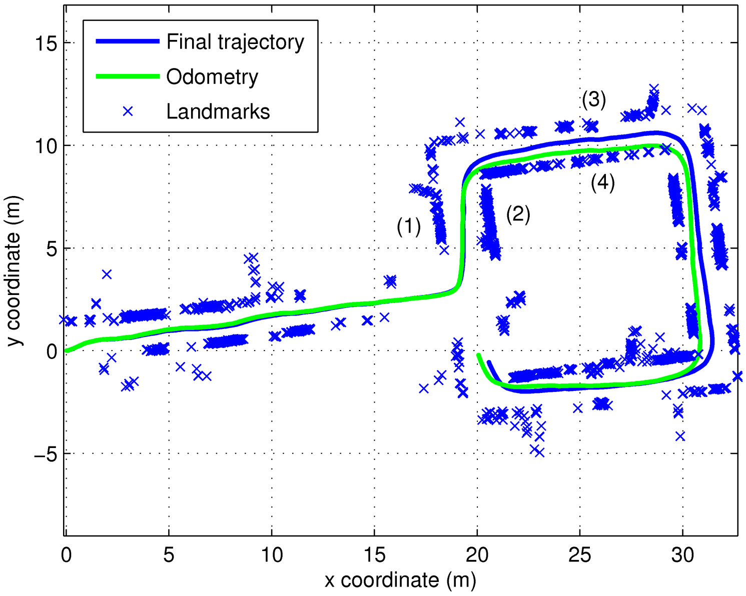


Figure 3.13: Global map of Borel building obtained using visual data [source: Joly (2010)]

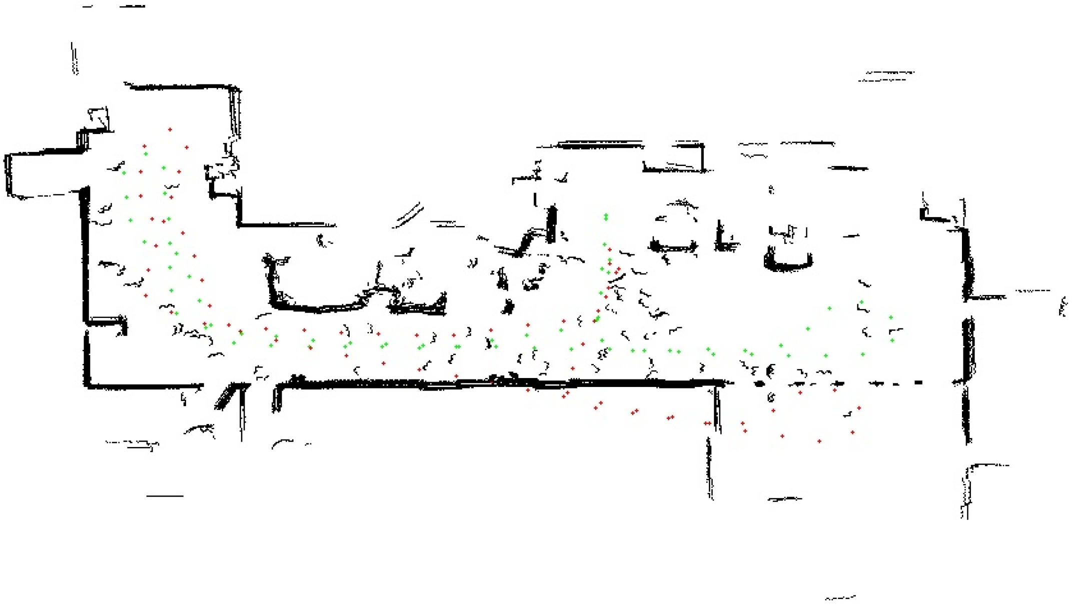


Figure 3.14: Global map obtained by SLAM together with the original and recomputed position of the robot at several key instants.

the robot at several instants in the sequence as given by the original odometry data (in red) and as computed by scan matching (in green) superimposed on the generated global map.

As before, although we did not perform closed-loop detection or corrections of any kind, the results are quite satisfactory. The recomputed odometry represents a big improvement over the original odometry that even drifts out of the building.

The parameters used in scan matching during the experiment are in table 3.1:

Half Window	2
Segment Hop	20cm
Max Range	10m
PSM Window	80^o
Min Segments	5
Max Iterations	50
Skip scans	30

Table 3.1: Parameters

3.5 Conclusion

We have proposed in this chapter a generalization of the Polar Scan Matching algorithm in order to deal with lasers with arbitrary angular resolution and bearing range. Our particular interest in this algorithm is due to its ability to work with lasers measurements in its original form, i.e., as the data is delivered by the laser rangefinder: the polar form. This will allow us later, to carry out a tightly coupled sensor fusion between the measurements obtained from the laser rangefinder and an omnidirectional camera, as will be shown in chapter 4.

In addition, our Enhanced PSM inherits the advantages of the original PSM such as $O(n)$ complexity pose estimation thanks to its matching bearing association rule. Furthermore, as any other point to point matching approach, it does not require the environment to be structured or contain any predefined features.

A map resulting from SLAM with EPSM has been compared to a map resulting from SLAM using a SAM algorithm and vision information of the same sequence, in order to compare and evaluate the validity of our results. This results encourage us more to use the raw information of both sensors in a composite scheme as will be detailed in the next chapter.

"Teachers open the door. You enter by yourself."

Chinese Proverb

4
Tightly-Coupled Sensors Fusion

Overview

The aim of this Chapter is twofold: firstly, it introduces the main aspects of omnidirectional vision. It will be described the peculiarities of the modeling including the use of spherical projection. Secondly, we will describe how to link images obtained by an omnidirectional camera with a laser range finder in order to build a composite laser/omnidirectional sensor that will enhance both, localization and map representation of the robot's environment.

Keywords: Vision sensors, omnidirectional cameras, unified projection model, tightly coupled fusion, floor segmentation.

Organization of this chapter:
This chapter is organized as follows:

Section 4.1 explores the world of omnidirectional vision through biology and human history. A brief state of the art on catadioptric cameras is given. The classification proposed in the literature for the different catadioptric systems is described.

Section 4.2 focuses on the advantages of central catadioptric sensors for robotics. The entire class of catadioptric systems is described in detail. Then, the Unified Projection model is presented in section 4.2.2.

Section 4.3 and 4.4 explains the parametrization and low-level feature extraction of "radial" lines in the omnidirectional image. The canny edge detector and the Randomized Hough Transform are described, and some results of the proposed procedure are given.

Section 4.5 describes the procedure developed to extract vertical lines from omnidirectional images (as stated in the previous section) and to estimate their 3D positions using information from the laser range finder. It will be shown how to build a 3D wired representation of the environment using a SLAM approach based on the detected lines.

Section 4.6 considers that the laser scan can be shifted along the vertical lines and could be used to predict where a virtual laser trace, corresponding to the floor, should project onto the omnidirectional image. A procedure to correct the segmentation of the floor with the laser trace is presented, which completed our laser/omnidirectional sensor that enhance both, localization and map representation of the robot's environment. Some results are discussed at the end of the section.

Section 4.7 gives the conclusion of the chapter

4.1 Omnidirectional vision

It is just natural for animals and humans to use vision to perform accurately navigation tasks. The visual sense gives us timely knowledge of our spatial surroundings, near and far, identifying all the objects in it to our consciousness. Without vision, to simply move from one place to another becomes complicated and dangerous, so it is equally natural to consider vision as a sensor for mobile robots. Visual sensing has many desirable features, including passivity, high resolution, and long range.

The sense of vision in humans relies on both eyes to transform information received as light into electrical pulses transmitted by neurons. How the brain processes this information to construct an internal representation of the environment and how we reason about our environment using this representation is the subject matter of the fields of perception and cognition in biological sciences. In robotics, the field of computer vision studies the task of constructing representations of the environment from visual data, while artificial intelligence investigates reasoning and planning based on these representations.

The area that encompasses the sight is called field of view or *field of vision* (FOV). In the animal kingdom, the field of view is adapted to the type of animal and the environment. Herbivores (rabbits, horses, cows, ...) have a large field of view (more than $300°$) but a small binocular field (around $50°$). Carnivores or primates in contrast usually have smaller field of views with bigger binocular regions. In robotics, to choose the FOV will depend on the application and the task to solve.

As explained before, by placing a perspective camera in front of a convex mirror, we can obtain $360°$ field of view of the environment. This system is called *catadioptric system*. It was probably Rees (1970), the first catadioptric sensor patented in the United States. He proposed a system combining a conventional camera with a hyperbolic mirror. Withal, it was not until 20 years after that researchers began to explore the advantages of omnidirectional vision in applications for robotics and computer vision.

One of the pioneers on catadioptric cameras was Yagi and Kawato (1990) who combined for the first time a conventional camera with a conic mirror (COPIS) for robotics applications. Also, Hong et al. (1991) made an omnidirectional vision sensor using a spherical mirror. Unfortunately, these types of systems did not satisfy the single viewpoint constraint. Later Yamazawa et al. (1993), combined a hyperbolic mirror with a conventional camera to generate omnidirectional images that satisfy the single viewpoint constraint. As explained by Yamazawa et al. (1993) and completed later with a complete analysis of the geometric properties by Baker and Nayar (1999), a single view point constraint (i.e. center of projection) is desirable for any catadioptric system in order to generate geometrically correct perspective images. From a theoretical point of view, it is easier to model and analyze systems that meet this condition. However, this constraint imposes special conditions when designing the sensor, such as a precise positioning of the components of the mirror and the camera.

Catadioptric systems can therefore, be classified on whether they have a single center of projection or not. The non-central catadioptric sensors have, by definition, more than one center of projection (several points of view) that forms a continuous region. This region is called a *caustic* (Swaminathan et al. (2006)). Two classic examples of caustic are show in figure 4.1

The presence of several centers of projection –as for conic or spherical mirrors– induces particular properties that are difficult to exploit, which is why the rest of this thesis will focus on central catadioptric sensors.

4.2 Central Catadioptric Models

Central catadioptric sensors are the most used systems in omnidirectional vision. They combine convex mirrors with conventional cameras to form the omnidirectional images preserving the single viewpoint property. This means that all light rays that enter the camera to form the image, passes through a single point (i.e. projection center) and the corresponding reflected rays (after a telecentric lens for parabolic mirrors) also pass through a single point (i.e. optical center).

The single viewpoint constraint is a desirable property because it allows the mapping of any part of the scene to a perspective plane without parallax, i.e., the generation of correct perspective images captured by catadioptric sensors. Even more interesting, it allows a simplification of projections models and therefore a simplifica-

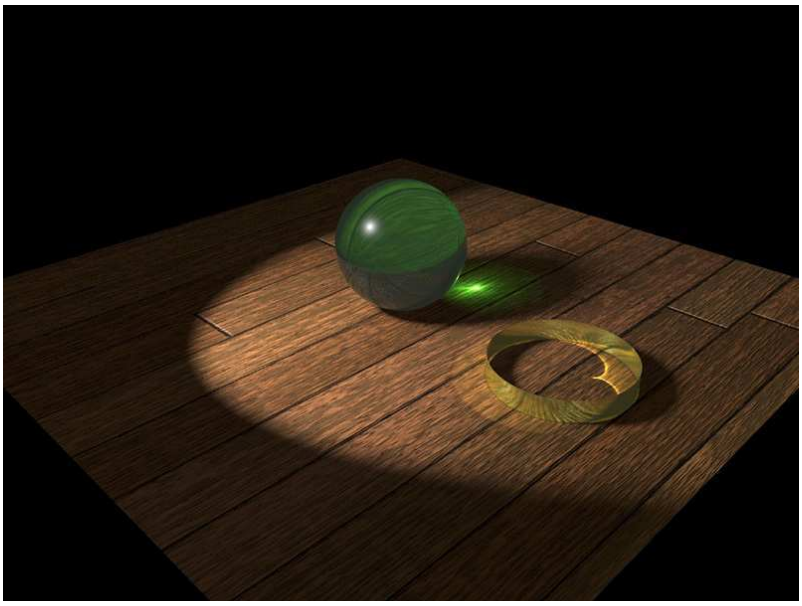


Figure 4.1: Two examples of caustics: A caustic caused by the refraction of a ray on the sphere and the bright light pattern inside the ring is the caustic for this scenario. [source: Nvidia Gelato Image Gallery]

Parabola	$\sqrt{x^2+y^2+z^2} = 2p - z$
Hyperbola	$\frac{(z-\frac{d}{2})^2}{a^2} - \frac{x^2}{b^2} - \frac{y^2}{b^2} = 1$
Ellipse	$\frac{(z-\frac{d}{2})^2}{a^2} + \frac{x^2}{b^2} + \frac{y^2}{b^2} = 1$
Plane	$z = \frac{d}{2}$

Table 4.1: Conic Equations

tion of theoretical and practical treatments.

Baker and Nayar (1998) derived the entire class of catadioptric systems with a single viewpoint which can be constructed using just a single conventional lens and a single mirror. The four configurations that have this property are an orthographic camera associated to a parabolic mirror or a perspective camera associated to a hyperbolic, elliptical or planar mirror. Figure 4.2 adapted by Mei (2007) from the work of Barreto (2003) depicts these cases under the assumption of the pinhole camera model:

1. *Parabolic mirror coupled with orthographic projection*: In this case, the single viewpoint is the focus (F) of the parabola. The distance between the camera and the mirror is not constrained. The single viewpoint constraint is verified whenever the camera is orthographic and the optical axis is aligned with the axis of the paraboloid. An orthographic projection can be obtain using telecentric optics [Watanabe and Nayar (1995)]. This system is somehow difficult to construct since the telecentric lens must be as wide as the mirror, which is why this sys-

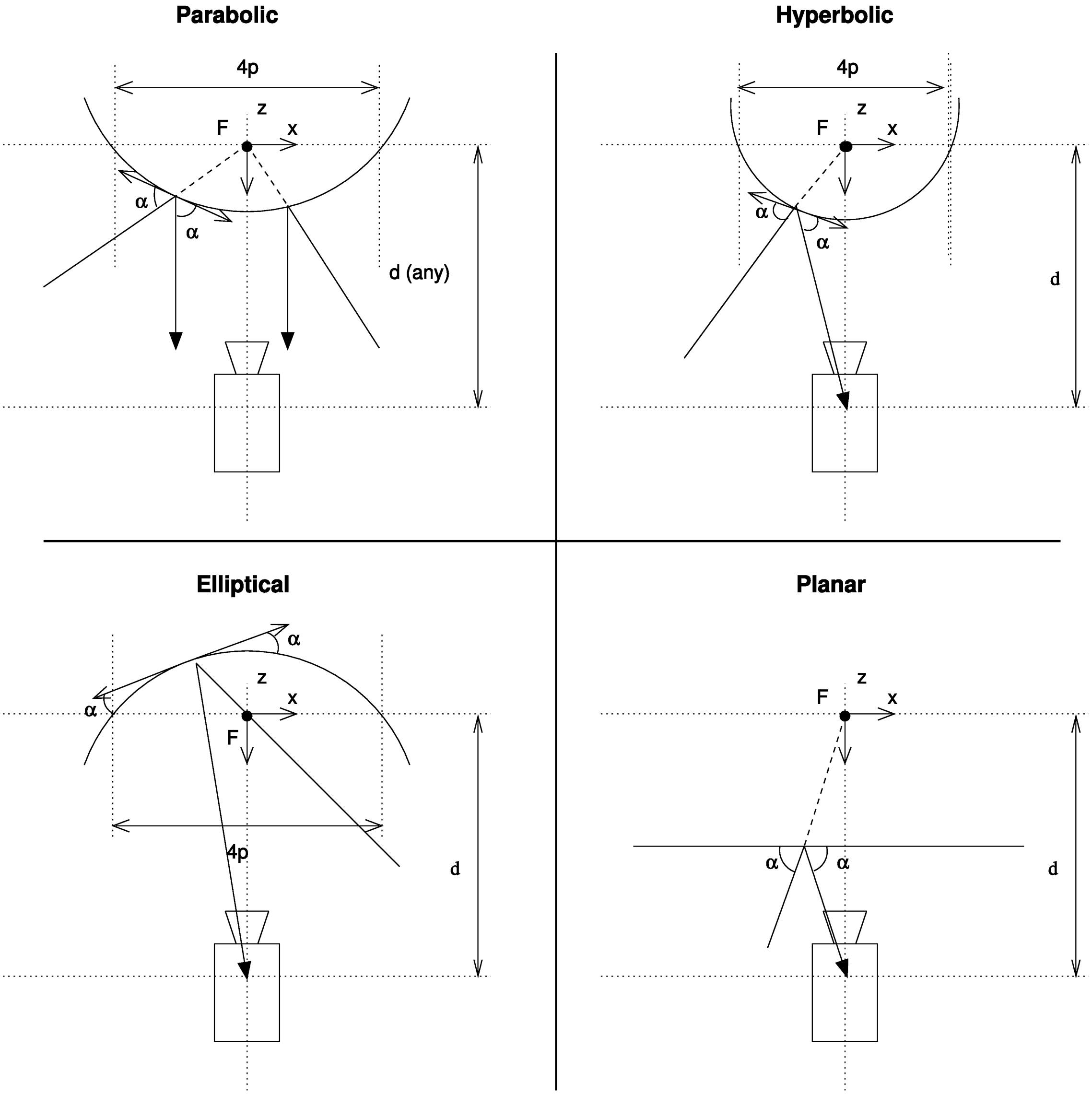


Figure 4.2: Catadioptric Sensors with a Single Viewpoint
[Source: Mei (2007)]

tems are very expensive. A ray of light incident with the focus of the parabola is reflected by the mirror to a ray of light parallel to its axis. The equation of the paraboloid is given in table 4.1. This system is often called *paracatadioptric camera*.

2. *Hyperbolic mirror coupled with perspective projection*: The design of this systems is very delicate. The pinhole camera have to be placed so as to have its focus on one foci of the hyperboloid. Any light ray going through the inner focus is reflected in another ray passing through the outer focus, i.e., the central point and the optic center coincide with both focus of the hyperboloid. Thus, by the reflective properties of the hyperboloid, when the rays reflect on the hyperbolic mirror they target the second focus, on doing so, the second focus acts as the single viewpoint. The corresponding hyperboloid equation is provided in table 4.1, where d is the distance between foci, $a = 1/2(\sqrt{d^2 + 4p^2} - 2p)$ and $b = \sqrt{p(\sqrt{d^2 + 4p^2} - 2p)}$

3. *Elliptic mirror coupled with perspective projection*: Similar to the hyperbolic case, the perspective camera must be positioned so that its optical center coincides with one of the two foci of the ellipsoid and orientated towards the other focus (the central point). Due to the fact that the perspective camera and the reflective surface are inside the ellipsoid, the field of view is slightly increased (less than half-sphere), which is why it is not used in practice. The corresponding ellipsoid equation is shown in 4.1, where d is the distance between foci, $a = 1/2(\sqrt{d^2 + 4p^2} + 2p)$ and $b = \sqrt{p(\sqrt{d^2 + 4p^2} + 2p)}$.

4. *Planar mirror with perspective projection*: A planar mirror also verifies the single viewpoint constraint. However, this is a degenerate case that bring us back to classical perspective vision, thus without any practical interest nor the possibility of increase the field of view. The effective projection center is behind the mirror in the perpendicular line passing through the camera center. As shown in figure 4.2 its distance to the camera center is twice the distance between the planar mirror and the camera.

The degenerated systems proposed by Baker and Nayar (1998) are the *spherical mirror with perspective projection* and the *conical mirror with perspective projection*. As mentioned previously, this configurations does not satisfy the single viewpoint constraint. Nevertheless, it has been proved that under geometric optics image formation model, the single viewpoint of cone mirror is actually realizable (Lin and Bajcsy (2001)). The interest on the conic system is due to the fact that cones are much more cheaper to manufacture with higher resolution. The narrow field of view on the contrary, makes it less attractive for robotics applications. As for the spherical mirror, we need to place the camera in the center of the sphere in order to obtain a single viewpoint. The central projection point will then coincide with the optical center and the center of the sphere, thus only seeing the camera itself. On the other hand, besides that spherical mirrors are less expensive, they are also easier to calibrate, which is why they are mostly used to built non central catadioptric sensors.

Geyer and Daniilidis (2000) propose a unified model for every central catadioptric system (figure 4.2) using the spherical perspective projection. In his thesis, he gave a geometrical proof of the equivalence of the projection on a quadric surface and the projection on the sphere. Later, a modified version was proposed by Barreto (2003). We will present this unified model in section 4.2.2. In the following, we explain geometrically the equivalence between the projection on a quadric and the projection on the sphere. This projections are well known, but presented here for the sake of completeness.

4.2.1 Quadric projection vs Sphere projection

In figure 4.3, the left side describes the classic projection model of a 3D point in the parabolic case. If the ray FP that cross the 3D point P and the focal point F intersects the mirror in M_1, it will be reflected parallel to the optical axis and will intersect the image plane in Q.

On the other hand, the right side of the figure represents the same projection using a sphere centered in F. The intersection of the ray FP and the circle is the point M_2. So, the 3D point P is first projected to M_2 and then projected from the North pole N to the image plane in the same point Q.

