SELECTIVE SUBMAP JOINING SLAM FOR AUTONOMOUS VEHICLES

Josep M. AULINAS MASÓ

Dipòsit legal: GI-1355-2011
http://hdl.handle.net/10803/48718

Universitat de Girona

P H D T H E S I S

Selective Submap Joining SLAM for autonomous vehicles

Presented by

Josep Aulinas

Institute of Informatics and Applications

COMPUTER VISION AND ROBOTICS GROUP

Thesis Advisors:

Joaquim Salvi
VICOROB
University of Girona

Yvan R. Petillot
OSL
Heriot Watt University

We, Joaquim Salvi and Yvan Petillot, professors of the Computer Vision and Robotics group at the University of Girona and the Ocean Systems Laboratory at Heriot Watt University, respectively,

ATTEST:

that this thesis "Selective Submap Joining SLAM for Autonomous Vehicles", submitted by Josep Aulinas for the degree of European Ph.D. in Technology, was carried under our supervision.

Signatures

Selective Submap Joining SLAM for Autonomous Vehicles

by Josep Aulinas

Submitted to the Doctorate on Technology Program
in fulfillment of the requirements for the degree of
Doctor of Philosophy

Abstract: Simultaneous Localization and Mapping (SLAM) do not result in consistent maps of large areas because of gradual increase of the uncertainty for long term missions. In addition, as the size of the map grows the computational cost increases, making SLAM solutions unsuitable for on-line applications. This thesis surveys SLAM approaches paying special attention to those approaches aimed to work on large scenarios. In addition, special focus is given to existing underwater SLAM applications. A technique based on using independent local maps together with a global stochastic map is then presented. This technique is called Selective Submap Joining SLAM (SSJS). A global map contains relative transformations between local maps, which are updated once a new loop is detected. Maps sharing several features are fused, maintaining the correlation between landmarks and vehicle's pose. The use of local maps reduces computational costs and improves map consistency. This approach is compared to state of the art techniques using the Victoria Park dataset, a well-known benchmark within SLAM community. Synthetic and experimental results show that SSJS approach is able to map large areas consistently with lower computational cost compared to state of the art methods. Once demonstrated the feasibility of SSJS, the method is adapted to be used on Autonomous Underwater Vehicle (AUV). In this thesis, two sets of experiments are presented: 1) the REMUS-100 AUV with a side-scan sonar, and 2) the SPARUS AUV with a down-looking camera. In both cases it is necessary to define their corresponding motion and observation models. Moreover, imaging techniques are necessary in order to extract and match robust landmarks, necessary for SLAM algorithms to work. Experiments on real data demonstrate the capability to produce consistent localization and precise mapping by combining automatic landmark extraction with SSJS.

Keywords: Simultaneous Localization and Mapping (SLAM), submapping SLAM, Autonomous Underwater Vehicle (AUV), underwater imaging.

Thesis Advisor: Professor Joaquim Salvi (VICOROB – University of Girona)
Thesis Co-Advisor: Professor Yvan R. Petillot (OSL – Heriot Watt University)

Unió Selectiva de Submapes en SLAM per Vehicles Autònoms

Josep Aulinas

Pel Programa de Doctorat en Tecnologia
en el compliment dels requisits per a obtenir
el grau de Doctor

Resum: Els algoritmes de localització i creació de mapes simultàniament (Simultaneous Localization and Mapping – SLAM) no produeixen mapes correctes de grans àrees a causa de l'augment gradual de la incertesa en les missions de llarga durada. El cost de computació augmenta a mesura que el mapa creix, de manera que les solucions de SLAM no són aplicables a temps real. Aquesta tesi presenta un estudi de les tècniques de SLAM, donant especial èmfasi a aquells treballs pensats per operar en entorns grans. També s'estudien aquells treballs de SLAM centrats en ambients submarins. En aquest context, es proposa una nova tècnica basada en l'ús de submapes independents i un mapa estocàstic global. Aquesta tècnica s'ha anomenat Unió Selectiva de Submapes en SLAM (SSJS) per Vehicles Autònoms. El mapa global conté les transformacions relatives entre mapes, que s'actualitzen en revisitar zones conegudes, és a dir, s'observen fites que ja formaven part del mapa. Així doncs, els submapes que comparteixen informació es fusionen, mantenint les correlacions entre el vehicle i les fites. L'ús de submapes redueix el cost de càlcul i millora la consistència del mapa. La tècnica proposada es compara amb les tècniques existents utilitzant les dades del Victoria Park, una base de dades de referència dins la comunitat de SLAM. Resultats sintètics i experimentals mostren que l'SSJS és capaç de cartografiar consistentment zones de grans dimensions i amb un cost menor en comparació a les altres tècniques. Un cop demostrada la viabilitat del sistema, el mètode s'adapta per a ser utilitzat en vehicles submarins autònoms. En aquesta tesi, es presenten dues sèries d'experiments: 1) amb el REMUS-100 equipat amb un sonar d'escombratge lateral, i 2) amb el SPARUS equipat amb una càmera mirant cap al fons. En ambdós casos, cal definir els models matemàtics que descriuen la dinàmica del vehicle i el principi de mesura dels sensors. A més a més, es desenvolupen tècniques de visió artificial per a extreure i associar fites, necessàries en els algoritmes de SLAM. Els experiments duts a terme utilitzant dades reals demostren la capacitat del sistema per localitzar el vehicle i generar cartografia, combinant l'extracció automàtica de fites amb el SSJS.

Director de Tesis: Professor Joaquim Salvi (VICOROB – Universitat de Girona)
Co-director de Tesis: Professor Yvan R. Petillot (OSL – Heriot Watt University)

Unión Selectiva de Submapas en SLAM para Vehículos Autónomos

Josep Aulinas

Programa de Doctorado en Tecnología
en cumplimiento de los requisitos para otener
el grado de Doctor

Resumen: Los algoritmos de localización y creación de mapas simultáneamente (Simultaneous Localization and Mapping – SLAM) no producen mapas correctos de grandes áreas debido al aumento gradual de la incertidumbre en las misiones de larga duración. Además, el coste de computación aumenta a medida que el mapa crece, de modo que las soluciones de SLAM no son aplicables a tiempo real. Esta tesis presenta un estudio de las técnicas de SLAM, prestando especial atención en aquellos trabajos pensados para operar en entornos grandes. También se estudian los trabajo de SLAM centrados en ambientes submarinos. En este contexto, se propone una nueva técnica basada en el uso de submapas independientes combinados con un mapa estocástico global. Esta técnica se ha llamado Unión Selectiva de Submapas en SLAM para Vehículos Autónomos. El mapa global contiene las transformaciones relativas entre mapas, que se actualizan al revisitar zonas conocidas, es decir, se observan puntos de referencia que ya formaban parte del mapa. Así pues, los submapas que comparten información se fusionan, manteniendo las correlaciones entre el vehículo y los puntos de referencia. El uso de submapas reduce el coste de computación y mejora la consistencia del mapa. La técnica propuesta se compara con las técnicas existentes utilizando los datos del Victoria Park, una base de datos de referencia dentro la comunidad de SLAM. Resultados sintéticos y experimentales muestran que la SSJS es capaz de cartografiar zonas de grandes dimensiones consistentemente y con un coste menor en comparación a las otras técnicas. Una vez demostrada la viabilidad del sistema, el método se adapta para ser utilizado en vehículos submarinos autónomos. En esta tesis, se presentan dos series de experimentos: 1) con el REMUS-100 equipado con un sonar de barrido lateral, y 2) con el SPARUS que lleva una cámara mirando hacia el fondo. En ambos casos, hay que definir los modelos matemáticos que describen la dinámica del vehículo y el principio de medición de los sensores. Además, se desarrollan técnicas de visión artificial para extraer y asociar puntos de referencia, muy necesarios en los algoritmos de SLAM. Los experimentos llevados a cabo utilizando datos reales demuestran la capacidad del sistema para localizar el vehículo y generar cartografía, combinando la extracción automática de puntos de referencia con el SSJS.

Director de Tesis: Profesor Joaquim Salvi (VICOROB – Universidad de Girona)
Co-director de Tesis: Profesor Yvan R. Petillot (OSL – Heriot Watt University)

Acknowledgments

I am heartily thankful to my supervisors Quim and Yvan, whose encouragement, guidance and support enabled me to understand the subject and they have made available support in many ways. I also owe my deepest gratitude to Xavi for his constant aid.

It is an honour for me to be part of the Computer Vision and Robotics group and I feel indebted to many of my colleagues to help me and specially for their friendship. I would like to show my gratitude to Luca for being always there. To my "nearest neighbours" in the lab, Sik, Sergio, Albert and Meritxell, many thanks for your moral and scientific support. I would also like to thank Josep, Tudor and Nuno for the many discussions maintained in the student canteen; Pio for his predisposition to share knowledge; and Ricard for his technical support in the lab.

Many thanks to the guys in CIRS for providing the data we used in some of the experiments presented in this thesis. Also, many thanks to those who I had the pleasure to shared a trip to a conference with, to the secretaries of the institute for managing the bureaucracy behind these trips, and to the principal investigators of the projects that funded these trips.

It is a pleasure to thank the colleagues from Ocean Systems Laboratory, who helped me during my stay at their lab. Spacial mention to Francesco and Nicolas for their friendship during my stay. Also many thanks to Yvan and Yan for making available the data that we used in some of the real experiments presented in this document.