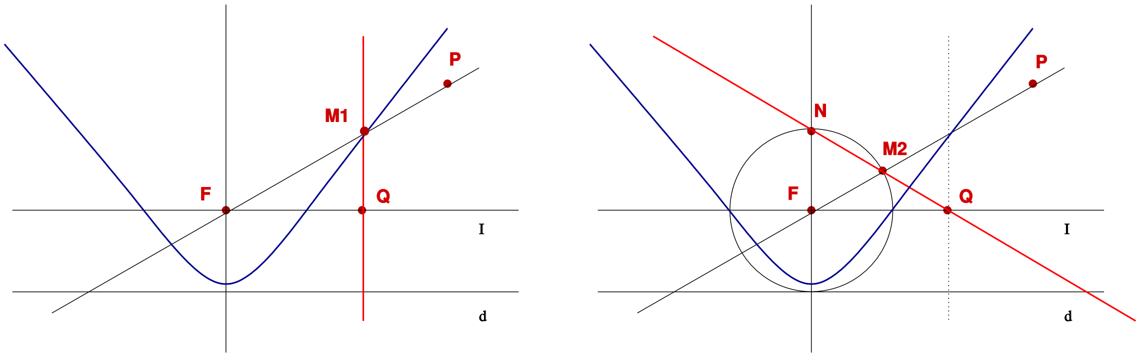


Figure 4.3: Quadric projection vs sphere projection

The parabolic projections of a 3D point P to I, in both cases, are coincident in Q, therefore equivalents. It is also possible to find an equivalence of this model for the hyperbolic case by choosing an appropriate projection center on the axis between the north pole N and F. Its position will depend on the shape of the mirror.

4.2.2 Unified Projection Model

In this section the Unified Projection Model (figure 4.4) proposed by Mei (2007) is presented, which is an extension of the models of Geyer (2003) and Barreto (2003).

	ξ	γ
Parabola	1	$-2pf$
Hyperbola	$\frac{df}{\sqrt{d^2+4p^2}}$	$\frac{-2pf}{\sqrt{d^2+4p^2}}$
Ellipse	$\frac{df}{\sqrt{d^2+4p^2}}$	$\frac{2pf}{\sqrt{d^2+4p^2}}$
Planar	0	-f
Perspective	0	f
d: distance between focal points $4p$: latus rectum		

Table 4.2: Unified Model Parameters

The projection of 3D points can be done by following the next steps:

1. The first step is the projection of the world points in the mirror frame $\boldsymbol{\mathcal{F}}_m$ onto the unit sphere,

$$(\boldsymbol{\mathcal{X}})_{\boldsymbol{\mathcal{F}}_m} \rightarrow (\boldsymbol{\mathcal{X}}_s)_{\boldsymbol{\mathcal{F}}_m} = \frac{\boldsymbol{\mathcal{X}}}{\|\boldsymbol{\mathcal{X}}\|} = (X_s, Y_s, Z_s)$$

2. Then, the points are changed to a new reference frame centered in $\boldsymbol{\mathcal{C}}_p = (0, 0, \xi)$,

$$(\boldsymbol{\mathcal{X}}_s)_{\boldsymbol{\mathcal{F}}_m} \rightarrow (\boldsymbol{\mathcal{X}}_s)_{\boldsymbol{\mathcal{F}}_p} = (X_s, Y_s, Z_s + \xi)$$

where the parameter $\xi \in [0, 1]$ depends on the geometry of the mirror.

3. Then, the points are projected onto the normalized image plane π_m,

$$\mathbf{m} = (\frac{X_s}{Z_s + \xi}, \frac{Y_s}{Z_s + \xi}, 1) = \hbar(\boldsymbol{\mathcal{X}}_s)$$

4. Finally, the coordinates of the points in the image frame are given by,

$$\mathbf{p} = \mathbf{K}\mathbf{m} = \begin{bmatrix} \gamma_1 & \gamma_1 s & u_0 \\ 0 & \gamma_2 & v_0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{m} = k(\mathbf{m})$$

where $\mathbf{K}$ is the so called camera projection matrix with (γ_1, γ_2) the generalized focal lengths, (u_0, v_0) the principal point and s the skew. The values for ξ and γ (detailed in table 4.2) are related to the mirror parameters (the ideal case $\gamma_1 = \gamma_2$).

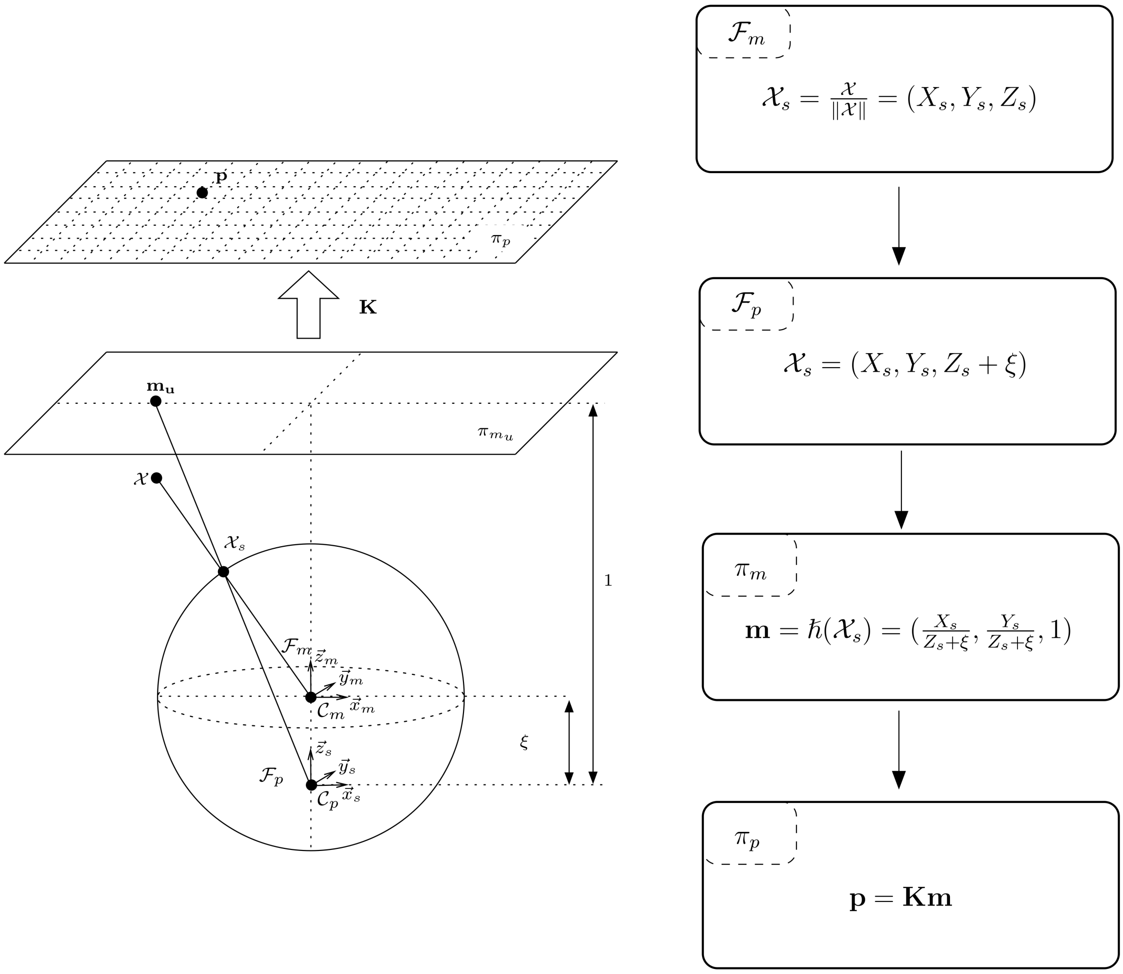


Figure 4.4: Unified Projection Model

4.2.3 Inverse Unified Projection Model

In the above section we explain the steps of the *direct* unified projection model, i.e., how to project a 3D point in the image plane. If we consider a pixel $\mathbf{p}$ in the image plane and we wish to know the coordinates of this point associated to the sphere frame, we applied the *inverse* unified projection model. As the function $\hbar$ is bijective then,

$$\hbar^{-1}(\mathbf{m}) = \begin{bmatrix} \frac{\xi+\sqrt{1+(1-\xi^2)(x^2+y^2)}}{x^2+y^2+1}x \\ \frac{\xi+\sqrt{1+(1-\xi^2)(x^2+y^2)}}{x^2+y^2+1}y \\ \frac{\xi+\sqrt{1+(1-\xi^2)(x^2+y^2)}}{x^2+y^2+1} - \xi \end{bmatrix} = \boldsymbol{\mathcal{X}}_s \tag{4.1}$$

The calculation of the point $\boldsymbol{\mathcal{X}}_s$ corresponding to a given point $\mathbf{p}$ is called *lifting*.

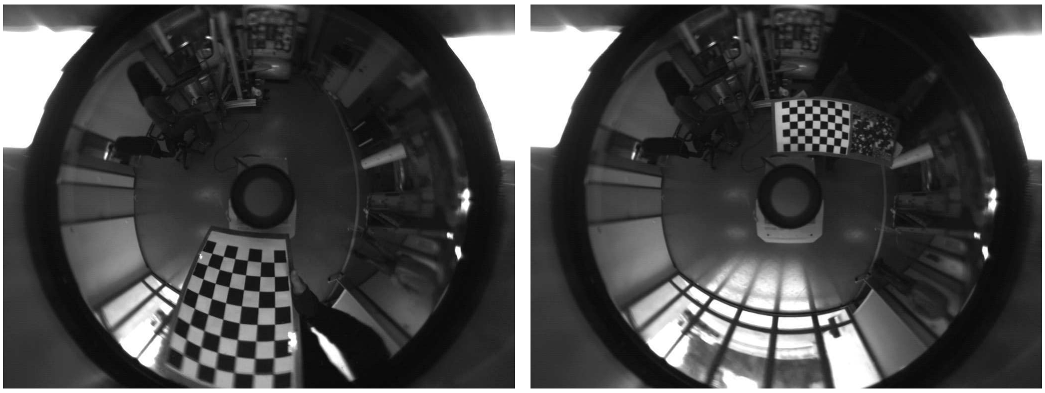
4.2.4 Calibration Issues

Despite the fact that omnidirectional cameras are a popular choice as visual sensors for robotics because of its many advantages, their practical use require a calibration phase which is not always evident and mostly can be time consuming. This is why in the last years, calibrations techniques for omnidirectional sensors have been the focus of attention of many researchers and now there are efficient methods to carry out this task. Precise calibration is a crucial step as it will later have an impact over the quality of the reconstruction and pose estimation.

The method that most of the researchers used is by projection of lines, in which, only the lines in the scene detected on the image are used (no need of any other metric information). Geyer and Daniilidis (1999) propose to use the detection of two sets of parallel lines in the scene in order to recover the intrinsic parameters of a parabolic catadioptric system. Barreto and Araujo (2005) studied the geometry of the central catadioptric projection of lines and used for calibration. They showed that any central catadioptric system can be fully calibrated from the image of a minimum of three lines. Ying and Hu (2004), propose a variant of the method that uses projection of lines and projections of spheres. They proved that the projection of spheres is more robust and has higher accuracy for calibration than using projection of lines.

Another method is by knowing world coordinates. This method uses a calibration patterns or calibration grids with control points whose 3D world coordinates are known [Vasseur and Mouaddib (2004), Scaramuzza et al. (2006)]. This methods are popular because they are accurate and easy to use.

Particularly, in order to recover the intrinsic parameter of the camera we used the method proposed by Mei and Rives (2006a, 2007) which relies on minimizing the reprojection error of points of a planar grid of known dimension. This method allows to accurately calibrate any catadioptric system without the need of knowing the mirror parameters. Additionally, it is possible to recover the parameters of the



(a)

(b)

Figure 4.5: Images used for the calibration of the camera

mirror. Two examples of the omnidirectional images used in this calibration step are shown in figure 4.5, where the calibration grid is visible. The results of the calibration for both mirrors are shown in table 4.3.

	Parabolic mirror	Hyperbolic mirror
Focal Length (γ_1)	255.9681	284.15152
Focal Length (γ_2)	262.7848	284.85383
x_i (ξ)	1	0.8711
Principal Point (u_0)	505.6540	519.96492
Principal Point (v_0)	393.6585	385.02783

Table 4.3: Calibration results for both mirrors

Calibration Laser and Catadioptric Camera

Once the laser and the camera has been calibrated separately, it is necessary to calibrate the laser range finder and the omnidirectional camera to find their extrinsic parameters, in order to be able to combine the data of both sensors. In other words, we need to find the rigid transformation from the camera coordinate system to the laser coordinate system.

There have been some studies for the extrinsic calibration of a camera and a laser, probably all based on the work of Zhang and Pless (2004) for standard perspective cameras and lasers with invisible beam. The idea was to combine a plane with known position (calibration grid) with the calculated distances obtained from the laser.

The work of Zhang and Pless was extended by Mei and Rives (2006b) to calibrate central catadioptric cameras and lasers with visible and invisible beam. We used this last approach in order to find the relative pose (rotation and translation) between the

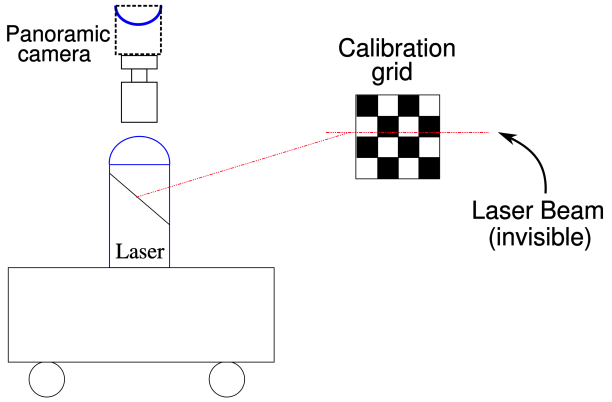


Figure 4.6: Association between a calibration grid and the sensors

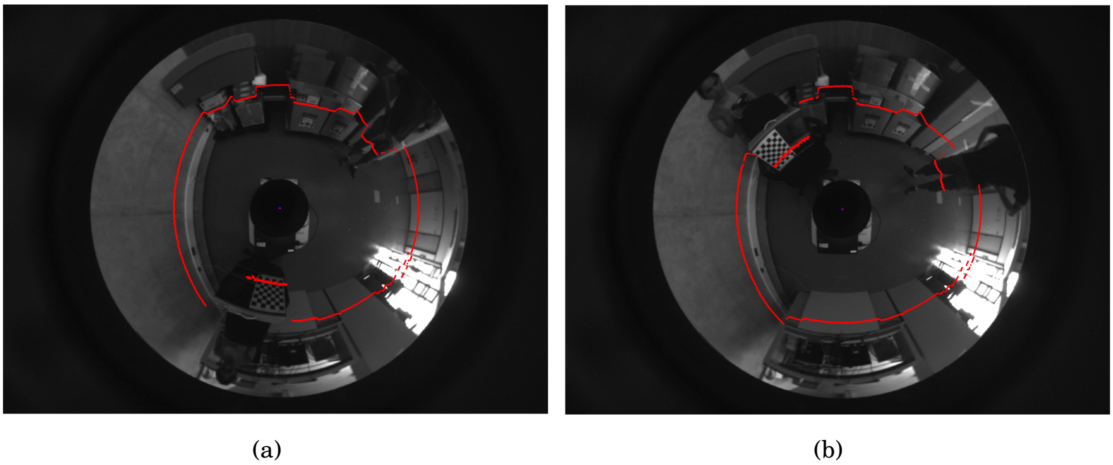


Figure 4.7: Laser data projected on omnidirectional images after calibration

sensors. Several omnidirectional images and laser scans were taking simultaneously as shown in figure 4.6. It is important that the calibration grid is visible in the images, and likewise, the laser beam crosses the calibration grid.

The results are shown in figure 4.7, where the laser points were reprojected into the omnidirectional image using the projection model described in section 4.2.2. It can be seen, that the results are quite accurate, which will be of vital importance for merging image and laser data for our hybrid sensor. We will see in further sections that in practice, calibration is not always perfect, and that even a small error in the calibration step will cause wrong data association.

4.3 Omnidirectional Lines Parametrization

A 3D line projected in a monocular imaging device can be parameterized by the normal noted $\mathbf{n}$ ($\mathbf{n} \in S^2$) formed by the line and the center of projection (see figure 4.8).

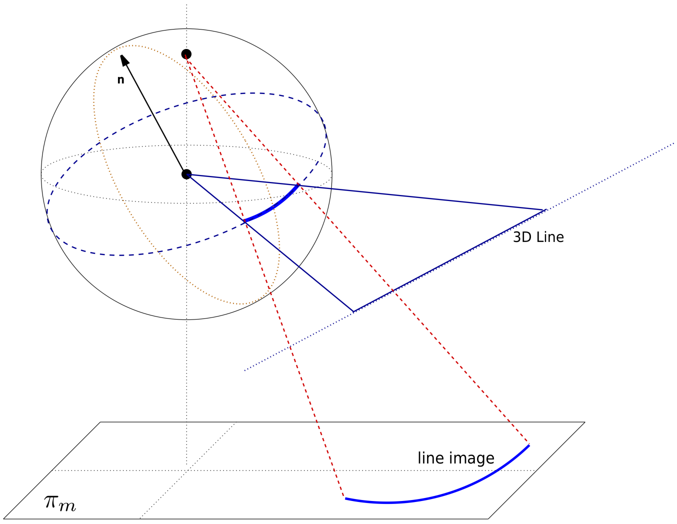


Figure 4.8: Projection of a 3D line in the image.

In the previous section, equation 4.1 relates a point on the normalized plane to a projective ray through the mirror center and $\boldsymbol{\mathcal{X}}_s$. Therefore, by multiplying by $\xi\frac{x^2+y^2+1}{-\xi-\sqrt{1+(1-\xi^2)(x^2+y^2)}}$, the projective equality is obtained:

$$\begin{cases} \hbar^{-1}(\mathbf{m}) \sim \begin{bmatrix} x \\ y \\ f(x,y) \end{bmatrix} \\ f(x,y) = 1 + \xi\frac{x^2+y^2+1}{-\xi-\sqrt{1+(1-\xi^2)(x^2+y^2)}} \end{cases} \tag{4.2}$$

This equation is valid for any central catadioptric device. When the camera is calibrated the values for $f(x,y)$ can be precalculated. The relation between $\mathbf{m}$ and $\mathbf{p}$ is linear and not very costly to compute (in particular if $r = 1$ and $s = 0$ which is often the case with modern cameras).
A point $\mathbf{p}$ on an omnidirectional line of parameter $\mathbf{n}$ verifies:

$$(\mathbf{n}^\top)\begin{bmatrix} x \\ y \\ f(x,y) \end{bmatrix} = 0 \tag{4.3}$$

This result will be useful for the extraction of the radial lines in the image, which will be explained more in detail in the next section.

4.4 Omnidirectional feature extraction

This section aims, once the camera has been calibrated, to describe the omnidirectional image processing used to make the most of the information provided by the omnidirectional camera. This processing will be used later to correct (relocate) the pose estimation of the laser-based approach and to reconstruct a 3D map of the surroundings of the robot. It is a well known fact that the efficiency and robustness of SLAM procedure is directly influenced by the choice of the map representation type. One way of modeling the environment structure is with geometric primitives, which provide a concise environment description. As our work focuses on indoor environments, we will be interested in a "low-level" feature extraction of *lines*, which are more like to characterize office-like indoor environments.

It has been proven [Geyer and Daniilidis (2002); Barreto (2003); Ying and Hu (2004)] that in central catadioptric systems, lines in the scene –corresponding to approximately vertical features (e.g. walls, facades, doors, windows...)– are projected to quasi radial lines in the image. Particularly, line images are circles in paracatadioptric images and conics in the hyperbolic case. The camera-mirror system was set to be perpendicular to the floor where the robot moves, which can guarantee that vertical lines in the scene are approximately mapped to radial lines on the camera image plane.

Before resuming our work, we shall define *low-level features* to be those basic features that can be extracted automatically from an image without any shape information (i.e., information about *spatial* relationships). It will allow us to work directly with the grey level binary image (raw data) obtained with the omnidirectional camera.

Canny edge detector

The first step of our low-level feature extraction of lines scheme is *edge detection*. This step is performed using Canny edge detector, which is a first order differentiation operator. Canny (1986) described in his paper, three performance criteria of his edge detection algorithm, namely:

1. Good detection: This aims to reduce the response to noise (using optimal smoothing). Canny was the first to demonstrate that Gaussian filtering is optimal for edge detection.

2. Good localization: which means a small bias in edge position with respect to the true edge. This was achieved with a *non-maximum suppression* process (which is equivalent to peak detection). This results in *thinning*: thin lines of edge points in the right place.

3. Only one response to a single edge: which means location of a single edge point in response to a change of brightness.