This thesis would not have been possible without my parents and sisters support and their advices. Specially, Montse and Rosa correcting my English.

Lastly, I offer my regards and many thanks to all of those who supported me in any aspect during the completion of the project.

Agraïments

Estic sincerament agraït als meus supervisors Quim i Yvan, pel seu suport i orientació. També, dec el més profund agraïment a en Xavi per la seva ajuda constant.

És un honor per a mi formar part del grup de Visió per Computador i Robòtica i em sento en deute amb molts dels meus col.legues per la seva juda i sobretot per la seva amistat. M'agradaria mostrar el meu agraïment a en Luca per ser-hi sempre quan he necessitat ajuda. També moltes gràcies als "veïns més propers" al laboratori: en Sik, en Sergio, l'Albert i la Meritxell, moltes gràcies pel vostre suport moral i científic. També m'agradaria donar les gràcies a en Josep, en Tudor i en Nuno per les moltes converses mantingudes a la hora de dinar; a en Pio per la seva predisposició a compartir els seus coneixements; i a en Ricard pel seu suport tècnic al laboratori.

Moltes gràcies als companys del CIRS per haver-nos proporcionat les dades que hem utilitzat en alguns dels experiments presentats en aquesta tesi. També, moltes gràcies als qui he tingut al plaer d'acompanyar en els viatges a conferències i torbades científiques, a les secretàries de l'institut per gestionar aquests viatges i als investigadors principals dels projectes que els han finançat.

És un plaer agrair als col.legues del Ocean Systems Laboratory, que em van ajudar durant la meva estada al seu laboratori. Menció especial per en Francesco i en Nicolas per la seva amistat durant la meva estada. També moltes gràcies a l'Yvan i a en Yan per posar a disposició les dades que hem utilitzat en alguns dels experiments reals que es presenten en aquest document.

Aquesta tesi no hauria estat possible sense el suport dels meus pares i germanes i als seus consells. Especialment, la Montse i la Rosa per corregir el meu anglès.

Finalment, salutacions i gràcies per a tots els que m'han donat suport en algun aspecte durant la realització del projecte.

Contents

List of Figures

List of Tables

List of Algorithms

List of Acronyms

ACFR	Australian Center for Field Robotics
AUV	Autonomous Underwater Vehicle
CEKF	Compressed Extended Kalman Filter
CICYT	Comisión Interministerial de Ciencia y tecnología
CIRS	Research Centre in Underwater Robotics
CIS	Conditionally Independent SLAM
CLSF	Constrained Local Submap Filter
CML	Concurrent Mapping and Localization
CTS	Constant Time SLAM
DCS	Divide and Conquer SLAM
DOF	Degree Of Freedom
DSM	Decoupled Stochastic Mapping
DVL	Doppler Velocity Log
EIF	Extended Information Filter
EKF	Extended Kalman Filter
EM	Expectation Maximization
EU	European Union
GCM	Global Climate Models
GPS	Global Positioning System
GRV	Gaussian Random Variable
HS	Hierarchical SLAM
HWU	Heriot Watt University
IF	Information Filter
IMU	Inertial Measuremet Unit
ITN	Initial Training Network
JCBB	Joint Compatibility Branch and Bound
JRI	Joint Research Institute

KF	Kalman Filter
LBL	Long-BaseLine
LMJ	Local Map Joining
MCM	Mine Counter Measures
MHTF	Multiple Hypothesis Tracking Filter
ML	Maximum Likelihood
MMSE	Minimum mean-square Error
MRF	Markov Random Field
NIS	Normalized Innovation Squared
NN	Nearest Neighbor
OSL	Ocean Systems Laboratory
PF	Particle Filter
PHD	Probability Hypothesis Density
REMUS	Remote Environmental Measuring UnitS
RJC	Randomized Joint Compatibility
RoI	Region of Interest
ROV	Remotely Operated Vehicle
RTS	Rauch-Tung-Striebel
SBL	Short-BaseLine
SEIF	Sparse Extended Unformation filter
SFM	structure from Motion
SIFT	Scale Invariant Feature Transform
SLAM	Simultaneous Localization and Mapping
SMC	Sequential Monte-Carlo
SSJS	Selective Submap Joining SLAM
SURF	Speed Up Robust Features
UDG	Universitat de Girona
UKF	Unscented Kalman Filter

USBL	Ultra Short-Baseline
VICOROB	Computer Vision and Robotics group
VISP	Vision, Image and Signal Processing

Chapter 1

Introduction

Contents

This thesis brings together aspects of underwater robotics, localization and mapping, and underwater imaging techniques. While the following chapters focus on the details for each of these aspects, this chapter briefly reviews the motivations and context behind this thesis. Section 1.1 presents the motivations behind this work. Section 1.2 describes the context in which this work has been carried out. Afterwards, an objectives statement is given in Section 1.3, and finally, the outline of this document is given, with a brief summary for each chapter.

1.1 Motivation

It is not really necessary to fly beyond the atmosphere to find unknown worlds. Here on Earth there are many untouched areas; places where no man has ever gone, among them there is the Hydrosphere. The Hydrosphere is composed of oceans, seas, lakes and other water systems present in the most superficial layer of the Earth. Thus, if one accepts that the Earth's surface is about seventy percent covered in water and that only three percent of this water is continental, one can strongly claim that there is a magnificent offshore world to explore. Scientists and science would make a significant step forward if the deepest secrets of this underwater world were known. Unfortunately, one faces complex technical issues and major limitations for human survival down there due to high pressure and vast darkness even only a few meters from the surface, as well as the impossibility of breathing underwater.

You may ask yourself, what significance seas and oceans may have for us. Water is vital. Firstly, human bodies are about seventy percent water [Harper 1977]. Water is an essential element in the human diet, without it a person could only survive three or four days at most. Secondly, water is a key element for our planet's climate [Parry 2007]. For example

Figure 1.1: Types of offshore oil and gas structures: 1, 2) conventional fixed platforms; 3) compliant tower; 4, 5) vertically moored tension leg and mini-tension leg platform; 6) Spar ; 7,8) Semi-submersibles ; 9) Floating production, storage, and offloading facility; 10) sub-sea completion and tie-back to host facility [NOAA 2008].

the European continent remains at habitable temperatures thanks to great ocean currents, specifically the Gulf Stream which brings warm water from areas near the equator and reaches the upper latitudes where the water meets with that from the icy polar cap [Willis 1995]. This phenomenon is key to keeping Europe from being under a constant layer of ice. Thirdly, the Hydrosphere is the habitat of many species. Finally, this huge mass of water is a source of energy that should not be underestimated, neither its constant effect exerted on the Earth, shaping it and giving it incredibly beautiful forms, nor its enormous potential as an energy source, especially nowadays with a more energy demanding society worldwide.

The preceding paragraph presented several points which are scientifically very interesting. The fact that humans and other living creatures need water to survive rises the interest on water related topics research. Finding out more about the Hydrosphere may well help us to become more efficient in terms of water consumption and use. Furthermore, observing and studying the Hydrosphere's behaviour will provide important information that might be very useful to complement and complete Global Climate Models (GCM), thus allowing us to foresee and prevent catastrophic consequences caused by climate changes. In addition, humans consume a large amount of fish, hence the need for massive aqua farming. Aqua farming facilities require special control and maintenance. It is also worth highlighting the growing importance of the oil industry's interest in drilling for oil underwater (see Figure 1.1), which is now becoming economically desirable thanks to unbridled consumption and the consequent increase in fuel prices. This type of industry requires intervention capabilities, not only for its construction and maintenance, but also in case of accidents. There are many examples of offshore oil catastrophes, for instance the oil spill along the coast of Galicia in 2002, where an oil tanker sank, or the more recent example of the catastrophic oil spill in the Gulf of Mexico in 2010. Other industries such as offshore wind farms require underwater

intervention systems during their installation and maintenance.

To achieve some of these objectives, one could send a fleet of divers to explore and collect samples, but our oceans are too vast. Moreover, water is a hostile environment for humans, first because a person can not breathe in it and second because the pressure increases dramatically with depth and human bodies are not prepared for such pressure levels. Over the past few decades, several underwater vehicles have been developed. Initially, underwater submersibles carrying a crew were used. Afterwards, unmanned vehicles were designed. These vehicles can travel longer distances and access hazardous environments without endangering human beings. Two good examples of this kind of vehicle are the Remotely Operated Vehicle (ROV) and, more recently, the AUV.

ROVs are unmanned vehicles that communicate with a vessel on the surface through a cable, also referred to as an umbilical. These cables shuttle energy, control data, and collected data like measures of vehicle's state or images acquired by an on-board camera back and forth. ROVs have the advantages of ease of use, reliability due to human teleoperation, and large range and depth of operation due to use of the umbilical cord. However, the major disadvantages of ROV lie in their need for skilled teleoperation by humans and high operational costs. Additional drawbacks to the ROV technology include operator fatigue, low operational efficiency, limited access to man made structres and possible loss of the vehicle due to umbilical cord damage [Xu 2006]. Instead, an AUV is completely wireless, no physical connection with the mother vessel is needed. AUVs are fully autonomous systems that use complex control and navigational algorithms. Most of these algorithms are hot topics in research nowadays. For instance, SLAM, also known as Concurrent Mapping and Localization (CML), is one of the fundamental challenges for autonomous vehicles [Durrant-Whyte 2006].