Because of its performance criteria, Canny edge detection operator is suitable for our purposes. It was applied to the grey level omnidirectional image to obtain a binary edge image. Figure 4.9(b) show the performance of the Canny edge detector

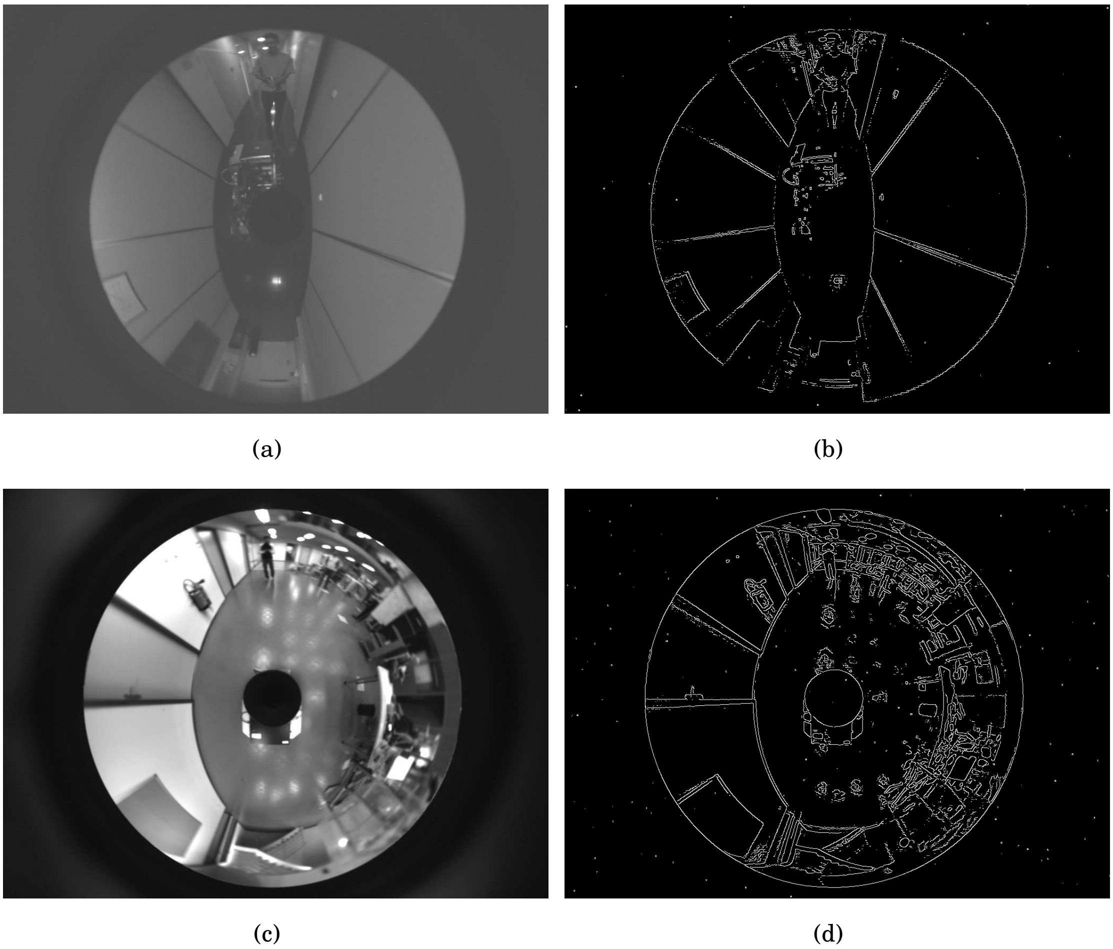


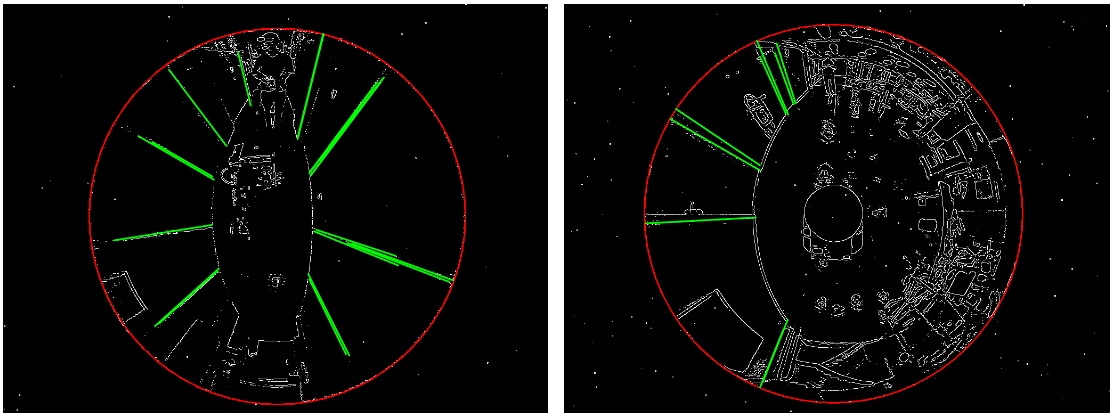
Figure 4.9: Canny edge detector: 4.9(b) Performance in a corridor. 4.9(d) Performance in the robotic hall.

for an image taken in a corridor, while in figure 4.9(d), the image was taken at the robotic hall. It is clear the accuracy of Canny edge detection performance.

Randomized Hough Transform for line extraction

The next step consist in applying the Randomized Hough Transform (RHT) –which is a probabilistic variant of the classical Hough Transform– to the binary edge image in order to extract lines in the omnidirectional image. The RHT has proved to overcome with the drawbacks of the classical Hough Transform (HT), being an efficient alternative [Xu and Oja (1993)]. Furthermore, it has the advantages of fast speed, small storage and high parameter resolution.

The idea of the RHT, is to avoid the voting procedure of the HT for detecting potential curves in the image by taking advantage of the fact that some curves can be fully characterized by a certain numbers of points on the curve. For example, a straight line can be determined by two points, an ellipse and a circle can be determined by



(a)

(b)

Figure 4.10: Randomized Hough Transform and circle detection

three points. the essence of the RHT process generally consists on the combination of three steps: *random sampling* in the image space, the *score accumulation* in the parameter space and *converging mapping* as the bridge between the two spaces.

Thus, the RHT estimate the n numbers of points on the curve by randomly extracting n values. This random data selection is also common in the RANSAC approach. Actually, RANSAC [Fischler and Bolles (1981)] use as well a combination of random sampling and converging mapping. The main difference between RANSAC and RHT is that RANSAC does not use *score accumulation* in the parameter space. Ransac is rather, a guess and test method.

Duda and Hart (1972) introduce the (ρ, θ) polar parametrization to the HT, which makes HT more efficient for line detection, moreover, they also demonstrated how a circle can be detected with HT. Xu and Oja (1993) used and extended this parametrization to estimate the parameters of a line. Mei and Malis (2006) described a more efficient parametrization for the omnidirectional lines, which can be obtained by directly estimating the normal. Thus the line image joining two points $\mathbf{m}_1$ and $\mathbf{m}_2$ has for normal $\mathbf{n}$ $(\hbar^{-1}(\mathbf{m}) \in S^2)$:

$$\mathbf{n} = \hbar^{-1}(\mathbf{m}_1) \times \hbar^{-1}(\mathbf{m}_2) \tag{4.4}$$

This parametrization allows us to obtain a 2-dimensional buffer in (n_x, n_y) by imposing $n_z \geq 0$.

In addition, since we are interested only in *quasi-radial* lines of the omnidirectional image, the calibration of the camera is needed to recover the image center (i.e., where all radial lines intersect). The image center can be corrected using a circle detection (also with Hough Transform). The image center will help us filter out the lines detected by the RHT that do not lie on radial directions. The results for two indoor environments, are shown in figure 4.10, where the radial lines are depicted in green and the result of the circle detection is depicted in red.

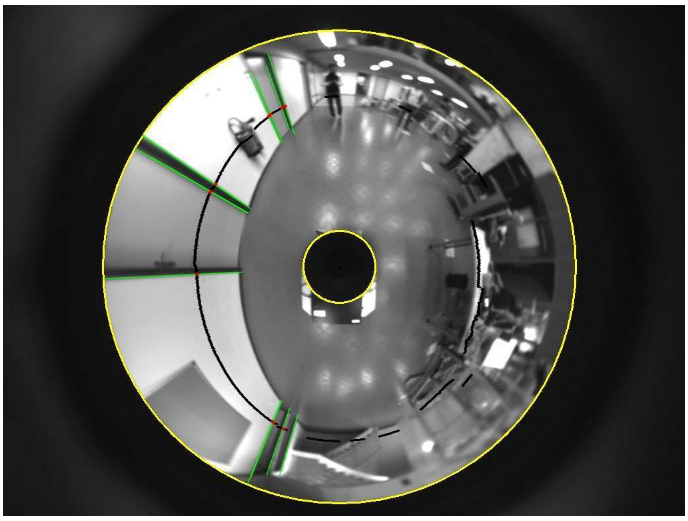


Figure 4.11: Detection of vertical lines and the corresponding laser measurements.

4.5 3D Vertical line extraction from omnidirectional images and laser scans

This section explains the procedure we developed to extract vertical lines from omnidirectional images and to estimate their 3D positions using information from the laser range finder.

Firstly, we project the laser information on the omnidirectional image in order to get an approximation of the depth information missing in the image. To achieve this, the unified projection model defined in section 4.2.2 is applied.

The generalized camera projection matrix $\mathbf{K}$ is computed from the generalized focal lengths (γ_1, γ_2) and the principal point (u_0, v_0):

$$\mathbf{K} = \begin{bmatrix} \gamma_1 & 0 & u_0 \\ 0 & \gamma_2 & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

Using $\mathbf{K}$, we can compute the normalized coordinates of a point $\mathbf{p}$ in the image (represented in the camera coordinate frame) as $\mathbf{m} = [x, y, 1]^T = \mathbf{K}^{-1}\mathbf{p}$. We then compute $\boldsymbol{\mathcal{X}}_s = [X_s, Y_s, Z_s]$ as follows (see figure 4.4):

$$\boldsymbol{\mathcal{X}}_s = \begin{bmatrix} \frac{\xi+\sqrt{1+(1-\xi^2)(x^2+y^2)}}{x^2+y^2+1}x \\ \frac{\xi+\sqrt{1+(1-\xi^2)(x^2+y^2)}}{x^2+y^2+1}y \\ \frac{\xi+\sqrt{1+(1-\xi^2)(x^2+y^2)}}{x^2+y^2+1} - \xi \end{bmatrix}$$

where ξ is the mirror parameter, which is equal to 1 for parabolic mirrors.

Then we extract the quasi-radial lines in the scene (as explained in the previous section), corresponding to approximately vertical features (e.g. walls, facades, doors,

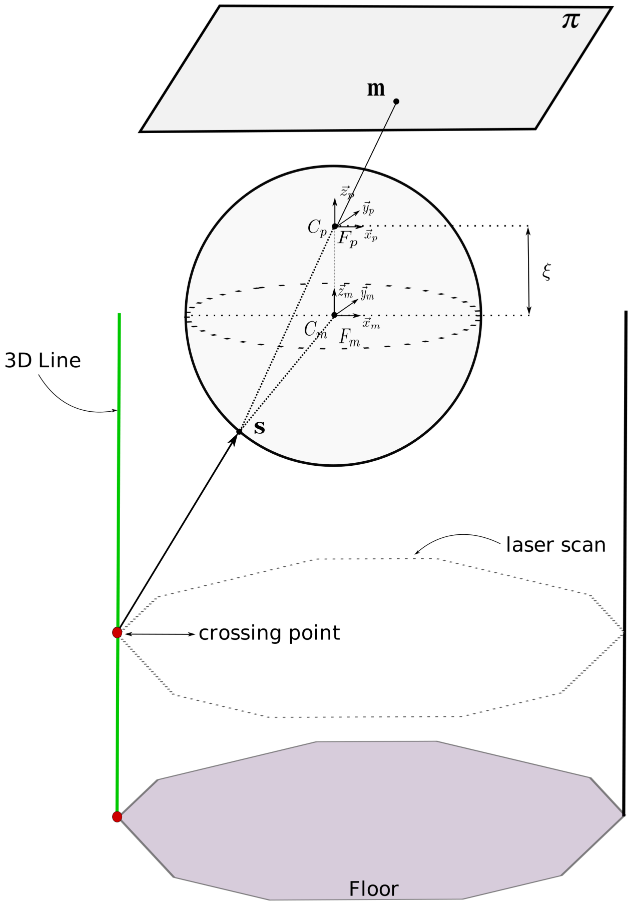


Figure 4.12: Extraction of 3D lines

windows). As we set the camera-mirror system perpendicular to the floor where the robot moves, we can guarantee that vertical lines are approximately mapped to radial lines on the camera image plane.

As shown in figure 4.11, by overlapping in the omnidirectional image the laser scan data and the radial lines we can find a laser range measurement corresponding to each vertical line. This allows to recover the depth information missing. We detect those laser measurements and save them in the original camera frame together with its corresponding point in the image plane (which also corresponds to a point on a vertical line).

Let $\mathbf{M}_0^s = [X_0^s, Y_0^s, 0]^T$ be a laser measurement lying on a vertical line expressed in the camera coordinate frame, Δ a 3D plane defined in the camera frame, and $\mathbf{m}_i^s = [x_i^s, y_i^s, z_i^s]^T$, $i = 1, 2$ the end points of the vertical line where the laser measurement lies expressed in the sphere (mirror) coordinate frame (see figure 4.12). These last points are computed by inverting the projections of the unified model as in depicted in section 4.2.3.

We reconstruct the 3D lines as follows. Let $\mathbf{u}^s$ be the director vector. For every

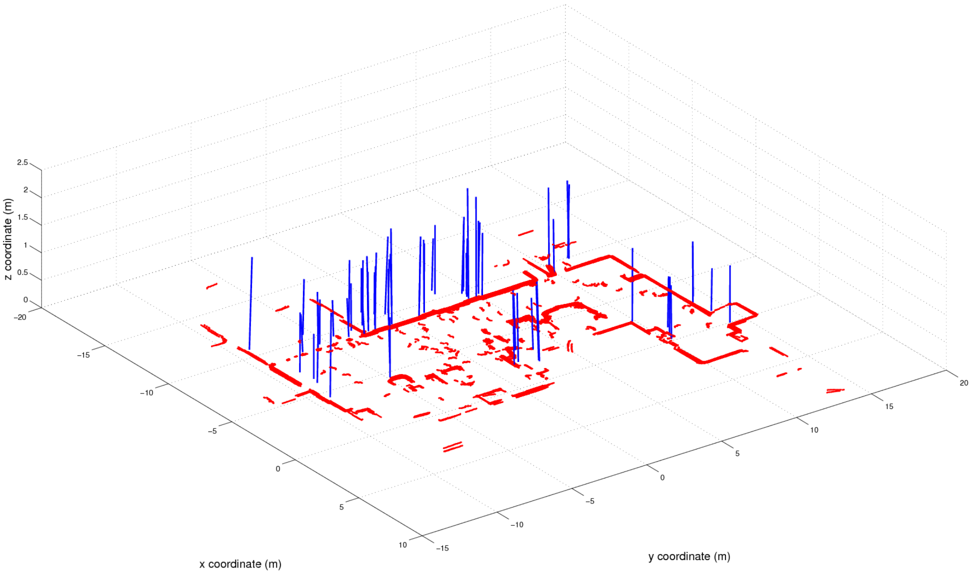


Figure 4.13: Environment with 3D lines

$\mathbf{M}_i^s \in \Delta$, the vector $\overrightarrow{\mathbf{M}_0^s\mathbf{M}_i^s}$ is collinear to $\mathbf{u}^s$. Thus,

$$\overrightarrow{\mathbf{M}_0^s\mathbf{M}_i^s} = \lambda_i \mathbf{u}^s \quad \Longrightarrow \quad \begin{cases} X_i - X_0^s = \lambda_i u_x^s \\ Y_i - Y_0^s = \lambda_i u_y^s \\ Z_i - Z_0^s = \lambda_i u_z^s \end{cases} \tag{4.5}$$

$$\overrightarrow{\mathbf{OM}_i^s} = \mu_i \overrightarrow{\mathbf{O}^s\mathbf{m}_i} \quad \Longrightarrow \quad \begin{cases} X_i = \mu_i x_i^s \\ Y_i = \mu_i y_i^s \\ Z_i = \mu_i z_i^s \end{cases} \tag{4.6}$$

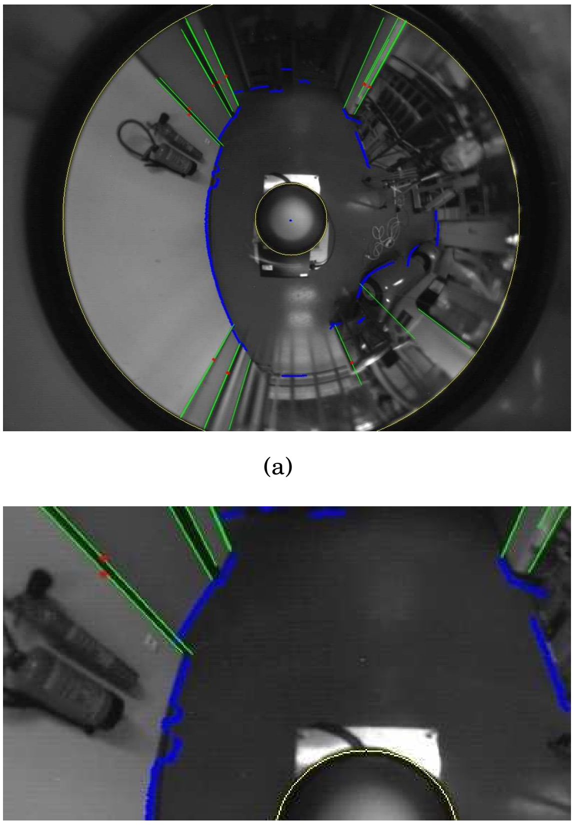
Substituting (4.6) in (4.5) we get the following system of equations

$$\begin{cases} \mu_i x_i^s - X_0^s = \lambda_i u_x^s \\ \mu_i y_i^s - Y_0^s = \lambda_i u_y^s \\ \mu_i z_i^s - Z_0^s = \lambda_i u_z^s \end{cases} \tag{4.7}$$

If Δ is a vertical plane in the sphere frame R_s, i.e. $\mathbf{u}^s = [0, 0, 1]^T$, then:

$$\begin{cases} \mu_i x_i^s - X_0^s = 0 \\ \mu_i y_i^s - Y_0^s = 0 \\ \mu_i z_i^s - Z_0^s = \lambda_i \end{cases} \tag{4.8}$$

Since $[x_i^s, y_i^s, z_i^s]^T$ and $[X_0^s, Y_0^s, 0]^T$ are known, we can compute μ_i for each i. Then we substitute in equation (4.6) to obtain the extreme points of the lines in Δ. Finally, we apply the homogeneous transformation to transform the coordinates of those points to the global coordinate system and trace the 3D lines. The result is shown in figure 4.13. We can observe that the vertical lines extracted are consistent with the 2D map.



(a)

(b)

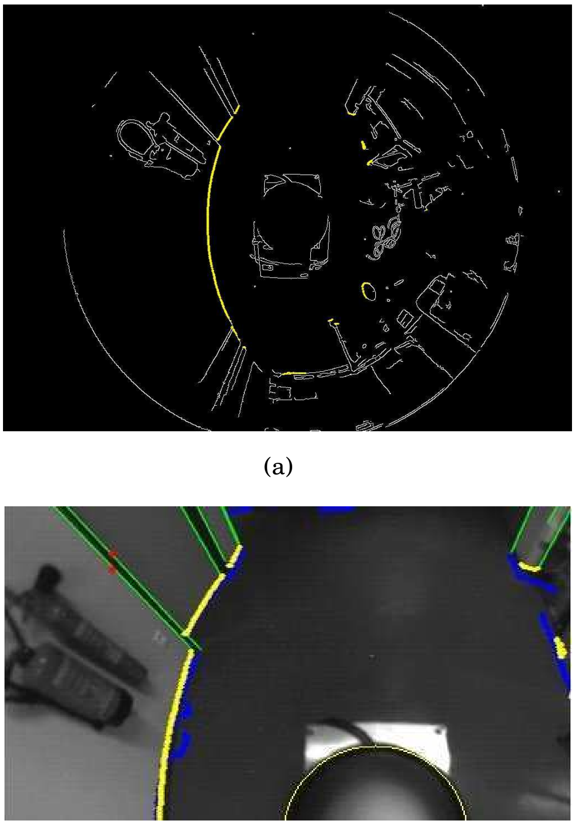
Figure 4.14: Floor detection. 4.14(a) Line extraction and Laser re-projection shifted at floor level. 4.14(b) Fire-extinguisher breaks the planarity hypothesis.

4.6 Floor detection from omnidirectional images and laser scans

As stated in the previous section, it is assumed that the distance between the laser frame and the floor is approximately known (which requires the plane to be horizontal). It is also assumed that the pose between the camera and laser frames is correctly estimated. Under these hypotheses, the laser scan can be shifted along the vertical lines and used to predict where a virtual laser trace, corresponding to the floor, should project onto the omnidirectional image.

Due to calibration errors, the predicted trace does not exactly match the real boundary of the floor observed in the image. In practice, the neighborhood of the predicted trace is searched for the closest element of contour detected in the image. This match is finally taken as the intersection between the floor plane and the walls and will be integrated into a partial 3D model.

As an example of this procedure, figure 4.14(a) shows the predicted laser measurements projected onto the floor in blue. From the zoomed image in figure 4.14(b)



(a)

(b)

Figure 4.15: Floor Correction. 4.15(a) Corrected floor detection. 4.15(b) Blue trace: Laser Scan shifted to the floor, yellow trace: Corrected floor detection

it can be seen that in certain cases the prediction may be wrong as in the case of the fire-extinguisher which breaks the planarity hypothesis of the wall. Figure 4.15(a) shows in white the result of applying the Canny edge detector to the entire image and in yellow the detected edge points found by searching the neighborhood of the laser floor plane, therefore giving a more accurate floor estimation. A better view of this correction can be seen in Fig. 4.15(b) where overlapped traces are shown.

4.7 Conclusion

In this chapter we have presented an original composite sensor approach that takes advantage of the information given by the omnidirectional camera and a laser range finder. A tightly integration of the raw data of both sensors was used for sensor fusion. We have proved to efficiently solve the Simultaneous Localization and mapping problem for indoor environments, and that using lines is possible to build a 3D wired representation of the environment. Furthermore, a simple but accurate procedure to detect the floor in the image is developed, which completed our laser/omnidirectional sensor that enhance both, localization and map representation of the robot's environment.

In our approach, the laser provides metric information of the environment that helps to fix a scale factor (removing the difficulty of propagating the scale factor) without the need to use multiple cameras. We have identified several advantages of combining laser and visual sensors. Our experimental results are encouraging and give us valuable insight into the possibilities offered by this composite sensor approach.

The next chapter will concentrate on an extension of the EPSM algorithm to exploit the information about vertical lines detected using omnidirectional images. Thanks to the accurate segmentation of the ground (floor), it will be possible to extract planes on the image that would allow a dense (textured) 3D reconstruction by warping the images onto the geometric model of the world. Finally, we believe the general approach can be extended to solve the full six degrees of freedom (6DOF) SLAM problem, which is an active field of research.

"We are judged by what we finish, not by what we start."
Johann Wolfang von Goethe

5
Appearance-Based SLAM

Overview

This chapter aims to give a brief introduction to visual SLAM and the different approaches found in the computer vision community. Basically, visual SLAM solve the simultaneous localization and mapping problem using visual landmarks (i.e., interest features), which are extracted and matched in successive images. The pose of the robot and of the features in the world are determined based on their observed relative movement. The images are used to map the position of the robot in the environment. Furthermore, it provides an additional odometry source, which is useful to define the location of the robot. Positions and orientations are relative to a reference frame, and therefore the definition of a pose can be shown to be equivalent to rigid body motion. A short overview of rigid body motion is given in Appendix C in order to state the basic terminology used in the following.

The second part of this chapter will describe a novel and efficient hybrid laser/vision appearance-based SLAM, in order to provide the mobile robot with rich 3D information about the environment. By combining the information from an omnidirectional camera and a laser range finder, reliable 3D positioning and an accurate 3D representation of the environment is obtained subject to illumination changes even in the presence of occluding and moving objects. A scan matching technique is used to initialize the tracking algorithm in order to ensure rapid convergence and reduce computational cost.

This approach complements the laser-based localization method described in chapter 3 and relies on the tightly coupled sensor developed in chapter 4.

Keywords: Visual SLAM, appearance-based SLAM, rigid body motion, hybrid sensor, spherical warping, 3D dense reconstruction.

Organization of this chapter:

This chapter is organized as follows:

Section 5.1 gives a brief introduction to visual SLAM and explains the different approaches found in the computer vision community.

Sections 5.2 and 5.3 describe the two major approaches for visual SLAM: feature-based approach and direct approach. The advantages and limitations of both methods are discussed.

Section 5.4 describes a novel generic robot-centered representation that is well adapted to the appearance-based SLAM method. It explains how central omnidirectional cameras can be modeled using two consecutive projections, i.e., a spherical projection followed by a perspective one.

Section 5.5 proposes an appearance-based localization method which minimizes a nonlinear cost function directly built from the augmented spherical view defined in section 5.4. It explains in detail how the sphere to sphere mapping is performed and each step of the our appearance-based SLAM method.

Section 5.6 presents the results that validate our method with real data obtained by the mobile robot in an indoor office-like environment.

Section 5.7 gives the conclusion of the chapter.

5.1 A survey of Visual SLAM

Estimating the motion of a camera while simultaneously reconstructing the environment in which it navigates is called *Visual Slam* [Davison (2003)]. In the computer vision community is also called *Structure From Motion* [Faugeras (1993)]. Depending on the information and the needed application, motion estimation can be classified in different categories. In the case where the robot moves in a static environment *ego-motion estimation* is used for self-localization and object avoidance, whilst dealing with dynamic environments *independent object motion estimation* have to be performed.

Most of the techniques for motion estimation follow three basic steps. First, distinctive image features are extracted by using adequate descriptors and operators –such as SIFT or Harris detector– and then tracked (or matched) between successive images. This solve the data association problem, however, the data association is never perfect and the outliers (aberrant measures) are usually rejected in a second step using a robust technique like RANSAC for example. By doing this, a set of corresponding points free from mismatches will allow to estimate a tensor containing the camera displacement (e.g. the essential matrix, the trifocal tensor, etc.). Once the camera displacement has been extracted from the tensor, it is possible to reconstruct the structure of the scene up to a scale factor. Some techniques were proposed in the literature by Torr and Zisserman (2000), Davison and Murray (2002) and Se et al. (2005) among many others.

The techniques that reconstruct simultaneously the camera pose and the scene structure can be classified in two main groups: *feature-based* methods and *direct* approaches. In the following sections the advantages and limitations of both methods are briefly discussed. As both techniques have their advantages and drawbacks, in order to take advantage of each one, we developed a novel class of methods, called *hybrid* approach.

5.2 Feature-Based Methods

Feature based methods rely on the reliable extraction and recognition of image features –i.e. geometric primitives such as points, edges, line segments or contours– from sensor data to perform the estimation. In this kind of methods the data association problem is solved first, i.e., find the matching between correspondence pixels, for which, it is separated from the computation of the nonlinear equation system.

For a standard scheme to visual SLAM, the process can be divided in 3 steps as follows:

1. *Data Extraction* Consists on filtering the raw images to detect and extract a sufficiently large set of features (e.g. points, lines, contours...), corresponding to the geometric elements that characterized the observed scene. In the literature there are a variety of feature detectors, most of them specialized in extracting key points as in Harris and Stephens (1988) or edges as in Canny (1986). An ideal detector should be capable to ideally extract the same features in all images, if they are visible. Unfortunately in practice, the ideal detector does not exist, so two measures are commonly adopted to evaluate their performance: accuracy and repeatability. *Accuracy* is achieved thanks to a pre-processing image step like smoothing or image gradients. *Repeatability* is the capability of extract the same features in both images which is extremely difficult to achieve. It generally depends on a threshold to decide if the feature exist or not in the image, and thanks to this, data association can be performed in the next step.

2. *Data Association* Consists on robustly matching the features between successive images (reference and current images), this could be a computationally expensive task. Thus, feature matching algorithms are generally simplified by

evaluating a similarity measure between some descriptors of each feature. Even though, as feature descriptors are not invariant to all the parameters involved, there will be several mismatches at the end of the matching process, which can be eliminated *a posteriori* using geometric constraints within a robust estimation technique, such as RANSAC [Fischler and Bolles (1981)] or M-estimators [Huber (1981)].

3. *Parameter Estimation* Consists on seeking of the parameters that optimally and robustly explain data association, given a model.

In other words, feature based methods minimize an error measure that is based on distances between a few corresponding features. Many successful applications can be found in the literature. See for some examples standard text books like Faugeras (1993) and Hartley and Zisserman (2004).

5.3 Direct Approaches

Direct approaches are also called intensity-based, appearance-based, template-based, or even texture-based in the computer vision community. In these approaches the intensity value of the pixels is directly exploited to recover the related parameters (Irani and Anandan (1999); Stein and Shashua (2000)). Which mean that in contrast with feature-based approaches, there is no feature extraction step. This methods are so popular because they simultaneously solve the data association and the parameter estimation problems, given the parametric model. Therefore, there are fewer assumptions made about the environment and consequently the system will be more portable and robust.

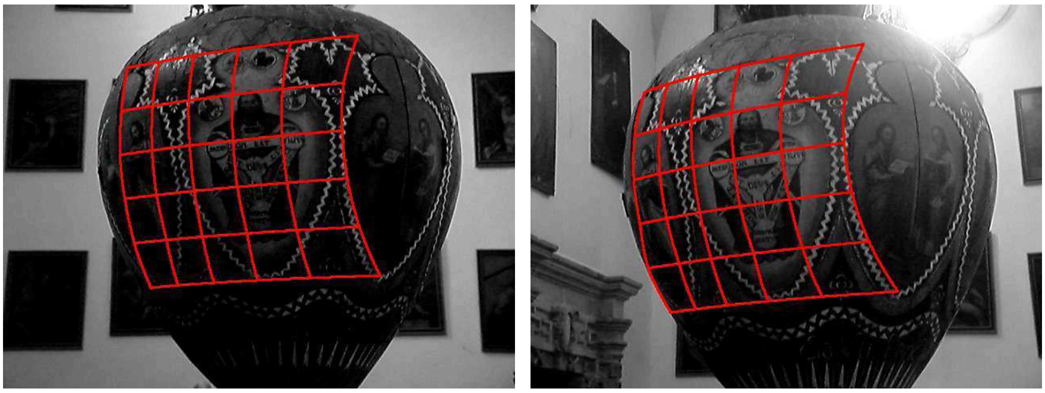
For a standard scheme on visual SLAM, these methods are usually formulated as an optimization problem and the process can be divided in three iterative steps that will be performed until convergence:

1. The transformation of the image, i.e., to *warp* the current image into the reference frame with current parameters. It is important to model an appropriate warping function.

2. Compute the difference between the reference and the warped images. Then, the computation of the similarity measure, i.e. the cost function (e.g. SSD).

3. The computation of an increment of the parameters that decreases the cost function and update the current parameters.

In other words, given an initial estimate of the related parameters, the optimal parameters will be found by obtaining a data association according to a similarity measure. It is important to remark that an initial estimate of the parameters sufficiently close to the true ones is needed, which is clearly one of the main limitations



(a) (b)

Figure 5.1: ESM visual tracking. A template is reliably be tracked even under illumination changes. [source: Malis (2007)]

of this kind of methods. Nevertheless, a dense mapping can be obtained without the need of any post-processing step, since the entire image is exploited, which is one of the major advantages of the direct approaches. Another strength concerns the simultaneous enforcement of structural constraints within the procedure, i.e., *a priori* instead of *a posteriori* as in feature-based methods.

5.3.1 Efficient Second Order Minimization (ESM)

Efficient Second-order Minimization (ESM) technique proposed by Malis (2004) is nowadays one of the most popular algorithms for visual tracking. It is classified as direct method because it does not rely on extracting features and then find correspondences based on certain feature matching criteria. It aims at finding an optimal transformation between two subsequent images [Benhimane and Malis (2004)] or between parts of it, as the visual tracking of rigid and deformable surfaces presented in Malis (2007).

To explain the ESM technique, let us consider the general least-squares minimization problem:

$$F(\mathbf{x}) = \frac{1}{2}\sum_{i=1}^{n}(f_i(\mathbf{x}))^2 = \frac{1}{2}\|\mathbf{f}(\mathbf{x})\|^2 \tag{5.1}$$

The necessary condition to obtain a (local or global) minimum of the cost function is that there exists a stationary point $\widetilde{\mathbf{x}}$ such that the derivative of the cost function is zero, which means:

$$\nabla_{\mathbf{x}} F|_{\mathbf{x}=\widetilde{\mathbf{x}}} = \mathbf{0} \tag{5.2}$$

where $\nabla_{\mathbf{x}}$ is the gradient operator with respect to the parameter $\mathbf{x}$. Generally, it is difficult to obtain a closed-form solution when equation (5.2) is nonlinear. ESM uses a second-order Taylor series of $\mathbf{f}$ about $\mathbf{x} = \mathbf{0}$ which gives:

$$\mathbf{f}(\mathbf{x}) = \mathbf{f}(\mathbf{0}) + \mathbf{J}(\mathbf{0})\,\mathbf{x} + \frac{1}{2}\mathbf{M}(\mathbf{0},\mathbf{x})\mathbf{x} + \boldsymbol{\mathcal{R}}(\|\mathbf{x}\|^3) \tag{5.3}$$

where the last term $\boldsymbol{\mathcal{R}}(\|\mathbf{x}\|^3)$ is a third-order Lagrange remainder and the matrices $\mathbf{J}(\mathbf{0})$ and $\mathbf{M}(\mathbf{z},\mathbf{x})$ are defined as follows:

$$\begin{aligned} \mathbf{J}(\mathbf{0}) &= \nabla_{\mathbf{x}}\mathbf{f}|_{\mathbf{x}=\mathbf{0}} \\ \mathbf{M}(\mathbf{z},\mathbf{x}) &= \nabla_{\mathbf{x}}\mathbf{J}|_{\mathbf{x}=\mathbf{z}}\mathbf{x} \end{aligned}$$

In the same way, the Taylor series of the Jacobian $\mathbf{J}$ about $\mathbf{x} = \mathbf{0}$ can be written as:

$$\mathbf{J}(\mathbf{x}) = \mathbf{J}(\mathbf{0}) + \mathbf{M}(\mathbf{0},\mathbf{x}) + \boldsymbol{\mathcal{R}}(\|\mathbf{x}\|^2) \tag{5.4}$$

Plugging equation (5.4) into equation (5.3) we obtain:

$$\mathbf{f}(\mathbf{x}) = \mathbf{f}(\mathbf{0}) + \frac{1}{2}(\mathbf{J}(\mathbf{0}) + \mathbf{J}(\mathbf{x}))\,\mathbf{x} + \boldsymbol{\mathcal{R}}(\|\mathbf{x}\|^3) \tag{5.5}$$

It is possible to keep the terms of this equation only to second-order as $\widetilde{\mathbf{x}} \approx \mathbf{0}$, thus, a second order approximation of $\mathbf{f}$ in $\widetilde{\mathbf{x}}$ is:

$$\mathbf{f}(\widetilde{\mathbf{x}}) \approx \mathbf{f}(\mathbf{0}) + \frac{1}{2}(\mathbf{J}(\mathbf{0}) + \mathbf{J}(\widetilde{\mathbf{x}}))\,\widetilde{\mathbf{x}} \tag{5.6}$$

This is the basis of the ESM algorithm and under certain conditions $\mathbf{J}(\widetilde{\mathbf{x}})\widetilde{\mathbf{x}}$ can be calculated without knowing the value of $\widetilde{\mathbf{x}}$.

Let $\mathbf{J}(\widetilde{\mathbf{x}})\widetilde{\mathbf{x}} = \mathbf{J}'\widetilde{\mathbf{x}}$, at the solution, $\mathbf{f}(\widetilde{\mathbf{x}}) = 0$, so our second-order least-square minimizer is the solution to:

$$\widetilde{\mathbf{x}} = -\left(\frac{\mathbf{J}(\mathbf{0}) + \mathbf{J}'}{2}\right)^{+}\mathbf{f}(\mathbf{0})$$

This result will be used in the following section with a small difference in order to adapt it to our minimization problem.

5.4 A novel generic robot-centered representation: Augmented Spherical View

In this section, a novel generic robot-centered representation is described that is well adapted to the appearance-based SLAM method. Central omnidirectional cameras can be modeled using two consecutive projections Geyer and Daniilidis (2000): a spherical projection followed by a perspective one. An omnidirectional image can thus be mapped onto a sphere by means of an inverse projection. A point $\boldsymbol{P} \in \mathbb{R}^3$ is projected as a point $\mathbf{q}$ on the unit sphere S^2 and the projection is given by $\mathbf{q} = \frac{\boldsymbol{P}}{\|\boldsymbol{P}\|}$.

The coordinates of $\mathbf{q}$ can be expressed using standard spherical coordinates. The aim of this section is to show how this spherical representation can be constructed.

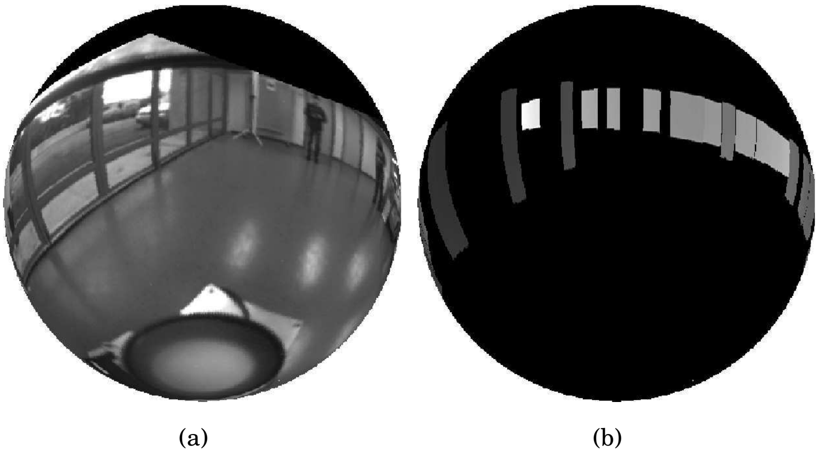


Figure 5.2: Augmented spherical view: at each pixel on the unit sphere is associated with a grey level intensity and the corresponding depth of the 3D point. 5.2(a) Grey levels on the unit sphere. 5.2(b) Depth spherical image.

Since the omnidirectional camera has been calibrated, the intrinsic parameters are known and the omnidirectional image plane $\mathcal{I}_p(u, v)$ can be mapped onto the unit sphere image $\mathcal{I}(\phi, \theta)$. This mapping is performed in four steps:

a) Sampling First, the unit sphere is sampled at a constant angle in a spherical grid defined with the maximal radius of the omnidirectional image in ϕ and the laser points in θ, to respect both laser sampling and omnidirectional sampling.

b) To image plane The spherical points q corresponding to couples (ϕ, θ) are mapped onto the image plane using the "Unified Projection Model" defined in Mei (2007), which is an extension of the models of Geyer and Daniilidis (2000) and Barreto (2003).

c) Interpolation The spherical image is obtained by interpolating the omnidirectional intensity image around the projected points as shown in figure 5.2(a).

d) Hybrid laser/image spherical view Finally, the augmented spherical view is constructed using the depth information from the laser range finder and the floor plane, together with lines extracted from the omnidirectional image. In order to have a more dense estimation in the sphere, the laser trace is propagated down to the floor and upwards (see chapter 4). The resulting spherical image is shown in figure 5.2(b).

In summary, the *current* augmented spherical view is denoted by $\boldsymbol{\mathcal{S}} = \{\boldsymbol{\mathcal{I}}, \boldsymbol{P}\}$ and the *reference* spherical view by $\boldsymbol{\mathcal{S}}^* = \{\boldsymbol{\mathcal{I}}^*, \boldsymbol{P}^*\}$. A superscript $*$ will be used throughout the paper to designate the reference view variables. $\boldsymbol{P}$ is initialized from the laser scan and the vertical 3D lines as shown in figure 5.3.

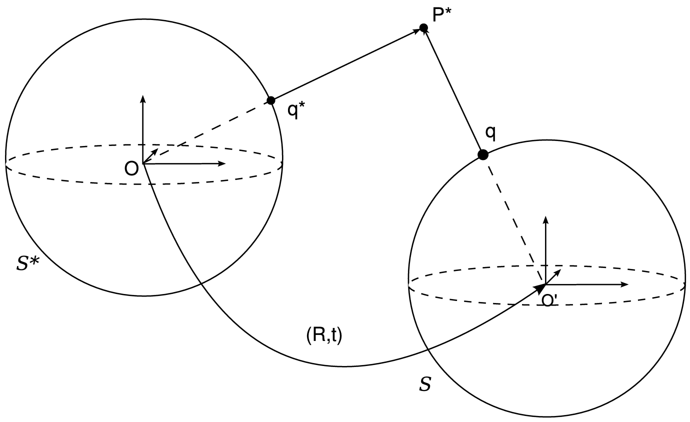


Figure 5.3: Sphere to Sphere mapping

5.5 Efficient hybrid laser/vision appearance-based localization

The main challenge of localization in the context of indoor environments is to obtain reliable odometry robust to illumination changes in the presence of occluding and moving objects. As expected for an odometry-based approach, the objective is to compute the trajectory of the robot (i.e. the laser/vision sensor) along a sequence by integrating elementary displacements estimated from the successive spherical views registered during motion.

An appearance-based localization method is proposed which minimizes a non-linear cost function directly built from the augmented spherical view defined above. As mentioned in the previous section, each pixel q of the spherical view $\mathcal{S}$ is associated with a brightness function $\mathcal{I}(\boldsymbol{P})$ and is augmented with the depth of the associated 3D point (when data is available). In the following, the *reference template* is denoted $\mathcal{R}^* = \{\mathbf{q}_1^*, \ldots, \mathbf{q}_n^*\}$, which defines the subset of the reference spherical view $\mathcal{S}^*$ where both the grey level and the depth values are available, where n is the number of pixels .

5.5.1 Sphere-to-sphere mapping

Consider a 3D point $\boldsymbol{P}^* \in \mathbb{R}^3$ and its projection q* onto the unit sphere (see figure 5.3). Using homogeneous coordinates, the spherical parameterization (θ, ϕ, ρ), gives:

$$\boldsymbol{P}^* = \begin{bmatrix} \rho\cos(\theta)\sin(\phi) \\ \rho\sin(\theta)\sin(\phi) \\ \rho\cos(\phi) \\ 1 \end{bmatrix} \tag{5.7}$$

and

$$\mathbf{q}^* = \frac{\boldsymbol{P}^*}{\|\boldsymbol{P}^*\|} = \begin{bmatrix} \cos(\theta)\sin(\phi) \\ \sin(\theta)\sin(\phi) \\ \cos(\phi) \\ 1 \end{bmatrix}. \tag{5.8}$$

The motion of the sensor or objects within the scene induces a deformation of the reference template. Denote $\overline{\mathbf{T}} = (\overline{\mathbf{R}}, \overline{\mathbf{t}}) \in \mathbb{SE}(3)$ the true current sensor pose relative to the reference sensor pose (homogeneous transformation matrix). The function $w(\boldsymbol{P}^*, \overline{\mathbf{T}})$ which warps the current sphere onto the reference one is defined as

$$\boldsymbol{\mathcal{I}}^*(\boldsymbol{P}^*) = \boldsymbol{\mathcal{I}}\left(w(\boldsymbol{P}^*, \overline{\mathbf{T}})\right), \quad \forall \boldsymbol{P}^* \in \boldsymbol{\mathcal{R}}^*. \tag{5.9}$$

In order to reduce computational time, the warping function is applied to the reference template $\boldsymbol{\mathcal{R}}^*$ only.

The warping function $w(\boldsymbol{P}^*, \overline{\mathbf{T}})$ defines a one-to-one mapping $\mathbf{q}^* \leftarrow \mathbf{q}^{cur}$ from the current sphere to the reference sphere such that

$$\mathbf{q}^{cur} = \frac{\boldsymbol{P}^{cur}}{\|\boldsymbol{P}^{cur}\|} = \frac{\overline{\mathbf{T}}\boldsymbol{P}^*}{\|\overline{\mathbf{T}}\boldsymbol{P}^*\|}. \tag{5.10}$$

The current image $\boldsymbol{\mathcal{I}}$ is then interpolated at points $\mathbf{q}^{cur}$ to obtain the corresponding intensities in spherical coordinates.