SLAM is a process by which a mobile robot can build a map of an environment and, at the same time, use this map to deduce its location. Initially, both the map and the vehicle's position are not known, but the vehicle has a known kinematic model, i.e. the movement of the vehicle can be estimated. The vehicle is moving in an unknown environment populated by artificial or natural objects. These objects are observed through on-board sensors and are then used as landmarks. The principle behind SLAM is similar to what humans do when walking in unknown areas; that is, after a while a human is more or less capable of telling where he thinks he is, but without being completely sure about his position. He is capable of guessing thanks to his senses. It will not be until the moment a known landmark is seen again, i.e. a sign, a mountain, a house..., that the walker will be able to tell where he is with a higher level of certainty. Similarly, an autonomous vehicle simultaneously estimates both its and landmark locations through SLAM techniques, while navigating a new area. The SLAM problem involves finding appropriate representation for both the observation and the motion models. These models are used to estimate the position of the vehicle. In order to do this, the vehicle uses sensors capable of acquiring measurements of the relative location

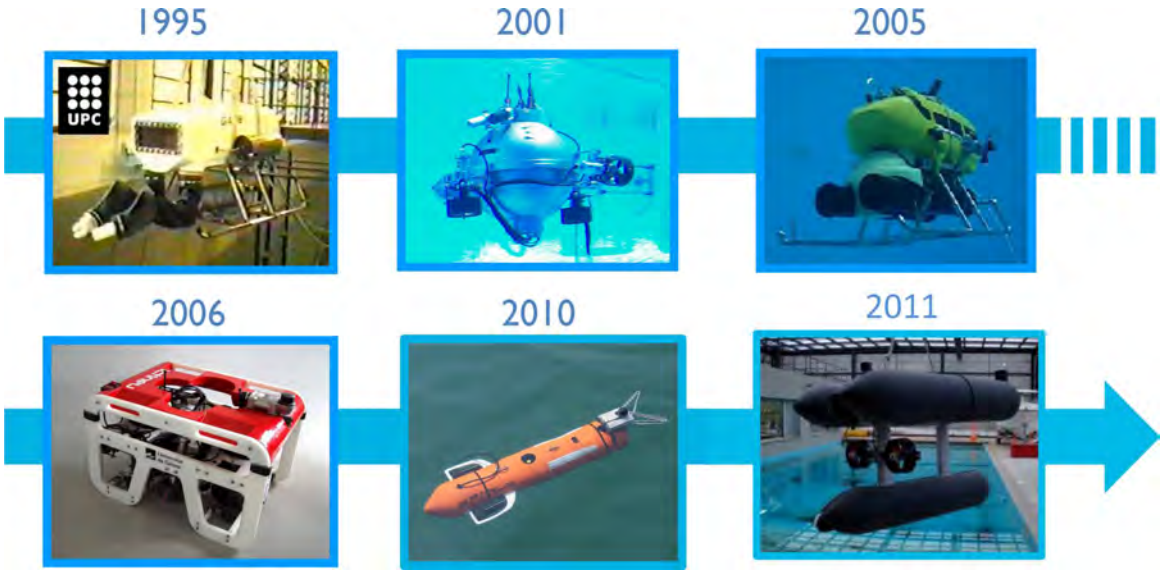


Figure 1.2: Prototypes developed by the VICOROB team. From left to right: GARBI-ROVfor, URIS, GARBI-AUV, ICTINEU, SPARUS and GIRONA-500. [Source: VICOROB]

between landmarks and the vehicle itself, which is the equivalent to human's senses.

In conclusion, underwater research is high-level research that produces significant technological advances as well as medical and scientific discoveries, which then become part of our daily lives, and contribute to the improvement of the quality of life.

1.2 Context

The increasing trend towards the study and research of underwater technology has been mainly motivated by the growing interest from oil and defense industries. Not to mention the enormous benefits that could be accomplished if humans managed to develop a safe an reliable technology for the exploration and exploitation of the ocean floor. In this context the Computer Vision and Robotics group (VICOROB) at the Universitat de Girona (UDG) is carrying out research in underwater robotics and underwater vision. The underwater robotics laboratory is responsible for the study of control architectures for autonomous robots, identification and modeling of underwater vehicles operating environment, design and development of vehicle simulators, missions in underwater environments, fusion of information from different sensors for navigation, location of vehicles and construction of visual maps of submerged structures, all of which involves the development and commissioning of underwater robots (see Figure 1.2). The Research Centre in Underwater Robotics (CIRS) is the building that houses the underwater robotics laboratory (see Figure 1.3). This building is located in the Scientific and Technological Park of the UDG. The complex is composed

Figure 1.3: The CIRS research facility. In the left part of the complex, workshops and offices. In the right part, the water tank and control room. [Source: VICOROB].

of two main buildings (see Figure 1.4); one contains laboratories, offices and workshops, the other contains a water tank and a supervision control room with a direct view of the water tank. The underwater vision laboratory focuses on the development of systems for the construction of geo-referenced underwater mosaics, including the construction of large scale maps, as well as 3D reconstruction of the seabed. VICOROB participates in several projects funded by the *Comisión Interministerial de Ciencia y tecnología (CICYT)* in its national R & D plans and by the European Union (EU) Framework Programmes.