Considering that an initial estimation $\widehat{\mathbf{T}}$ of current image pose fully represents the pose of the current camera with respect to a reference sphere, the tracking problem is reduced to estimating the incremental pose $\mathbf{T}(\mathbf{x})$ assuming $\exists \tilde{\mathbf{x}} : \mathbf{T}(\tilde{\mathbf{x}})\widehat{\mathbf{T}} = \overline{\mathbf{T}}$. This estimate is updated by a homogeneous transformation $\widehat{\mathbf{T}} \leftarrow \mathbf{T}(\mathbf{x})\widehat{\mathbf{T}}$. The unknown parameters $\mathbf{x} \in \mathbb{R}^6$ are determined by the integral of a constant velocity twist[1] that produces the pose $\mathbf{T}$ in 6 degrees of freedom:

$$\mathbf{x} = \int_0^1 (\boldsymbol{\omega}, \boldsymbol{\upsilon})dt \in \mathfrak{se}(3). \tag{5.11}$$

The pose and the twist are related via the exponential map[2] by $\mathbf{T} = e^{[\mathbf{x}]_\wedge}$, where the operator $[.]_\wedge$ is defined as

$$[x]_\wedge = \begin{bmatrix} [\boldsymbol{\omega}]_\times & \boldsymbol{\upsilon} \\ 0 & 0 \end{bmatrix} \tag{5.12}$$

and where $[.]_\times$ represents the skew symmetric matrix operator. Hence, the current camera pose can be estimated by minimizing a nonlinear least squares cost function:

$$\boldsymbol{\mathcal{C}}(\mathbf{x}) = \sum_{\boldsymbol{P}^* \in \boldsymbol{\mathcal{R}}^*} \left(\boldsymbol{\mathcal{I}}\left(w\left(\boldsymbol{P}^*, \mathbf{T}(\mathbf{x})\widehat{\mathbf{T}}\right)\right) - \boldsymbol{\mathcal{I}}^*\left(\boldsymbol{P}^*\right)\right)^2. \tag{5.13}$$

[1] see Appendix C, section C.2.1 for a description.

[2] see Appendix C, section C.2.2 for a description.

5.5.2 Minimization of the cost function

The aim now is to minimize the difference in image intensity from the cost function (5.13) in an accurate and robust manner. Since this is a nonlinear function of unknown parameters, an iterative minimization procedure is used. The minimization technique is quite similar to that used in Comport et al. (2007), so only the broad lines of the method are outlined here. Rather than using a standard sum-of-squared differences (SSD) technique based on an L_2 norm, a robust M-estimator[3] (Huber (1981)) is used in order to reject the outliers due to illumination changes, moving objects or occlusions in the scene. The objective function therefore becomes:

$$\mathcal{O}(\mathbf{x}) = \rho\left(\sum_{\boldsymbol{P}^* \in \boldsymbol{\mathcal{R}}^*} \boldsymbol{\mathcal{I}}\left(w\left(\boldsymbol{P}^*, \mathbf{T}(\mathbf{x})\widehat{\mathbf{T}}\right)\right) - \boldsymbol{\mathcal{I}}^*(\boldsymbol{P}^*)\right), \tag{5.14}$$

where $\rho(u)$ is a robust weighting function (see Comport et al. (2007); Huber (1981) for more details).

The robust objective function is minimized by $\nabla\mathcal{O}(\mathbf{x})|_{\mathbf{x}=\tilde{\mathbf{x}}} = \mathbf{0}$, where ∇ is the gradient operator with respect to the unknown parameters of $\mathbf{x}$ from equation (5.11) and it is assumed that there exists a stationary point $\mathbf{x} = \tilde{\mathbf{x}}$ which is the global minimum within the convergence domain (which needs an initialization close to the solution).

The Jacobian of the objective function (5.14) can be decomposed in three parts:

$$\mathbf{J}(\mathbf{x})|_{\mathbf{x}=\tilde{\mathbf{x}}} = \mathbf{J}_{\boldsymbol{\mathcal{I}}^*}\mathbf{J}_w\mathbf{J}_{\mathbf{T}}. \tag{5.15}$$

Here $\mathbf{J}_{\boldsymbol{\mathcal{I}}^*}$ is the image gradient computed on the reference sphere with respect to spherical coordinates (θ, ϕ) of dimension $n \times 2n$, $\mathbf{J}_w$ is the derivative of spherical projection in (5.8) of dimension $2n \times 3n$, and $\mathbf{J}_{\mathbf{T}}$ depends on parametrization of $\mathbf{x}$ from (5.11) and has dimension $3n \times 6$. The objective function (5.14) is iteratively minimized by computing $\widehat{\mathbf{T}} \leftarrow \mathbf{T}(\mathbf{x})\widehat{\mathbf{T}}$ with the vector of unknown parameters $\mathbf{x}$ such that:

$$\mathbf{x} = -\lambda(\mathbf{DJ})^{+}\mathbf{D}(\boldsymbol{\mathcal{I}} - \boldsymbol{\mathcal{I}}^*), \tag{5.16}$$

where $(\mathbf{DJ})^{+}$ is the pseudo-inverse, $\mathbf{D}$ the diagonal matrix determinate from the robust function $\rho(u)$, and λ is a gain factor that ensures the exponential decay of the error.

5.5.3 Initialization step

It is a well known fact that direct iterative methods suffer from convergence problems when initialized far from the solution. This is also true for our method where an initialization sufficiently close to the solution is needed to ensure rapid convergence and reduce computational cost. This initial guess is obtained from the laser data using the 2D scan matching technique developed in chapter 3, which is accurate enough to ensure fast convergence for the appearance-based method.

[3] see Appendix D, for an overview about the calculation.

5.6 Implementation Results

The method is validated using a sequence of 3,262 images and laser scans which were obtained by manually driving the robot in an indoor environment. The exploration trajectory constitutes a closed loop of about 40 meters across the robotic hall.

Figure 5.4(a) shows the map and the pose estimated by scan matching (in green) and the original odometry given by the robot encoders (in red). The shift that can be observed in the 2D map at the end of the loop is caused by *erroneous* laser measurements resulting in the failure of the scan matching process. Even in the presence of these errors it can be seen that the spherical tracking succeeds. This can be seen in figure 5.4(b) in blue. In this case the shift is corrected due to the complementary between the laser and vision data leading to overall robustness and accuracy.

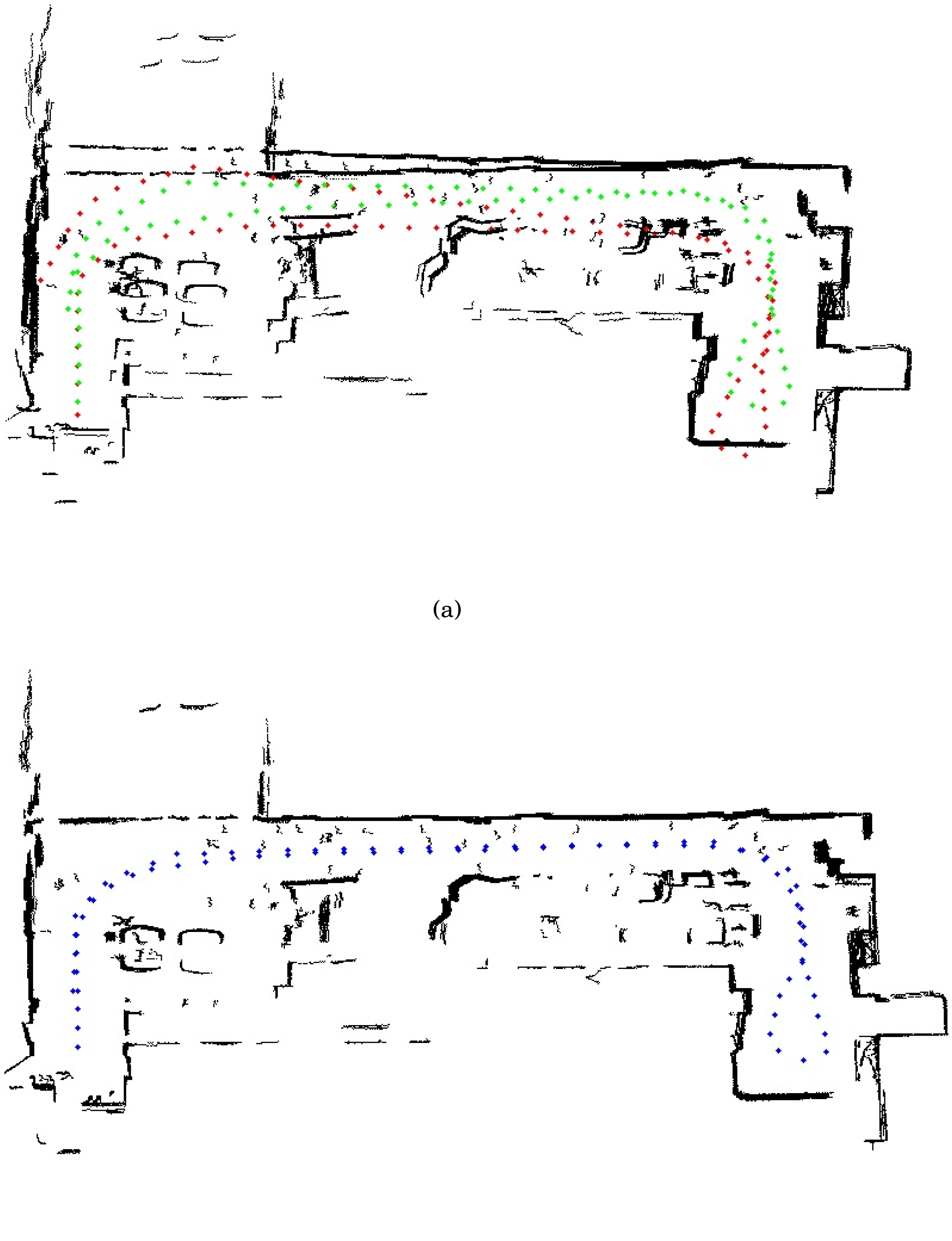
A representation of the images used for the pose estimation with the spherical tracking method is shown in figure 5.5. Observe visually that the final current warped image 5.5(b) (i.e. after convergence) is correctly matched with respect to the reference one in figure 5.5(a). Notice also that in figure 5.5(c), the moving pedestrian is rejected by the robust estimator function because his position in the current warped image differs too much from the reference one. In addition, the algorithm is capable of rejecting specular reflections on the ground and in the windows.

Under the assumption made in chapter 3 that the walls are vertical, from figure 5.5(d) it is clear that the error is negligible for walls, while non vertical textured objects were not matched completely and were correctly rejected. As an example, notice the slanted calibration checkerboard on the left bottom of the images that is perfectly rejected in figure 5.5(e) and 5.5(f). Some parts of the static pedestrian on the left are partially matched because these parts are untextured and do not generate any matching errors, therefore estimation is not affected.

Figure 5.6 shows the 3D textured reconstruction with the correctly matched 3D points obtained from the spherical tracking algorithm. Only the points that were not rejected by the robust estimation were used (i.e. with weight equal to 1). This leads to the rejection of moving pedestrians, that were plotted on the 2D maps in figure 5.4 as well as non-planar/vertical textured objects. The resulting 3D model was rendered in OpenGL and allows walk-through as well as bird eye views. An available video[1], will better illustrate the incremental generation of a 2D map with both estimations of the robot trajectory, as well as the representations of the images used for the pose estimation with the spherical tracking algorithm. A walk-through the 3D reconstructed model is shown at the end of the video.

In summary, the fusion of information from laser scan matching and an appearance-based method improves the robustness of localization and mapping. The robot trajectory is correctly estimated and the drift is minimized. The obtained 3D textured map represents the environment with a good level of precision as seen through the

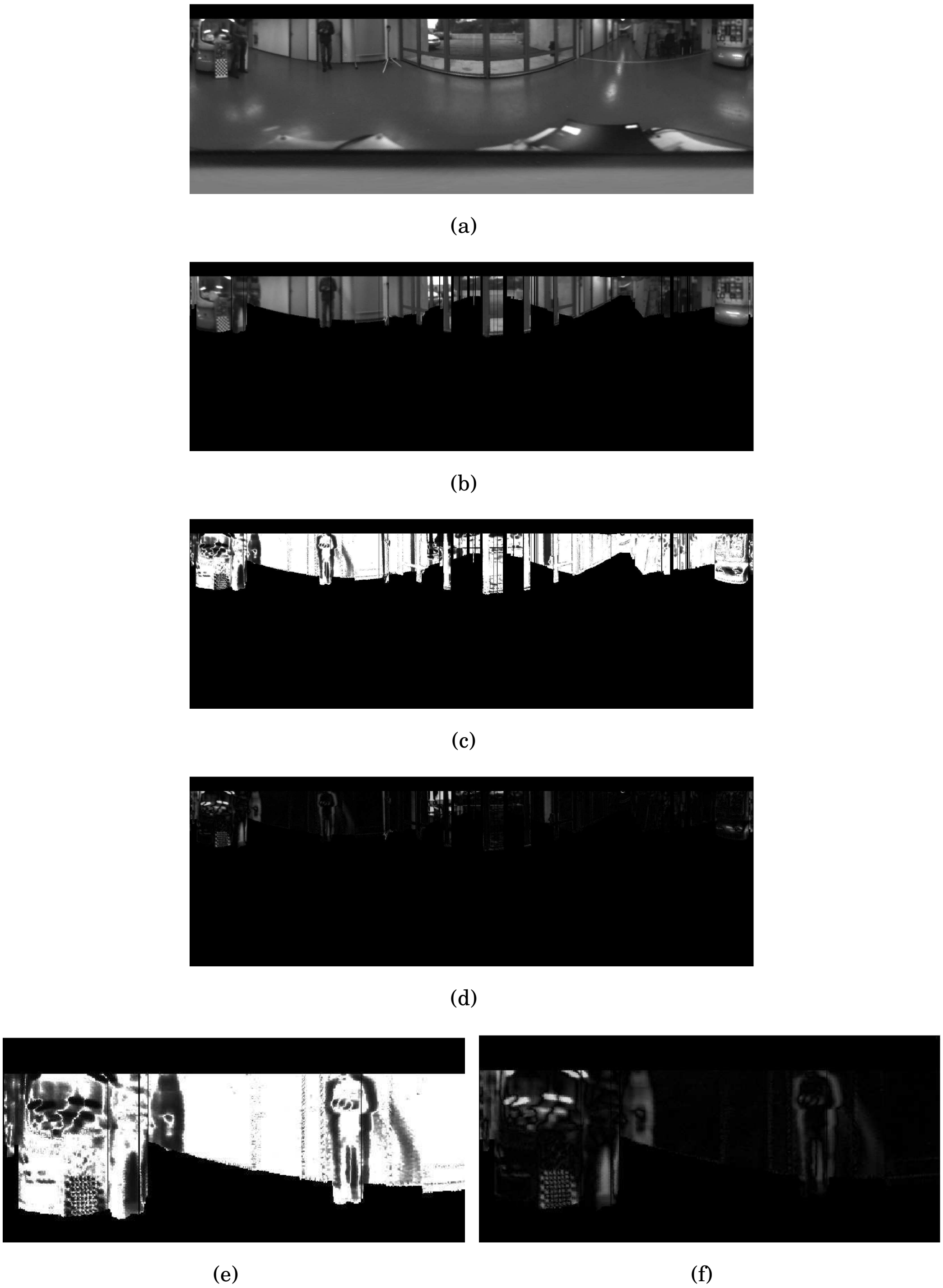
[1] https://www-sop.inria.fr/arobas/videos/HybridLaserOmni_IROS10.mp4



(a)

(b)

Figure 5.4: 2D global maps obtained with the same laser data. 5.4(a) Map with EPSM pose estimation. 5.4(b) Map with spherical pose estimation.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 5.5: Images used for pose estimation. 5.5(a) Reference spherical image. 5.5(b) Current warped image. 5.5(c) Estimated rejection weights. 5.5(d) Final error. 5.5(e) Weights Zoom. 5.5(f) Error zoom.

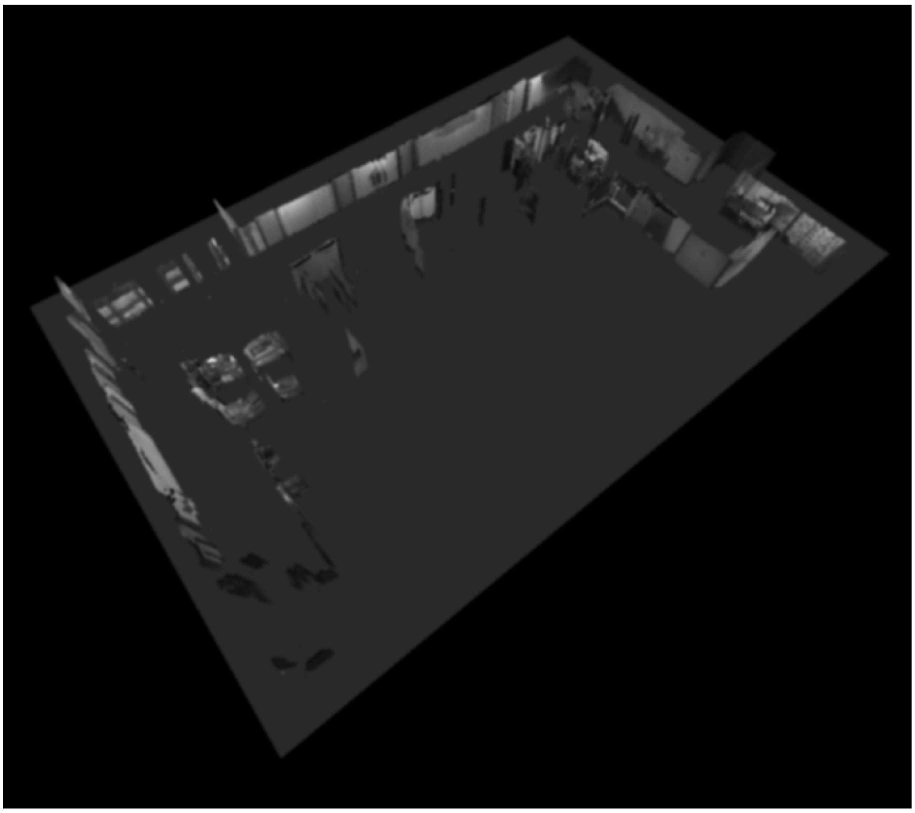


Figure 5.6: 3D reconstruction.

visually satisfying 3D model.

5.7 Conclusion

Although the SLAM problem has been solved using many different approaches, some important problems need to be addressed that are often directly linked to the sensors used. Laser range finders cannot always help in evaluating the translation of a robot moving in a straight line in a corridor leading to potential observability problems. Mapping in dynamic environments is also hard using laser data only due to 2D measurements and slow acquisition rate. On the other hand, using exclusively visual sensors introduces issues such as propagating correctly the scale factor.

The hybrid laser/vision appearance-based approach presented in this chapter has proved to be very efficient in obtaining reliable 3D odometry subject to illumination changes and in the presence of occluding and moving objects. A complete set of 3D points can be easily mapped to reconstruct a dense and consistent representation of the environment. As expected, the initialization of the tracking algorithm close to the solution using scan matching ensures fast exponential decrease of the error and avoids local minima.

It is important to remark, that in this approach, the data is obtained by two full

360° field of view sensors: laser range finder and an omnidirectional camera. The experimental results are encouraging and provide a valuable insight into the possibilities offered by this hybrid approach.

"One never notices what has been done; one can only see what remains to be done."

Marie Curie

6 Conclusions and Perspectives

This chapter summarizes the contributions of this work and proposes potential research avenues that we have identified through our investigations.

6.1 Conclusions

We have presented the overall picture of the Simultaneous Localization and Mapping problem throughout this thesis work. Contemporaneous research in this field has mostly been focused in novel estimation and filtering methods of the SLAM problem to map large-scale indoor or outdoor environments. Alas, the complexity of the environments subject to mapping has been limited by the sensors used. Our research contributes to this field by proposing a novel *tightly coupled composite laser/vision sensor* for indoor SLAM.

This composite sensor takes advantage of the native polar form of laser range finder measurements and the raw vision data from omnidirectional cameras fusing them within a tightly integration scheme. Vision sensors are a relatively cheap way to obtain rich 3D information on the environment, but lack the information about the depth that precise range-bearing measurements from rangefinders can provide. Combined, they can provide efficient and robust (3DOF or 6DOF) motion estimation for mobile robots.

The first part of our research, was devoted to make an large bibliography search for the state of the art on the SLAM problem. This gave us the insights of the scheme to follow to achieve our final objective. Then, we were interested in developing the most of the information provided by the laser rangefinder. This is why, a chapter was dedicated to propose a 2D laser-based SLAM. To achieve this, we propose the Enhanced Polar Scan Matching algorithm, which is a generalization of the original Polar Scan Matching technique proposed by Diosi. The Enhanced Polar Scan Matching algorithm, as any other scan matching algorithm, uses the odometry provided by the wheel encoders of the mobile robot to obtain an initial estimate for the pose of the robot.

As for the vision sensors, we started by analyzing the projection models associated to omnidirectional sensors. We showed how to parameterize omnidirectional lines and proposed an efficient methodology to extract 3D vertical lines from omnidirectional images and laser scans. Furthermore, we showed that, under the planarity assumption, the laser scan can be shifted along the vertical lines to predict where a virtual laser trace –corresponding to the floor– should project in the omnidirectional image. Due to calibration errors that are always present in practice, the predicted trace does not exactly match the real boundary of the floor. Thus, we also propose a technique to correct the segmentation of the floor, where the neighborhood of the predicted trace is searched for the closest element of contour detected in the image. Afterwards, the segmented floor will be the intersection between the floor plane and the walls and will be integrated into a partial 3D model. This completed our *tightly coupled laser/omnidirectional sensor* that enhance both, localization and map representation of the robot's environment.

The last part of this research was dedicated to present a background on visual SLAM and to propose a novel and efficient laser/omnidirectional appearance-based SLAM relying on our tightly coupled sensor described above. This technique, is based on a novel generic robot-centered representation that is well adapted to the appearance-based SLAM method and will provide the mobile robot with rich 3D information about the environment. Furthermore, reliable 3D positioning and a simple but accurate representation of the environment is obtained robust to illumination changes even in the presence of occluding and moving objects. The Enhanced Polar Scan Matching technique proposed above, is used to initialize the tracking algorithm in order to ensure rapid convergence and reduce computational cost.

6.2 Perspectives

In perspective, we have considered several research directions that could be pursued to improve the results obtained so far.

In this thesis work, we are not using any algorithm to close the loop and we did not explore the problem of identifying previously observed places. Extending our algorithm with loop closure detection will allow to detect previously visited locations an will improve the accuracy mapping and the precision in the estimation of the pose of the robot. Being able to detect previously visited places is of great importance to solve the problem of global localization and to recover the robot from kidnapping, a situation occurring when the robot is displaced by something out of its control (e.g., taking an elevator or being transported from one location to another). Therefore, solving the loop closure problem will not only improve the SLAM performance, but will as well enable new capabilities to the robot.

Likewise, a learning step can be added to the image registration step, in which some key images will be saved, in order to allow a better and safety navigation of the robot if it returns to previously observed scenes. It will also allow more accuracy in the building the map process.

Calibration was not a priority when we decided to carry out this thesis work. However, it is an important step if we want to achieve robust results. Thus, being a little more ambitious, it is well known fact that the projective properties of panoramic sensors are strongly related to the intrinsic parameters. Thus, visual tracking could also provide a very suitable way of calibrating the sensor, and should be considered for future work in order to build a fully autonomous robotic system.

As for the appearance-based SLAM algorithm, we have considered to improve it by including the extension of the formalism to deal with non-planar scenes. In this case, the problem will be to formalize the optimization problem so that the estimation can be decoupled into two separate minimization steps:

1. pose estimation,
2. depth refinement

Another direction will be to fuse the initialization, through EPSM scan matching, into the non-linear estimation scheme. Furthermore, the dense map could also be improved by fusing the set of the detected 3D points into a global dense map, for example by tracking planes instead of points. This will give a more accurate and precise view of the environment.

The experiments for this thesis work were entirely perform in indoor environments, i.e., we did not work in dynamic environments. However, a laser/vision sensor could without any doubt help in re-identifying previously observed dynamic features. One possible way to do this is using semantic perception, which is an strategy for graphically representing concepts and which remains an active research field for mobile robots. The performance of mobile robots and the quality of the map can be significantly improved by incorporating semantic information. Related work have studied the use of 2D or 3D laser alone for semantic mapping and very few use images. Future plans include using our hybrid sensor to recognize, identify and classify objects in order to build a semantic and more accurate 3D representation of dynamic environments (ex. using Point Cloud Library (PCL)).

Last but not least, one interesting application for our work could be to use our 3D tracking algorithm and the retrieved pose for *Augmented Reality* (AR). Augmented reality offers a vision of the real world that is enhanced by superimposing information computed in the virtual world. Therefore, AR supplements reality, rather than completely replacing it as virtual reality does (i.e. completely synthetic data). Azuma (1997) explains that a defining notion for AR, is the requirement of being interactive. This mean that is necessary to have both a 3D relationship of virtual objects, as seen by the user, an the incrustation of these objects in real-time into the real world.

The concept of AR depends on the application for which it will be used, having each one specific requirements. In the medical domain applications include visualization of 3D ultrasound images, MRI and CT scan visualization [Soler et al. (2004)]. For military purposes such as the battlefield augmented reality software for urban environments proposed by Julier et al. (1999). A detailed survey on AR applications can be found in Azuma (1997).

As AR is related to the alignment of virtual objects in the scene with real objects in such a way that they are visually acceptable. This requires recovery of the pose between the environment and the camera. Future work will also focus on combining AR with our 3D pose estimation tracker.

Appendix

A
The Kalman filter

A.1 Discrete Kalman Filter (KF)

The discrete Kalman filter addresses the problem of trying to estimate the state x of a discrete time-controlled process determined by the following equations:

$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{B}\mathbf{u}_t + \mathbf{w}_{t-1}$$

with a measurement **z** with Gaussian noise **v** that can be written:

$$\mathbf{z}_t = \mathbf{H}\mathbf{x}_t + \mathbf{v}_t$$

- $\mathbf{w}_t$ represents the process noise, $\mathbf{w} \sim N(0, \mathbf{Q})$.
- $\mathbf{v}_t$ represents the measurement noise, $\mathbf{v} \sim N(0, \mathbf{R})$.

Prediction and time update equations

The state at the time of the next measurement can be predicted:

$$\mathbf{x}_{t+1/t} = \mathbf{A}\mathbf{x}_{t/t} + \mathbf{B}\mathbf{u}_t$$

The state prediction covariance:

$$\mathbf{P}_{t+1/t} = \mathbf{A}\mathbf{P}_{t/t}\mathbf{A}^\top + \mathbf{Q}_t$$

Measurement update equations

The innovation weighted by the filter gain, plus the predicted state, form the updated state estimate:

$$\mathbf{x}_{t+1/t+1} = \mathbf{x}_{t+1/t} + \mathbf{W}_{t+1}\mathbf{v}_{t+1}$$

The updated state covariance is:

$$\mathbf{P}_{t+1/t+1} = \mathbf{P}_{t+1/t} - \mathbf{W}_{t+1}\mathbf{S}_{t+1}\mathbf{W}_{t+1}^{\top}$$

where:

- the *innovation* $\mathbf{v}$ is defined as the difference between the predicted and actual next measurement:

  $$\mathbf{v}_{t+1} = \mathbf{z}_{t+1} - \mathbf{H}\mathbf{x}_{t+1/t}$$

- the Kalman gain $\mathbf{W}$ is defined as:

  $$\mathbf{W}_{t+1} = \mathbf{P}_{t+1/t}\mathbf{H}_{t+1/t}^{\top}S_{t+1}^{-1}$$

- the innovation covariance $\mathbf{S}$ is defined by:

  $$\mathbf{S}_{t+1} = \mathbf{H}_{t+1/t}\mathbf{P}_{t+1/t}\mathbf{H}_{t+1/t}^{\top} + \mathbf{R}_{t+1}$$

  with $\mathbf{R}$ as the measurement noise covariance.

A.2 Extended Kalman Filter (EKF)

The state transition and measurement equations are often nonlinear. The Extended Kalman Filter (EKF) is an extension of the Kalman filter to cope with these nonlinearities. The related mathematical simplifications come however at a price: the distributions are not correctly modeled and the linearization will lead to inconsistencies. In practice however, the results obtained are often satisfactory.

Prediction and time update equations

$$\mathbf{x}_{t+1/t} = f(\mathbf{x}_{t/t}, \mathbf{u}_t)$$
$$\mathbf{P}_{t+1/t} = (\nabla_{\mathbf{x}} f)_{t/t}\mathbf{P}_{t/t}(\nabla_{\mathbf{x}} f)_{t/t}^{\top} + \mathbf{Q}_t$$

where:

- f is the state update equation.
- $\mathbf{x}_{t/t}$ is the state estimate at time t based on the information at time t.
- $\mathbf{x}_{t+1/t}$ is the state estimate at time $t+1$ based on the time update model (i.e. without integrating the measurement information).
- $\mathbf{P}$ correspond to the covariance matrices.
- $\mathbf{Q}$ is the process noise covariance.

Measurement update equations

$$\mathbf{x}_{t+1/t+1} = \mathbf{x}_{t+1/t} + \mathbf{W}_{t+1}\mathbf{v}_{t+1}$$

$$\mathbf{P}_{t+1/t+1} = \mathbf{P}_{t+1/t} - \mathbf{W}_{t+1}\mathbf{S}_{t+1}\mathbf{W}_{t+1}^{\top}$$

The measurement update equations add the information from the new measurements to correct the estimate from the model. $\mathbf{v}$ is called the *innovation* and corresponds to the amount of *unpredicted* information obtained from the new measurement. $\mathbf{W}$ is the Kalman gain and expresses how much trust we can have in the measurement.

$$\mathbf{v}_{t+1} = \mathbf{z}_{t+1} - h(\mathbf{x}_{t+1/t})$$

$$\mathbf{W}_{t+1} = \mathbf{P}_{t+1/t}(\nabla_{\mathbf{x}}h)_{t+1/t}^{\top}\mathbf{S}_{t+1}^{-1}$$

$$\mathbf{S}_{t+1} = (\nabla_{\mathbf{x}}h)_{t+1/t}\mathbf{P}_{t+1/t}(\nabla_{\mathbf{x}}h)_{t+1/t}^{\top} + \mathbf{R}_{t+1}$$

$\mathbf{R}$ is the measurement noise covariance.

B

Fundamentals of Probability Theory

B.1 Probability Theory

B.1.1 Product Rule

The following equation is called the product rule

$$\begin{aligned} P(x,y) &= P(x|y)P(y) \\ &= P(y|x)P(x) \end{aligned}$$

B.1.2 Independence

If x and y are independent, we have

$$P(x,y) = P(x)P(y)$$

B.1.3 Bayes' Rule

The Bayes' rule is given by

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)}$$

The denominator is a normalizing constant that ensures that the posterior of the left hand side adds up to 1 over all possible values. Thus, we often write

$$P(x|y) = \eta P(y|x)P(x)$$

In case the background knowledge e is given, Bayes' rule turns into

$$P(x|y,e) = \frac{P(y|x,e)P(x|e)}{P(y|e)}$$

B.1.4 Marginalization

The marginalization rule is the following equation

$$P(x) = \int_y P(x,y)dy$$

In the discrete case, the integral turns into a sum

$$P(x) = \sum_y P(x,y)$$

B.1.5 Law of Total Probability

The law of total probability is a variant of the marginalization rule, which can be derived using the product rule

$$P(x) = \int_y P(x|y)P(y)dy$$

and the corresponding sum for the discrete case

$$P(x) = \sum_y P(x|y)P(y)$$

B.1.6 Markov Assumption

The Markov assumption (also called Markov property) characterizes the fact that a variable x_t depends only on its direct predecessor state $x_t - 1$ and not on $x_{t'}$ with $t' < t - 1$

$$P(x_t|x_{1:t-1}) = P(x_t|x_{t-1})$$

B.2 Probability Density Functions

The probabilistic approach to estimation is based on the concept of the *probability density function* (PDF) which is used to define the uncertainty distribution of a set of random variables. In other words, a PDF $P(\mathrm{x})$ expresses, for a particular random vector x, the likelihood that the true state of x lies within a particular region of the state-space $\mathbf{X}$. For the purposes of this thesis it is sufficient to understand the following PDF properties.

- A PDF $P(\mathbf{x})$ represents a functional mapping $P : \mathbf{x} \rightarrow \mathbb{R}$ for all $\mathbf{x} \in \mathbf{X}$.
- A PDF $P(\mathbf{x})$ is non-negative for all values of random vector $\mathbf{x}$,

$$P(\mathbf{x}) \geq 0, \ \forall \mathbf{x} \in \mathbf{X}$$

- The area (or volume) under a PDF is one,

$$\int_{-\infty}^{\infty} P(\mathbf{x}) d\mathbf{x} = 1$$

If there exist two random vectors $\mathbf{x}$ and $\mathbf{y}$ where the value of $\mathbf{x}$ is to some degree dependent on the value of $\mathbf{y}$, then there exists a *conditional* PDF $P(\mathbf{x}|\mathbf{y})$. The conditional PDF $P(\mathbf{x}|\mathbf{y})$ may be understood as the probability or likelihood of $\mathbf{x}$ given a fixed value of $\mathbf{y}$. If $\mathbf{x}$ and $\mathbf{y}$ are independent then $P(\mathbf{x}|\mathbf{y}) = P(\mathbf{x})$.

C
Fundamentals of 3D motion

Even though the concepts presented in this appendix, are well founded concepts, we present them here for sake of completeness and to state the basic terminology used throughout the thesis.

The representation of the pose (position and orientation) and the structure of rigid objects and their kinematics forms the basis for the study of Visual SLAM. The terminology and notation required to represent coordinate transformations and the velocity of a rigid object moving through the three-dimensional world (3D euclidean space), will be defined first.

C.1 Rigid body Motion: Definitions

We consider the 3D space as being Cartesian, i.e., a three-dimensional Euclidean space $\mathbb{E}^3$. The position of a point $\mathbf{P} \in \mathbb{E}^3$ of the rigid body, is defined to be relative to an inertial cartesian coordinate frame. Thus, with the set of three orthonormal axes that represent the reference frame, the position of a 3D point can be written using three coordinates as:

$$\mathbf{P} = (X, Y, Z) \in \mathbb{R}^3 \tag{C.1}$$

Similarly, the motion trajectory of a point can be represented as a parameterized curve as:

$$\mathbf{P}(t) = (X(t), Y(t), Z(t)) \in \mathbb{R}^3 \tag{C.2}$$

A rigid body transformation is defined as follows:

Definition 1 *Special Euclidean Transformation*

A mapping $\mathbf{m} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ *satisfies the following properties:*

1. *Lenght is preserved:* $\|\mathbf{m}(\mathbf{P}_1) - \mathbf{m}(\mathbf{P}_2)\| = \|\mathbf{P}_1 - \mathbf{P}_2\|$ *for all points* $\mathbf{P}_1, \mathbf{P}_2 \in \mathbb{R}^3$
2. *The cross product is preserved:* $\mathbf{m}_*(\mathbf{v} \times \mathbf{w}) = \mathbf{m}_*(\mathbf{v}) \times \mathbf{m}_*(\mathbf{w})$ *for all vectors* $\mathbf{v}, \mathbf{w} \in \mathbb{R}^3$*, where vectors transform according to* $\mathbf{m}_*(\mathbf{v}) = \mathbf{m}(\mathbf{P}_1) - \mathbf{m}(\mathbf{P}_2)$

The set of all such transformations is denoted as the special Euclidean group $\mathbb{SE}(3)$

C.1.1 Representing Rotations

If the coordinate frame is chosen to be right handed, then $det(\mathbf{R}) = +1$, and the properties of a rotation matrix are such that its columns are mutually orthonormal, i.e., $\mathbf{R}\mathbf{R}^\top = \mathbf{R}^\top\mathbf{R} = \mathbf{I}$. The special orthogonal subgroup of dimension 3, also called rotation group, is defined as:

$$\mathbb{SO}(3) = \{\mathbf{R} \in \mathbb{R}^{3\times 3} \mid \mathbf{R}\mathbf{R}^\top = \mathbf{I}, \det(\mathbf{R}) = +1\}$$

If we consider rotations parameterized by α, β, γ around the Euler angles, each elementary rotation is defined by the following matrices:

$$\mathbf{R}_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) \\ 0 & -\sin(\alpha) & \cos(\alpha) \end{bmatrix}, \ \mathbf{R}_y(\beta) = \begin{bmatrix} \cos(\beta) & 0 & -\sin(\beta) \\ 0 & 1 & 0 \\ \sin(\beta) & 0 & \cos(\beta) \end{bmatrix}, \ \mathbf{R}_z(\gamma) = \begin{bmatrix} \cos(\gamma) & \sin(\gamma) & 0 \\ -\sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and the matrix describing the overall rotation being composed of the product of these elementary matrices $\mathbf{R} = \mathbf{R}_x\mathbf{R}_y\mathbf{R}_z$:

$$\mathbf{R} = \begin{bmatrix} \cos(\beta)\cos(\gamma) & -\cos(\beta)\sin(\gamma) & \sin(\beta) \\ \sin(\alpha)\sin(\beta)\cos(\gamma) + \cos(\alpha)\sin(\gamma) & -\sin(\alpha)\sin(\beta)\sin(\gamma) + \cos(\alpha)\cos(\gamma) & -\sin(\alpha)\cos(\beta) \\ -\cos(\alpha)\sin(\beta)\cos(\gamma) + \sin(\alpha)\sin(\gamma) & \cos(\alpha)\sin(\beta)\sin(\gamma) + \sin(\alpha)\cos(\gamma) & \cos(\alpha)\cos(\beta) \end{bmatrix}$$

Remark: *The order of multiplication of elementary rotation matrices is not commutative* ■

C.1.2 Representing Pose and Structure

A rigid transformation $\mathbf{m} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ is composed of rotational and translational motions.

The minimal representation of a rigid transformation is parameterized as a six parameter pose vector, composed of translation and rotation parameters as:

$$ {}^a\mathbf{r}_b = ({}^a\mathbf{t}_b, {}^a\,\boldsymbol{\Omega}_b) $$

where r represents the pose (position and orientation) between a reference frame a and another frame b, been ${}^a\mathbf{t}_b = ({}^a t_{b,x}, {}^a t_{b,y}, {}^a t_{b,z})$ the vector of translation parameters between frames a and b along the axes x, y, z and ${}^a\boldsymbol{\Omega}_b = ({}^a\Omega^a_{b,x,}\Omega_{b,y}, {}^a\Omega_{b,z})$ the rotational parameters between frame a and b around the axes x, y, z.

The motion of a rigid body can be represented with elements of the special euclidean group $\mathbb{SE}(3)$:

$$ \mathbb{SE}(3) = \{\mathbf{m} = (\mathbf{R}, \mathbf{t}) \mid \mathbf{R} \in \mathbb{SO}(3), \mathbf{t} \in \mathbb{R}^3\} \tag{C.3} $$

The homogeneous representation of $\mathbf{m}$ is obtained in matrix form as:

$$ {}^a\mathbf{M}_b = \begin{bmatrix} {}^a\mathbf{R}_b & {}^a\mathbf{t}_b \\ \mathbf{0}_3 & 1 \end{bmatrix} \tag{C.4} $$

This set of homogeneous transformations belongs to the 6-dimensional Lie Group of rigid body motions in $\mathbb{SE}(3)$. In addition, the pose between frame a and frame c can be expressed as a composition of homogeneous transformation matrices as :

$$ {}^a\mathbf{M}_c = {}^a\mathbf{M}_b{}^b\mathbf{M}_c \tag{C.5} $$

where ${}^b\mathbf{M}_a = {}^a\mathbf{M}_b{}^{-1}$ and the inverse transformation is:

$$ {}^a\mathbf{M}_b{}^{-1} = \begin{bmatrix} {}^a\mathbf{R}_b^\top & -{}^a\mathbf{R}_b^\top{}^a\mathbf{t}_b \\ \mathbf{0}_3 & 1 \end{bmatrix} $$

C.2 Velocity of a Rigid Body

A rigid body velocity is defined as a 6-dimensional *twist* vector $\mathbf{v} = (\boldsymbol{\upsilon}, \boldsymbol{\omega})$ where $\boldsymbol{\upsilon} = (\upsilon_x, \upsilon_y, \upsilon_z)$ is the linear component of the velocity vector and $\boldsymbol{\omega} = (\omega_x, \omega_y, \omega_z)$ the angular velocity. A twist vector is the tangent vector to an element $\mathbf{m}(t)$ of $\mathbb{SE}(3)$.

Let us consider a point $\mathbf{P}$ with coordinates in a spatial reference frame a and another reference frame b. The homogeneous relation between a single point in two different reference frames can be written as:

$$ {}^a\bar{\mathbf{P}}(t) ={}^a\,\mathbf{M}_b(t){}^b\bar{\mathbf{P}} \tag{C.6} $$

where the bar on $\bar{\mathbf{P}} = (X, Y, Z, 1)$ is an homogeneous point. The velocity of the point with respect to frame a_*, where $*$ refers to an *instantaneous* reference frame, is obtained by deriving the motion of its coordinates with respect to time:

$$ {}^{a_*}\dot{\mathbf{P}}(t) = \frac{d}{dt}{}^a\mathbf{P}(t) \tag{C.7} $$

By deriving equation C.6 with respect to time, the velocity of the point in frame a_* gives:

$$ {}^{a_*}\dot{\bar{\mathbf{P}}} = {}^{a_*}\dot{\mathbf{M}}_b {}^b\bar{\mathbf{P}} $$

where the homogeneous velocity of a point is ${}^{a_*}\dot{\bar{\mathbf{P}}} = (v_x, v_y, v_z, 1)$. The velocity of the point with respect to the spatial reference frame a can then be related to its coordinates in the *instantaneous* spatial reference frame ${}^{a_*}\mathbf{P}$ by using composition C.5:

$$ {}^{a_*}\dot{\bar{\mathbf{P}}} = {}^{a_*}\dot{\mathbf{M}}_b {}^a\mathbf{M}_b^{-1} {}^a\bar{\mathbf{P}} = {}^{a_*}\dot{\mathbf{M}}_b {}^b\mathbf{M}_a {}^a\bar{\mathbf{P}} \tag{C.8} $$

The 4×4 velocity mapping in equation C.8 is called a twist and is then easily obtained by using equation C.4 as:

$$ {}^{a_*}\dot{\mathbf{M}}_b {}^a\mathbf{M}_b^{-1} = \begin{bmatrix} {}^{a_*}\dot{\mathbf{R}}_b {}^{a_*}\dot{\mathbf{t}}_b \\ \mathbf{0}_3 \quad 1 \end{bmatrix} \begin{bmatrix} {}^a\mathbf{R}_b^\top - {}^a\mathbf{R}_b^\top {}^a\mathbf{t}_b \\ \mathbf{0}_3 \quad\quad 1 \end{bmatrix} = \begin{bmatrix} {}^{a_*}\dot{\mathbf{R}}_b {}^a\mathbf{R}_b^\top - {}^{a_*}\dot{\mathbf{R}}_b {}^a\mathbf{R}_b^\top {}^a\mathbf{t}_b + {}^{a_*}\dot{\mathbf{t}}_b \\ \mathbf{0}_3 \quad\quad\quad\quad\quad\quad 1 \end{bmatrix} \tag{C.9} $$

where ${}^{a_*}\dot{\mathbf{t}}_b$ is the translational velocity of a point in frame a_* with respect to a point in frame b.