Some examples of underwater technologies research financed by Comisión Interministerial de Ciencia y tecnología (CICYT) at VICOROB are: AIRSUB - Industrial Applications of underwater robotics (DPI2005-09001-C03-01), RAUVI – Reconfigurable AUV for Intervention Missions (DPI2008-06545-C03-03) (see Figure 1.2), ACUAVISION – Computer Vision systems for underwater mapping and aquaculture (DPI2007-66796-C03-02), FOTOGEO – Development of a modular system to build georeferenced photomosaics of the ocean floor (CTM2004-04205), and an Autonomous Robot for Environmental Monitoring (CTM2007-64751/MAR).

Furthermore, EU is funding robotics R & D as part of the so called Seventh Framework Programme, which stresses the need for collaborative research and international cooperation. Some examples of underwater robotics research at VICOROB financed by the EU are the MOMARNET – Monitoring deep seafloor hydrothermal environments on the Mid-Atlantic Ridge and the TRIDENT – Marine Robots and Dexterous Manipulation for Enabling Autonomous Underwater Multi-purpose Intervention Missions. In addition, the group is part of the Initial Training Network (ITN), FreeSubNet – a European research training network on key technologies for intervention Autonomous Underwater Vehicles, along with several European research groups, including the Ocean Systems Laboratory (OSL) at Heriot Watt University (HWU).

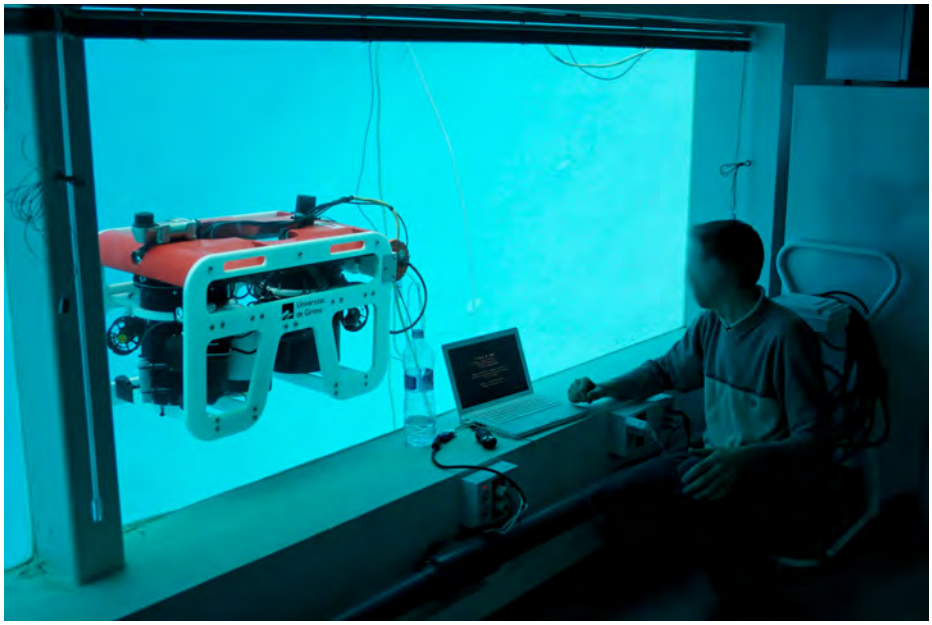
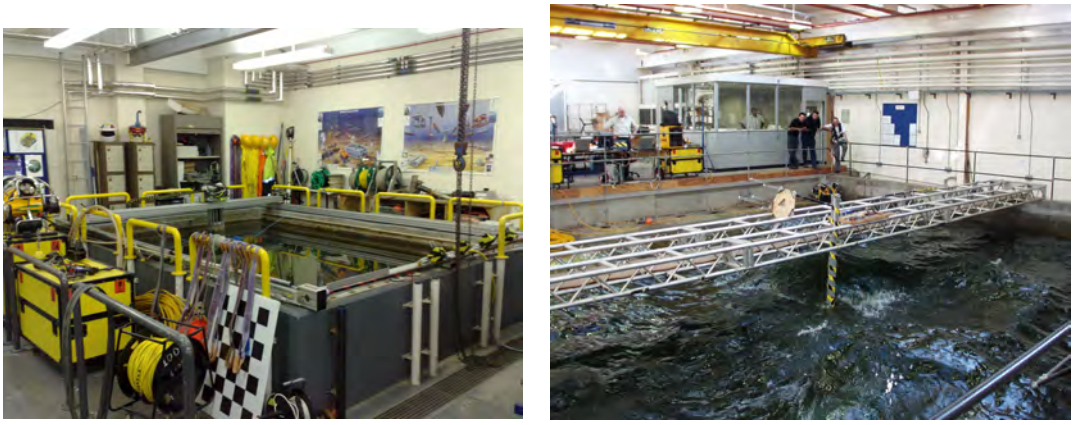


Figure 1.4: Control room at CIRS with direct views to the water tank. [Source: VICOROB].