C.2.1 Velocity Twist

The 4×4 twist matrix from equation C.9 is shown to be related to a minimal 6 parameter vector by defining the angular component ${}^{a_*}\boldsymbol{\omega}_a \in \mathbb{R}^3$ and the linear component ${}^{a_*}\boldsymbol{\upsilon}_a \in \mathbb{R}^3$ of the velocity of a point in the body frame b passing through instantaneous spatial frame a as:

$$ {}^{a_*}\mathbf{v}_a = \begin{bmatrix} {}^{a_*}\boldsymbol{\upsilon}_a \\ [{}^{a_*}\boldsymbol{\omega}_a]_\times \end{bmatrix} = \begin{bmatrix} -{}^{a_*}\dot{\mathbf{R}}_b {}^a\mathbf{R}_b^\top {}^a\mathbf{t}_b + {}^{a_*}\dot{\mathbf{t}}_b \\ {}^{a_*}\dot{\mathbf{R}}_b {}^a\mathbf{R}_b^\top \end{bmatrix} $$

Therefore, there exists an operator, based on the skew-symmetric operator $[.]_\times$ for rotations, which transforms a full twist matrix to its minimal vector form defined as:

$$ [\mathbf{v}]_\wedge = \begin{bmatrix} [\boldsymbol{\omega}]_\times & \boldsymbol{\upsilon} \\ 0 & 0 \end{bmatrix} \tag{C.10} $$

The space velocity twists can be written as a 4×4 homogeneous twist matrix $[\mathbf{v}]_\wedge$ as:

$$ \mathfrak{se}(3) = \{[\mathbf{v}]_\wedge \in \mathbb{R}^{4\times4} | [\omega]_\times \in \mathfrak{so}(3), \mathbf{v} \in \mathbb{R}^3\} \subset \mathbb{R}^{4\times4} $$

where $\mathfrak{se}(3)$ is the Lie Algebra of the Lie Group $\mathbb{SE}(3)$ and $\mathfrak{so}(3)$ is the Lie Algebra of the Lie Group $\mathbb{SO}(3)$.

C.2.2 Exponential Map

The relationship between the velocity of a moving body and its pose is called *the exponential map*, which transforms, exponentially, the velocity vector v to its corresponding pose r.

If G is a matrix Lie group with Lie algebra $\mathfrak{g}$, then the exponential mapping for G is the map:

$$\exp : \mathfrak{g} \rightarrow G$$

From equation C.3, the generators of the translation $(\mathbf{A}_1, ..., \mathbf{A}_3)$ and rotation $(\mathbf{A}_4, ..., \mathbf{A}_6)$ of the Lie algebra $\mathfrak{se}(3)$ can be obtained by differentiation, thus:

$$\mathbf{A}_1 = \begin{bmatrix} \mathbf{0} & \mathbf{b}_x \\ \mathbf{0} & 0 \end{bmatrix}, \mathbf{A}_2 = \begin{bmatrix} \mathbf{0} & \mathbf{b}_y \\ \mathbf{0} & 0 \end{bmatrix}, \mathbf{A}_3 = \begin{bmatrix} \mathbf{0} & \mathbf{b}_z \\ \mathbf{0} & 0 \end{bmatrix}, \tag{C.11}$$

$$\mathbf{A}_4 = \begin{bmatrix} [\mathbf{b}_x]_\times & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix}, \mathbf{A}_5 = \begin{bmatrix} [\mathbf{b}_y]_\times & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix}, \mathbf{A}_6 = \begin{bmatrix} [\mathbf{b}_z]_\times & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix}$$

Using Rodrigues' formula, it is possible to obtain an explicit formulation of the exponential map.

Theorem 1 *Let* $\mathbf{A}(\mathbf{x}) \in \mathfrak{se}(3)$*, with* $\mathbf{v} = (x_1, x_2, x_3)$ *(translational component) and* $\omega = (x_4, x_5, x_6)$ *(rotational component) :*

$$\begin{cases} e^{\mathbf{A}} = \begin{bmatrix} e^{[\omega]_\times} & \frac{\left((\mathbf{I} - e^{[\omega]_\times})[\omega]_\times + \omega\omega^\top\right)\mathbf{v}}{\|\omega\|^2} \\ \mathbf{0} & 1 \end{bmatrix} & \text{if } \|\omega\| \neq 0 \\ e^{\mathbf{A}} = \begin{bmatrix} \mathbf{I} & \mathbf{v} \\ \mathbf{0} & 1 \end{bmatrix} & \text{otherwise} \end{cases}$$

D
Robust Estimation

As explained in chapter 5 section 5.5.2 a robust M-estimation technique was used to reject outliers not corresponding to the definition of the objective function. Robust techniques are mostly used where a highly redundant set of measurements – as is the case of a set of dense correspondences– are involved. In general, outliers occurred because of illumination changes, occlusions, matching error or simply noise in the image.

In this appendix we give the calculation of weights for each image feature.

D.1 Robust M-Estimator

The weights ω_i which represent the different elements of the D matrix and reflect the confidence of each feature, are usually given by Huber (1981) as:

$$\omega_i = \frac{\psi(\delta_i/\sigma)}{\delta_i/\sigma} \tag{D.1}$$

where

$$\psi(\delta_i/\sigma) = \partial\rho\frac{(\delta_i/\sigma)}{\partial\mathbf{r}} \tag{D.2}$$

where $\psi()$ is the M-estimate, also called *the influence function*, and δ_i is the normalized residual given by:

$$\delta_i = \Delta_i - \mathrm{Med}(\Delta) \tag{D.3}$$

where Med(Δ) corresponds to the median value taken across all the residues and where σ is a scale that corresponds to a robust estimate of the standard deviation of the inlier data. This is a critical value that can impact heavily on the efficiency of the method. For traditional M-estimators, scale is treated as a tuning constant which can be chosen manually. Alternatively, a robust statistic can be used to compute it. Particularly, we used the median absolute deviation (MAD), which is a robust statistic that allows to reject up to 50% of outliers. The MAD is given by:

$$\widehat{\sigma} = \frac{1}{\Phi^{-1}(0.75)}\text{Med}_i(\sigma - \text{Med}_j(\sigma_j)) \tag{D.4}$$

n^{o} indent where $\Phi()$ is the cumulative normal distribution function and $\frac{1}{\Phi^{-1}(0.75)} = 1.48$ represents one standard deviation of the normal distribution, and is used to make the MAD consistent with the normal distribution. It is important to remark that the MAD is a very good tradeoff between outliers rejection efficiency and computation efficiency.

The introduction of the weighting matrix **D** into the minimization scheme (equation 5.16 in section 5.5.2) is achieved via an iteratively re-weighted least-squares implementation.

E
Publications of the Author

Conference papers

1. G. Gallegos and P. Rives. *Indoor SLAM based on Composite Sensor Mixing Laser Scans and Omnidirectional Images.* In 23rd IEEE International Conference on Robotics and Automation, ICRA 2010, Anchorage, USA, May 2010.

2. G. Gallegos, M. Meilland, P. Rives and A.I. Comport. *Appearance-Based SLAM Relying on a Hybrid Laser / Omnidirectional Sensor.* In IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2010, Taipei, Taiwan, October 2010.

Bibliography

Akashi, H. and H. Kumamoto: 1977, 'Random sampling approach to state estimation in switching environments'. *Automatica* **13**(4), 429 – 434. [21]

Arras, K., J. Castellanos, and R. Siegwart: 2002, 'Feature-based multi-hypothesis localization and tracking for mobile robots using geometric constraints'. In: *IEEE International Conference on Robotics and Automation*, Vol. 2. pp. 1371 – 1377 vol.2. [20]

Azuma, R.: 1997, 'A Survey of Augmented Reality'. *Presence: Teleoperators and Virtual Environments* **6**(4), 355–385. [102]

Bailey, T., J. Nieto, J. Guivant, M. Stevens, and E. Nebot: 2006a, 'Consistency of the EKF-SLAM Algorithm'. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp. 3562 –3568. [20, 21]

Bailey, T., J. Nieto, and E. Nebot: 2006b, 'Consistency of the FastSLAM algorithm'. In: *IEEE International Conference on Robotics and Automation*. pp. 424 –429. [20, 22]

Baker, S. and S. K. Nayar: 1998, 'A Theory of Catadioptric Image Formation'. In: *International Conference on Computer Vision (ICCV)*. Washington, DC, USA, p. 35, IEEE Computer Society. [14, 64, 66]

Baker, S. and S. K. Nayar: 1999, 'A Theory of Single-Viewpoint Catadioptric Image Formation'. *International Journal of Computer Vision* **35**(1), 1 – 22. [63]

Barreto, J. P.: 2003, 'General central projection systems, modeling, calibration and visual servoing'. Ph.D. thesis, University of Coimbra, Department of Electrical and Computer Engineering. [64, 67, 74, 89]

Barreto, J. P. and H. Araujo: 2005, 'Geometric properties of central catadioptric line images and their application in calibration'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **27**(8), 1327 –1333. [70]

Benhimane, S. and E. Malis: 2004, 'Real-time image-based tracking of planes using Efficient Second-order Minimization'. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp. 943–948. [14, 87]

Besl, P. J. and N. D. McKay: 1992, 'A Method for Registration of 3-D Shapes'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **14**, 239–256. [13, 25, 46]

Biber, P., H. Andreasson, T. Duckett, and A. Schilling: 2004, '3D Modeling of Indoor Environments by a Mobile Robot with a Laser Scanner and Panoramic Camera'. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp. 3430–3435. [3, 4]

Biber, P. and W. Straßer: 2003, 'The Normal Distribution Transform: A New Approach to Laser Scan Matching'. In: *International Conference on Intelligent Robots and Systems*, Vol. 3. pp. 2743–2748. [46]

Blanc, G., Y. Mezouar, and P. Martinet: 2005, 'Indoor Navigation of a Wheeled Mobile Robot along Visual Routes'. In: *Proceedings of the IEEE International Conference on Robotics and Automation*. pp. 3354 – 3359. [30]

Borenstein, J. and L. Feng: 1996, 'Measurement and Correction of Systematic Odometry Errors in Mobile Robots'. *IEEE Transactions on Robotics and Automation* **12**, 869–880. [33]

Bosse, M., P. Newman, J. Leonard, M. Soika, W. Feiten, and S. Teller: 2003, 'An Atlas framework for scalable mapping'. In: *IEEE International Conference on Robotics and Automation*, Vol. 2. pp. 1899–1906. [20]

Bosse, M. and R. Zlot: 2008, 'Map Matching and Data Association for Large-Scale Two-dimensional Laser Scan-based SLAM'. *International Journal of Robotics Research* **27**(6), 667–691. [21, 46]

Brevi, D., F. Fileppo, R. Scopigno, F. Abrate, B. Bona, S. Rosa, and F. Tibaldi: 2009, 'Hybrid localization solutions for robotic logistic applications'. In: *IEEE International Conference on Technologies for Practical Robot Applications*. pp. 167–172. [41]

Broida, T. J., S. Chandrashekhar, and R. Chellappa: 1990, 'Recursive 3D Motion Estimation from a Monocular Image Sequence'. *IEEE Transactions on Aerospace and Electronic Systems* **26**, 639–656. [13]

Brown, D. C.: 1958, 'A solution to the general problem of multiple station analytical stereotriangulation'. Technical Report RCP-MTP Data Reduction Technical Report No.43, Patrick Air Force Base, Florida. (also designated as AFMTC 58-8). [23]

Brown, D. C.: 1976, 'The bundle adjustment — progress and prospects'. *International Archives of Photogrammetry* **21**(3), 0–33. [23]

Calonder, M.: 2010, 'EKF-SLAM vs FastSLAM: A comparison'. Technical Report CVLAB-REPORT 2010-001, EPFL Lausanne. [22]

Canny, J.: 1986, ‘A Computational Approach to Edge Detection’. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-8**(6), 679 –698. [74, 85]

Carpin, S.: 2008, ‘Fast and accurate map merging for multi-robot systems’. *Autonomous Robots* **25**, 305–316. [28]

Castellanos, J., J. Martinez, J. Neira, and J. Tardos: 1998, ‘Simultaneous map building and localization for mobile robots: a multisensor fusion approach’. In: *IEEE International Conference on Robotics and Automation*, Vol. 2. pp. 1244 –1249. [28]

Charbonnier, L. and O. Strauss: 1995, ‘A suitable polygonal approximation for laser range data’. In: *Intelligent Vehicules Symposium*. Detroit, Mi., pp. 118–123. [13]

Chatila, R. and J. Laumond: 1985, ‘Position referencing and consistent world modeling for mobile robots’. In: *IEEE International Conference on Robotics and Automation*, Vol. 2. pp. 138 – 145. [28, 30]

Chong, K. S. and L. Kleeman: 1997, ‘Sonar based map building for a mobile robot’. In: *IEEE International Conference on Robotics and Automation*, Vol. 2. pp. 1700 –1705. [28]

Chong, K. S. and L. Kleeman: 1999, ‘Mobile Robot Map Building from an Advanced Sonar Array and Accurate Odometry’. *International Journal of Robotics Research* **18**, 20–36. [13]

Choset, H. and K. Nagatani: 2001, ‘Topological simultaneous localization and mapping (SLAM): toward exact localization without explicit localization’. *IEEE Transactions on Robotics and Automation* **17**(2), 125 –137. [30]

Clerentin, A., L. Delahoche, C. Pégard, and E. Brassart: 2000, ‘A Localization Method Based on Two Omnidirectional Perception Systems Cooperation’. In: *IEEE International Conference on Robotics and Automation*. pp. 1219–1224. [3, 14]

Cobzas, D., H. Zhang, and M. Jagersand: 2003, ‘Image-Based Localization with depth-enhanced image map’. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp. 1570–1575. [3, 14]

Comport, A., E. Malis, and P. Rives: 2007, ‘Accurate Quadrifocal Tracking for Robust 3D Visual Odometry’. In: *IEEE International Conference on Robotics and Automation*. Rome, Italy. [14, 92]

Comport, A., E. Malis, and P. Rives: 2010, ‘Real-time Quadrifocal Visual Odometry’. *International Journal of Robotics Research* **29**(2-3), 245–266. [14]

Courbon, J., G. Blanc, Y. Mezouar, and P. Martinet: 2007, ‘Navigation of a nonholonomic mobile robot with a memory of omnidirectional images’. In: *ICRA Workshop on "Planning, perception and navigation for Intelligent Vehicles"*. Rome, Italy. [30]

Cox, I. J.: 1991, 'Blanche-an experiment in guidance and navigation of an autonomous robot vehicle'. *IEEE Transactions on Robotics and Automation* **7**(2), 193 –204. [13]

Davison, A., Y. Gonzalez-Cid, and N. Kita: 2004, 'Real-time 3D SLAM with wide-angle vision'. In: *IFAC Symposium on Intelligent Autonomous Vehicles*. [14]

Davison, A. J.: 2003, 'Real-Time Simultaneous Localisation and Mapping with a Single Camera'. In: *IEEE International Conference on Computer Vision*. Washington, DC, USA, p. 1403, IEEE Computer Society. [13, 27, 84]

Davison, A. J. and D. W. Murray: 2002, 'Simultaneous Localization and Map-Building Using Active Vision'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**, 865–880. [85]

Dempster, A. P., N. M. Laird, and D. B. Rubin: 1977, 'Maximum likelihood from incomplete data via the EM algorithm'. *Journal of the Royal Statistical Society, SERIES B* **39**(1), 1–38. [24]

Diosi, A. and L. Kleeman: 2005, 'Laser Scan Matching in polar coordinates with application to SLAM'. In: *IEEE International Conference on Robotics and Automation*. Edmonton, Canada, pp. 3317–3322. [xiv, 5, 46, 49]

Diosi, A., G. Taylor, and L. Kleeman: 2005, 'Interactive SLAM using Laser and Advanced Sonar'. In: *IEEE International Conference on Robotics and Automation*. pp. 1103 – 1108. [29]

Dissanayake, G., P. Newman, S. Clark, H. F. Durrant-whyte, and M. Csorba: 2001, 'A solution to the simultaneous localization and map building (SLAM) problem'. *IEEE Transactions on Robotics and Automation* **17**, 229–241. [19]

Doucet, A., N. d. Freitas, K. P. Murphy, and S. J. Russell: 2000, 'Rao-Blackwellised Particle Filtering for Dynamic Bayesian Networks'. In: *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*. San Francisco, CA, USA, pp. 176–183. [22]

Duda, R. O. and P. E. Hart: 1972, 'Use of the Hough transformation to detect lines and curves in pictures'. *Communications of the ACM* **15**, 11–15. [76]

Dudek, G. and M. Jenkin: 2000, *Computational principles of mobile robotics*. Cambridge University Press. [34, 50]

Durrant-Whyte, H. and T. Bailey: 2006, 'Simultaneous localization and mapping: part I'. *IEEE Robotics Automation Magazine* **13**(2), 99 –110. [15]

Elfes, A.: 1987, 'Sonar-Based Real-World Mapping and Navigation'. *IEEE Journal of Robotics and Automation* **3**, 249–265. [13]

Elfes, A.: 1990, 'Occupancy Grids: A Stochastic Spatial Representation for Active Robot Perception'. In: *Proceedings of the Conference on Uncertainty in Artificial Intelligence*. New York, NY, pp. 136–146, Elsevier Science. [28]

Faugeras, O.: 1993, *Three-Dimensional Computer Vision - A geometric viewpoint*. Cambridge: MIT Press. [38, 84, 86]

Fischler, M. A. and R. C. Bolles: 1981, 'Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography'. *Communications of the ACM* **24**, 381–395. [76, 86]

Fu, S., H. ying Liu, L. fang Gao, and Y. xian Gai: 2007, 'SLAM for Mobile Robots Using Laser Range Finder and Monocular Vision'. In: *Mechatronics and Machine vision in Practice*. [14]

Gallegos, G., M. Meilland, P. Rives, and A. I. Comport: 2010, 'Appearance-Based SLAM Relying on a Hybrid Laser/Omnidirectional Sensor'. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. Taipei, Taiwan. [6]

Gallegos, G. and P. Rives: 2010, 'Indoor SLAM based on Composite Sensor Mixing Laser Scans and Omnidirectional Images'. In: *IEEE International Conference on Robotics and Automation*. Anchorage, USA. [5]

Geyer, C.: 2003, 'Catadioptric Projective Geometry:theory and applications'. Ph.D. thesis, University of Pennsylvania. [67]

Geyer, C. and K. Daniilidis: 1999, 'Catadioptric camera calibration'. In: *IEEE International Conference on Computer Vision*, Vol. 1. pp. 398 –404. [70]

Geyer, C. and K. Daniilidis: 2000, 'A Unifying Theory for Central Panoramic Systems and Practical Applications'. In: *European Conference on Computer Vision*. pp. 445–461. [3, 14, 67, 88, 89]

Geyer, C. and K. Daniilidis: 2002, 'Catadioptric Projective Geometry'. *International Journal of Computer Vision* **45**, 223–243. [74]

Gordon, N. J., D. J. Salmond, and A. F. M. Smith: 1993, 'Novel approach to nonlinear/non-Gaussian Bayesian state estimation'. *IE Proceedings-F, Radar and Signal Processing* **140**(2), 107–113. [21]

Greenspan, R. L.: 1996, 'GPS and Inertial Integration'. *Global positioning System: Theory and Applications* **2**, 187–220. [2, 41]

Gutmann, J. S.: 2000, 'Robuste Navigation Autonomer Mobiler System'. Ph.D. thesis, Albert-Ludwigs-Universität Freiburg. [48]

Gutmann, J. S., T. Weigel, and B. Nebel: 2000, 'A Fast, Accurate, and Robust Method for Self-Localization in Polygonal Environments Using Laser-Range-Finders'. *Advanced Robotics Journal* **14**, 2001. [44]

Handschin, J. E. and D. Q. Mayne: 1969, 'Monte Carlo Techniques to Estimate the Conditional Expectation in multi-stage non-linear filtering'. *International Journal of Control* **9**, 547–559. [21]

Harris, C. and M. Stephens: 1988, 'A combined corner and edge detector'. In: *Proceedings of the 4th Alvey Vision Conference*. pp. 147–151. [85]

Hartley, R. I. and A. Zisserman: 2004, *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition. [14, 86]

Hong, J., X. Tan, B. Pinette, R. Weiss, and E. Riseman: 1991, 'Image-based homing'. In: *IEEE International Conference on Robotics and Automation*, Vol. 1. pp. 620 –625. [63]

Huber, P.-J.: 1981, *Robust Statistics*. Wiler, New York. [86, 92, 119]

Irani, M. and P. Anandan: 1999, 'All About Direct Methods'. In: *Workshop on Vision Algorithms: Theory and Practice*. pp. 267–277. [86]

Isard, M. and A. Blake: 1996, 'Contour Tracking By Stochastic Propagation of Conditional Density'. In: *European Conference on Computer Vision*. pp. 343–356. [21]

Joly, C.: 2010, 'Contributions aux méthodes de localisation et cartographie simultanées par vision omnidirectionnelle'. Ph.D. thesis, INRIA Sophia Antipolis, Project-team ARobAS. [xiv, 56, 58]

Julier, S., R. King, B. Colbert, J. Durbin, and L. Rosenblum: 1999, 'The software architecture of a real-time battlefield visualization virtual environment'. In: *IEEE Virtual Reality*. pp. 29 –36. [102]

Julier, S. J. and J. K. Uhlmann: 1997, 'A New Extension of the Kalman Filter to Nonlinear Systems'. In: *International Symposium on Aerospace/Defense Sensing, Simulate and Controls*. pp. 182–193. [20]

Julier, S. J. and J. K. Uhlmann: 2001, 'A Counter Example to the Theory of Simultaneous Localization and Map Building'. In: *IEEE International Conference on Robotics and Automation*. pp. 4238–4243. [21]

Kalman, R. E.: 1960, 'A New Approach to Linear Filtering and Prediction Problems'. *Transactions of the ASME–Journal of Basic Engineering* **82**(Series D), 35–45. [18]

Kalman, R. E. and R. S. Bucy: 1961, 'New Results in Linear Filtering and Prediction Theory'. *Transactions of the ASME Journal of Basic Engineering* **83**(D), 95 – 108. [18]

Kanazawa, K., D. Koller, and S. Russell: 1995, 'Stochastic Simulation Algorithms for Dynamic Probabilistic Networks'. In: *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*. San Francisco, CA, pp. 346–351, Morgan Kaufmann. [21]

Kitagawa, G.: 1996, 'Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models'. *Journal of Computational and Graphical Statistics* **5**(1), 1–25. [21]

Kleeman, L.: 2001, 'Advanced Sonar Sensing'. In: *International Symposium of Robotics Research*. pp. 485–498. [13]

Kleeman, L.: 2003, 'Advanced sonar and odometry error modeling for simultaneous localisation and mapping'. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp. 699–704. [13]

Kuipers, B.: 1978, 'Modeling Spatial Knowledge'. *Cognitive Science* **2**, 129–153. [30]

Kuipers, B. and Y.-T. Byun: 1991, 'A Robot Exploration and Mapping Strategy Based on a Semantic Hierarchy of Spatial Representations'. *Journal of Robotics and Autonomous Systems* **8**, 47–63. [29, 30]

Lemaire, T., C. Berger, I.-K. Jung, and S. Lacroix: 2007, 'Vision-Based SLAM: Stereo and Monocular Approaches'. *International Journal of Computer Vision* **74**(3), 343–364. [14]

Leonard, J. J. and H. F. Durrant-Whyte: 1991, 'Mobile robot localization by tracking geometric beacons'. *IEEE Transactions on Robotics and Automation* **7**(3), 376 –382. [19]

Leonard, J. J. and H. J. S. Feder: 2000, 'A Computationally Efficient Method for Large-Scale Concurrent Mapping and Localization'. In: *Robotics Research: The Ninth International Symposium (D Koditschek and J. Hollerbach, Eds.)*. Snowbird, Utah, USA, pp. 169–176, Springer Verlag. [20]

Leonard, J. J. and P. Newman: 2003, 'Consistent, convergent, and constant-time SLAM'. In: *Proceedings of the 18th international joint conference on Artificial intelligence*. San Francisco, CA, USA, pp. 1143–1150, Morgan Kaufmann Publishers Inc. [20]

Li, D., J. Wang, and S. Babu: 2006, 'Enhancing the Performance of Ultra-Tight Integration of GPS/PL/INS: A Federated Filter Approach'. *Journal of Global Positioning Systems* **5**(1-2), 96–104. [41]

Lin, S.-S. and R. Bajcsy: 2001, 'True single view point cone mirror omni-directional catadioptric system'. Vol. 2. pp. 102 –107 vol.2. [66]

Lu, F. and E. Milios: 1997a, 'Globally Consistent Range Scan Alignment for Environment Mapping'. *Autonomous Robots* **4**, 333–349. [13, 28]

Lu, F. and E. Milios: 1997b, 'Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans'. *Journal of Intelligent and Robotic Systems* **20**, 249–275. [13, 46]

Lucas, B. D. and T. Kanade: 1981, 'An iterative image registration technique with an application to stereo vision'. In: *International Joint Conference on Artificial intelligence*. San Francisco, CA, USA, pp. 674–679, Morgan Kaufmann Publishers Inc. [25]

Malis, E.: 2004, 'Improving vision-based control using efficient second-order minimization techniques'. In: *IEEE International Conference on Robotics and Automation*, Vol. 2. pp. 1843 – 1848. [87]

Malis, E.: 2007, 'An efficient unified approach to direct visual tracking of rigid and deformable surfaces'. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 2. pp. 2729 –2734. [xiv, 87]

Martinez-Cantin, R. and J. Castellanos: 2005, 'Unscented SLAM for large-scale outdoor environments'. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp. 3427 – 3432. [20]

Mei, C., 'Calibration Toolbox'. http://www.robots.ox.ac.uk/~cmei/Toolbox.html. [7]

Mei, C.: 2007, 'Laser-Augmented Omnidirectional Vision for 3D Localisation and Mapping'. Ph.D. thesis, INRIA Sophia Antipolis, Project-team ARobAS. [xiii, 15, 64, 65, 67, 89]

Mei, C., S. Benhimane, E. Malis, and P. Rives: 2008, 'Efficient homography-based tracking and 3D reconstruction for single viewpoint sensors'. *IEEE Transactions on Robotics*. [14]

Mei, C. and E. Malis: 2006, 'Fast central catadioptric line extraction, estimation, tracking and structure from motion'. In: *IEEE International Conference on Intelligent Robots and Systems*. [76]

Mei, C. and P. Rives: 2006a, 'Calibrage non biaise d'un capteur central catadioptrique'. In: *RFIA, Reconnaissance des Formes et Intelligence Artificielle*. [70]

Mei, C. and P. Rives: 2006b, 'Calibration between a Central Catadioptric Camera and a Laser Range Finder for Robotic Applications'. In: *IEEE International Conference on Robotics and Automation*. [7, 71]

Mei, C. and P. Rives: 2007, 'Single View Point Omnidirectional Camera Calibration from Planar Grids'. In: *IEEE International Conference on Robotics and Automation*. [70]

Miro, J. V., G. Dissanayake, and W. Zhou: 2005, 'Vision-based SLAM using natural features in indoor environments'. In: *International Conference on Intelligent Sensors, Sensor Networks and Information Processing*. pp. 151–156. [14]

Montemerlo, M.: 2003, 'FastSLAM: A Factored Solution to the Simultaneous Localization and Mapping Problem with Unknown Data Association'. Ph.D. thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA. [22]

Moravec, H.: 1996, 'Robot Spatial Perception by Stereoscopic Vision and 3D Evidence Grids'. Technical Report CMU-RI-TR-96-34, Robotics Institute, Pittsburgh, PA. [28]

Moravec, H. and A. E. Elfes: 1985, 'High Resolution Maps from Wide Angle Sonar'. In: *Proceedings International Conference on Robotics and Automation*. pp. 116 – 121. [28]

Mouragnon, E., M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd: 2006, 'Monocular Vision-Based SLAM for Mobile Robots'. In: *International Conference on Pattern Recognition*. Washington, DC, USA, pp. 1027–1031. [14]

Mouragnon, E., M. Lhuillier, D. M., F. Dekeyser, and P. Sayd: 2009, 'Generic and real-time structure from motion using local bundle adjustment'. *Image Vision Computer* **27**, 1178–1193. [24]

Mourikis, A. I., N. Trawny, S. I. Roumeliotis, A. E. Johnson, A. Ansar, and L. Matthies: 2009, 'Vision-aided inertial navigation for spacecraft entry, descent, and landing'. *IEEE Transactions on Robotics* **25**(2), 264–280. [41]

Moutarlier, P. and R. Chatila: 1989, 'Stochastic multisensory data fusion for mobile robot location and environment modeling'. In: *5th International Symposium on Robotics Research*. Tokio. [19]

Moutarlier, P. and R. Chatila: 1990, 'An experimental system for incremental environment modelling by an autonomous mobile robot'. In: *Experimental Robotics I*, Vol. 139 of *Lecture Notes in Control and Information Sciences*. pp. 327–346. [19]

Nakamura, T., S. Takamura, and M. Asada: 1996, 'Behavior-Based Map Representation for a Sonar-based Mobile Robot by Statistical Methods'. In: *International Conference on Intelligent Robots and Systems*. pp. 276–283. [13]

Nayar, S. K.: 1997, 'Catadioptric Omnidirectional Camera'. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. Washington, DC, USA, p. 482, IEEE Computer Society. [14]

Nebot, E., F. Masson, J. Guivant, and H. Durrant-Whyte: 2003, 'Robust Simultaneous Localization and Mapping for Very Large Outdoor Environments'. In: *Experimental Robotics VIII*, Vol. 5 of *Springer Tracts in Advanced Robotics*. Springer Berlin / Heidelberg, pp. 200–209. [21]

Neira, J. and J. D. Tardos: 2001, 'Data association in stochastic mapping using the joint compatibility test'. *IEEE Transactions on Robotics and Automation* **17**(6), 890–897. [20, 21]

Ni, K., D. Steedly, and F. Dellaert: 2007, 'Out-of-Core Bundle Adjustment for Large-Scale 3D Reconstruction'. In: *IEEE International Conference on Computer Vision*. [24]

Nieto, J., T. Bailey, and E. Nebot: 2005, 'Scan-SLAM: Combining EKF-SLAM and scan correlation'. In: *International Conference on Field and Service Robotics*. pp. 129–140. [13]

Nieto, J., T. Bailey, and E. Nebot: 2007, 'Recursive scan-matching SLAM'. *Robotics and Autonomous Systems* **55**, 39–49. [13]

Nieto, J., T. Bailey, and E. Nebot: 2008, 'Scan-SLAM: Recursive Mapping and Localisation with Arbitrary-Shaped Landmarks'. In: *Robotics Science and Systems Conference*. [44]

Nieto, J., J. Guivant, and E. Nebot: 2004, 'The HYbrid metric maps (HYMMs): a novel map representation for DenseSLAM'. In: *IEEE International Conference on Robotics and Automation*, Vol. 1. pp. 391 – 396. [31]

Paz, L. M., J. Guivant, J. D. Tardós, and J. Neira: 2007, 'Data Association in O(n) for Divide and Conquer SLAM'. In: *Robotics: Science and Systems*. [21]

Ramos, F., J. Nieto, and H. Durrant-Whyte: 2007, 'Recognising and modelling landmarks to close loops in outdoor SLAM'. In: *IEEE International Conference on Robotics and Automation*. Rome, Italy. [44]

Rees, D. W.: 1970, 'Panoramic television viewing system'. *United States Patent No.3,505,465*. [63]

Rubin, D. B.: 1988, 'Using the SIR algorithm to simulate posterior distributions'. In: *Bayesian Statistics 3*, J.M. Bernardo, M.H. DeGroot, D.V. Lindley, and A.F.M. Smith, editors. Oxford University Press, p. 395–402. [21]

Rusinkiewicz, S. and M. Levoy: 2001, 'Efficient Variants of the ICP Algorithm'. In: *International Conference on 3D Digital Imaging and Modeling*. pp. 145–152. [46]

Scaramuzza, D., A. Martinelli, and R. Siegwart: 2006, 'A Toolbox for Easily Calibrating Omnidirectional Cameras'. In: *International Conference on Intelligent Robots and Systems*. pp. 5695 –5701. [70]

Se, S., D. G. Lowe, and J. J. Little: 2005, 'Vision-based global localization and mapping for mobile robots'. *IEEE Transactions on Robotics* **21**, 364–375. [85]

Shi, J. and C. Tomasi: 1994, 'Good features to track'. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. pp. 593 –600. [25]

Silveira, G., E. Malis, and P. Rives: 2007, 'An Efficient Direct Method for Improving visual SLAM'. In: *IEEE International Conference on Robotics and Automation*. Rome, Italy. [14]

Smith, R. and P. Cheeseman: 1986, 'On the Representation and Estimation of Spatial Uncertainty'. *International Journal of Robotics Research* **5**, 56–68. [2, 16, 19]

Smith, R., M. Self, and P. Cheeseman: 1986, 'Estimating Uncertain Spatial Relationships in Robotics'. *Conference on Uncertainty in Artificial Intelligence* pp. 435–461. [2, 16, 19]

Soler, L., S. Nicolau, J. Schmid, C. Koehl, J. Marescaux, X. Pennec, and N. Ayache: 2004, 'Virtual Reality and Augmented Reality in Digestive Aurgery'. In: *IEEE and ACM International Symposium on Mixed and Augmented Reality*. pp. 278 – 279. [102]

Soloviev, A.: 2008, 'Tight coupling of GPS, laser scanner, and inertial measurements for navigation in urban environments'. In: *IEEE/ION Position, Location and Navigation Symposium*. pp. 511–525. [41]

Stein, G. and A. Shashua: 2000, 'Model-based brightness constraints: on direct estimation of structure and motion'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(9), 992 –1015. [86]

Stone, H.: 1996, 'Mars Pathfinder Microrover A Small, Low-Cost, Low-Power Spacecraft'. In: *Proceedings of the AIAA Forum on Advanced Developments in Space Robotics*. [1]

Swaminathan, R., M. D. Grossberg, and S. K. Nayar: 2006, 'Non-Single Viewpoint Catadioptric Cameras: Geometry and Analysis'. *International Journal of Computer Vision* **66**(3), 211–229. [63]

Tardós, J. D., J. Neira, P. M. Newman, and J. J. Leonard: 2002, 'Robust mapping and localization in indoor environments using sonar data'. *International Journal of Robotics Research* **21**, 311–330. [13, 21]

Thrun, S.: 1998, 'Learning metric-topological maps for indoor mobile robot navigation'. *Artificial Intelligence* **99**(1), 21 – 71. [31]

Thrun, S.: 2002, 'Robotic Mapping: A Survey'. In: G. Lakemeyer and B. Nebel (eds.): *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann. [26]

Thrun, S., W. Burgard, D. Fox, H. Hexmoor, and M. Mataric: 1998a, 'A Probabilistic Approach to Concurrent Mapping and Localization for Mobile Robots'. In: *Machine Learning*, Vol. 31. pp. 29–53. [24]

Thrun, S., J. S. Gutmann, D. Fox, W. Burgard, and B. Kuipers: 1998b, 'Integrating topological and metric maps for mobile robot navigation: A statistical approach'. In: *Proceedings of the National Conference on Artificial Intelligence (AAAI)*. [30]

Thrun, S., C. Martin, Y. Liu, D. Hahnel, R. Emery-Montemerlo, D. Chakrabarti, and W. Burgard: 2004, 'A real-time expectation-maximization algorithm for acquiring multiplanar maps of indoor environments with mobile robots'. *IEEE Transactions on Robotics and Automation* **20**(3), 433 – 443. [24, 28]

Tomatis, N., I. Nourbakhsh, and R. Siegwart: 2001, 'Simultaneous localization and map building: a global topological model with local metric maps'. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 1. pp. 421 –426. [31]

Torr, P. and A. Zisserman: 2000, 'Feature Based Methods for Structure and Motion Estimation'. In: *Vision Algorithms: Theory and Practice*. pp. 278–295, Springer-Verlag. [85]

Triggs, B., P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon: 1999, 'Bundle Adjustment - A Modern Synthesis'. In: *Proceedings of the International Workshop on Vision Algorithms: Theory and Practice*. London, UK, pp. 298–372, Springer-Verlag. [23]

Vasseur, P. and E. Mouaddib: 2004, 'Central Catadioptric Line Detection'. In: *Fifteenth British Machine Vision Conference*. London, UK, pp. 57–66. [70]

Victorino, A.: 2002, 'La commande référencée capteur: une approche robuste au problème de navigation, localisation, et cartographie simultaées pour un robot d'intérieur'. Ph.D. thesis, INRIA Sophia Antipolis, Project-team ICARE. [36, 48]

Victorino, A. and P. Rives: 2004, 'An Hybrid Representation Well-adapted to the Exploration of Large Scale Indoors Environments'. In: *IEEE International Conference on Robotics and Automation*. New Orleans, Lu, USA. [30]

Victorino, A., P. Rives, and J.-J. Borrelly: 2003a, 'Safe Navigation for Indoor Mobile Robots, Part I: A Sensor-Based Navigation Framework'. *International Journal of Robotics Research* **22**(12), 1005–1019. [30]

Victorino, A., P. Rives, and J.-J. Borrelly: 2003b, 'Safe Navigation for Indoor Mobile Robots, Part II: Exploration, Self-Localization and Map Building'. *International Journal of Robotics Research* **22**(12), 1019–1041. [30]

Watanabe, M. and S. K. Nayar: 1995, 'Telecentric Optics for Computational Vision'. In: *European Conference on Computer Vision*. pp. 439–451. [64]

Wei, S.-C., Y. Yagi, and M. Yachida: 1996, 'On-line Map Building Based On Ultrasonic and Image Sensor'. In: *IEEE International Conference on Systems, Man and Cybernetics*, Vol. 2. pp. 1601–1605. [13]

Weiß, G. and E. V. Puttkamer: 1995, 'A Mapbased On Laserscans Without Geometric Interpretation'. In: *Intelligent Autonomous Systems*. pp. 403–407. [46]

West, M.: 1993, 'Mixture models, Monte Carlo, Bayesian updating and dynamic models'. *Computing Science and Statistics* **24**, 325–333. [21]

Xu, L. and E. Oja: 1993, 'Randomized Hough Transform (RHT): Basic Mechanisms, Algorithms, and Computational Complexities'. *CVGIP: Image Understanding* **57**(2), 131 – 154. [75, 76]

Yagi, Y. and S. Kawato: 1990, 'Panorama scene analysis with conic projection'. In: *IEEE International Workshop on Intelligent Robots and Systems*, Vol. 1. pp. 181–187. [63]

Yamazawa, K., Y. Yagi, and M. Yachida: 1993, 'Omnidirectional imaging with hyperboloidal projection'. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 2. pp. 1029 –1034. [63]

Ying, X. and Z. Hu: 2004, 'Catadioptric Camera Calibration Using Geometric Invariants'. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **26**, 1260–1271. [70, 74]

Zhang, Q. and R. Pless: 2004, 'Extrinsic calibration of a camera and laser range finder (improves camera calibration)'. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Vol. 3. pp. 2301 – 2306. [71]

Développement d’un Capteur Composite Vision/Laser à Couplage Serré pour le SLAM d’Intérieur

Résumé : Depuis trois décennies, la navigation autonome en environnement inconnu est une des thématiques principales de recherche de la communauté robotique mobile. En l’absence de connaissance sur l’environnement, il est nécessaire de réaliser simultanément les tâches de localisation et de cartographie qui sont extrêmement interdépendantes. Ce problème est connu sous le nom de **SLAM** (*Simultaneous Localization And Mapping*).

Pour obtenir des informations précises sur leur environnement, les robots mobiles sont équipés d’un ensemble de capteurs appelé *système de perception* qui leur permet d’effectuer une localisation précise et une reconstruction fiable et cohérente de leur environnement. Nous pensons qu’un système de perception composé de l’odométrie du robot, d’une camera omnidirectionnelle et d’un télémètre laser 2D est suffisant pour résoudre de manière robuste les problèmes de SLAM.

Dans ce contexte, nous proposons une approche *appearance-based* pour résoudre les problèmes de SLAM et effectuer une reconstruction 3D fiable de l’environnement. Cette approche repose sur un couplage serré entre les capteurs laser et omnidirectionnel permettant d’exploiter au mieux les complémentarités des deux types de capteurs. Une représentation originale et générique robot-centrée est proposée. Une vue augmentée sphérique est construite en projetant dans l’image omnidirectionnelle les mesures de profondeur du télémètre laser et une estimation de la position du sol. Notre méthode de localisation de type *appearance-based* minimise une fonction de coût non-linéaire directement construite à partir de la vue sphérique augmenté décrite précédemment. Cependant comme dans toutes les méthodes récursives d’optimisation, des problèmes de convergence peuvent survenir quand l’initialisation est loin de la solution. Ce problème est aussi présent dans notre méthode où une initialisation suffisamment proche de la solution est nécessaire pour s’assurer une convergence rapide et pour réduire les couts de calcul. Pour cela, on utilise un algorithme de PSM amélioré pour construire une prédection du déplacement du robot.

Mots clés : SLAM, Localisation, Cartographie, Optimisation, Odometrie Visuelle, Laser, Vision Omnidirectionelle.

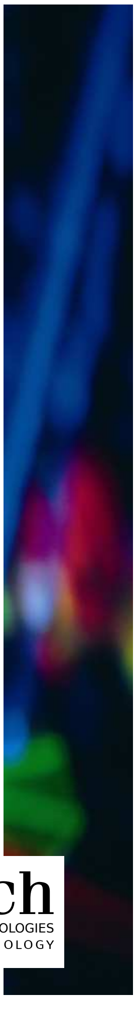
Development of a Tightly-Coupled Composite Vision/Laser Sensor for Indoor SLAM

Abstract: Autonomous navigation in unknown environments has been the focus of attention in the mobile robotics community for the last three decades. When neither the location of the robot nor a map of the region are known, localization and mapping are two tasks that are highly inter-dependent and must be performed concurrently. This problem, is known as *Simultaneous Localization and Mapping* (**SLAM**).

In order to gather accurate information about the environment, mobile robots are equipped with a variety of sensors that together form a perception system that allows accurate localization and reconstruction of reliable and consistent representations of the environment. We believe that a perception system composed of the odometry of the robot, an omnidirectional camera and a 2D laser range finder provide enough information to solve the SLAM problem robustly.

In this context we propose an *appearance-based* approach to solve the SLAM problem and reconstruct a reliable 3D representation of the environment. This approach relies on a tightly-coupled laser/omnidirectional sensor in order to take profit of the complementarity of each sensor modality. A novel generic robot-centered representation that is well adapted to the appearance-based SLAM is proposed. This augmented spherical view is constructed using the depth information from the laser range finder and the floor plane, together with lines extracted from the omnidirectional image. The appearance-based localization method minimizes a non-linear cost function directly built from the augmented spherical view. However, recursive optimization methods suffer from convergence problems when initialized far from the solution. This is also true for our method where an initialization sufficiently close to the solution is needed to ensure rapid convergence and reduce computational cost. A Enhanced Polar Scan Matching algorithm is used to obtain this initial guess of the position of the robot to initialize the algorithm.

Keywords: SLAM, Localization, Mapping, Optimization, Visual Odometry, Laser, Omnidirectional Vision.



MINES
ParisTech

ParisTech
INSTITUT DES SCIENCES ET TECHNOLOGIES
PARIS INSTITUTE OF TECHNOLOGY