OSL is within the Joint Research Institute (JRI), together with the Vision, Image and Signal Processing (VISP) group from HWU. Both groups work to develop computer vision and underwater robotics. OSL and VISPhave been involved in this thesis. The OSL team developed their own underwater vehicles and also have a Remote Environmental Measuring UnitS (REMUS) AUV. Besides the vehicles, OSL has two water tanks (see Figure 1.5): one capable of generating waves (see Figure 1.5(b)) and simulate realistic environmental conditions. The OSL team closely collaborates with underwater technology companies such as SeeByte, Subsea7 and others, and has a great international reputation for consulting in important areas, such as transport, offshore and underwater structures, and defense.

Good examples of the experience acquired by VICOROB and OSLin underwater robotics are the vehicles they developed for the Student Autonomous Underwater Challenge - Europe (SAUC-E). SAUC-E's main objective is to promote research and development of underwater technology. VICOROB won the competition with the ICTINEU in 2006 and SPAUS in 2010 (see Figure 1.2). The OSL team won the competition with NESSIE III in 2008 and NESSIE IV in 2009, while NESSIE VT made it to the final in 2010 (see Figure 1.6).

The expertise of both groups together with the needs of their projects have been the ideal context and motivation for developing this thesis.



(a) Test Tank. (b) Cartesian plotter wave tank.

Figure 1.5: Facilities available at OSL. [Source: OSL]

1.3 Objectives Statement

The on-board sensors of an autonomous vehicle provide information about the vehicle and about the environment. Fusing this information by means of filtering techniques can provide the vehicle's state and map estimates, together with its uncertainties. Therefore, the basic objective of this thesis is **to enable large area 2D and 3D localization and mapping from sensor readings acquired with robotic vehicles**.

Given a sequence of measurements acquired from an autonomous vehicle, techniques allowing simultaneous localization and mapping are studied, designed and implemented. In order to achieve this objective, it is necessary to develpe auxiliary techniques that can:

- generate localization and map estimates that are globally consistent,
- provide uncertainty estimates for both the localization and the map,
- use navigation and sensor data to constrain the map and vehicle localization,
- reliably extract and match features,
- scale to hundreds or thousands of landmarks, long term missions and large areas,
- yield additional benefits such as developing a system useful for different autonomous vehicles (i.e, terrestrial, aerial and underwater).

1.4 Thesis Outline

The material presented in this thesis is structured as follows:

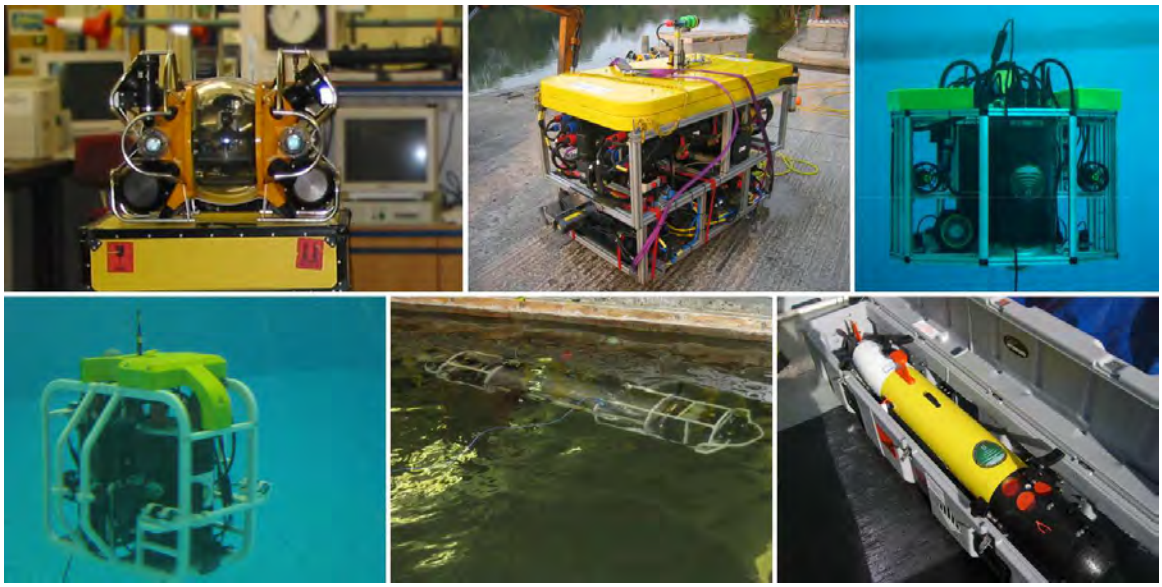


Figure 1.6: Experimental capabilities of the OSL. From top-left to bottom-right: Hyball, Rauver, Nessie III, Nessie IV, Nessie VT and REMUS-100 AUV. [Source: OSL].

- **Chapter 2** presents the state of the art of the main topics involved in this work. First, filtering techniques are analysed and compared. Then, special attention is given to SLAM approaches that tackle the problem by dividing the map into small submaps. Afterwards, the most significant underwater SLAM projects are introduced. Finally, underwater imaging algorithms are briefly summarized.

- **Chapter 3** describes the basis behind SSJS. First, a probabilistic background is given to introduce the reader to the topic. Then, the SSJS is described step by step. Finally, a comparison between several submapping approaches is given in order to assess the performance of SSJS.

- **Chapter 4** focuses on the description of feature and matching approaches. Three different types of sensors are used in the experiments presented in this work: a laser range finder, a side-scan sonar and an optical camera. In this section, algorithms to extract features and match them are presented and validated for each of these three sensors.

- **Chapter 5** presents SSJS implementations for two different AUVs: REMUS-100 and SPARUS. The former uses a side-scan sonar as the remote sensing device, while the later uses an optical camera. Both implementations are detailed in this section together with experimental results.

- **Chapter 6** gives conclusions. A list of publications and scientific collaborations is

given. Finally, future perspectives introduced by this work are commented on.

- **Appendix A** describes 2D and 3D transformations commonly used in stochastic mapping.

Chapter 2

State of the Art

Contents

A solution to the SLAM problem has been seen as a "holy grail" by the mobile robotics community as it would provide the means of making a robot truly autonomous. For this reason over the last two decades several research groups have been working on different approaches. A brief overview of SLAM is given in Section 2.1. In section 2.2, the most significant SLAM filtering approaches are briefly described, denoting their advantages and disadvantages. Section 2.3 presents existing submapping SLAM strategies. Section 2.4 summarizes underwater SLAM approaches. Finally, a survey of underwater imaging is given in Section 2.5. A general classification is given at the end of each section comparing the surveyed techniques.

2.1 Overview

Robotic platforms are gaining importance in scientific, industrial, defense, and transportation applications. These robotic platforms demand autonomous localization solutions. For this reason, SLAM algorithms have been extensively researched over the last two decades.

SLAM is a process by which a mobile robot can build a map of an environment and at the same time use this map to deduce its location. Initially, both the map and the vehicle position are unknown, the vehicle has a known kinematic model and is moving through the unknown environment populated by artificial and/or natural landmarks. A simultaneous estimate of both robot and landmark locations is required. The SLAM problem involves finding appropriate representation for both the observation and the motion models [Durrant-Whyte 2006, Bailey 2006]. In order to do this, the vehicle must be equipped with sensors capable of acquiring measurements of the relative location between landmarks and the vehicle itself. Figure 2.1 shows a conceptual idea behind the SLAM problem.

Several research groups have worked and are currently working on SLAM. Research groups that have made a significant contribution to SLAM are listed below:

- *Active Vision Group, University of Oxford*[1] (Andrew Davison, David Murray and Ian Reid among others).
- *Robotics Research Group, University of Oxford*[2] (Paul Newman among others).
- *Robotics Perception and Real-time Group, University of Zaragoza*[3] (José A. Castellanos, Juan D. Tardós, José Neira and José M. Martinez-Montiel among others).
- *Department of Ocean Engineering at MIT, Computer Science and Artificial Intelligence Laboratory*[4] (John Leonard, Mathew Walter among others)
- *Australian Center for Field Robotics, University of Sidney*[5] (Stefan William, Eduardo Nebot, Oscar Pizarro among others)
- Other important names in the context of SLAM are: Sebastian Thrun at the Stanford University, Wolfram Burgard at the University of Freiburg and Dieter Fox at the University of Washington.

The most commonly used sensors on autonomous robotics can be categorized into laser-based, sonar-based, and vision-based systems. Additional sensors such as, compasses, infrared technology and Global Positioning System (GPS), are used to improve perception of robot conditions and the outside world [Thrun 2002], . However, all these sensors are subject to errors, often referred to as measurement noise, and range limitations making it necessary to navigate through the environment; for example, light and sound cannot penetrate walls.

Laser ranging systems are accurate active sensors [Guivant 2000b, Grisetti 2005]. Their most common configuration operates on the time of flight principle by sending a laser pulse

[1] http://www.robots.ox.ac.uk/ActiveVision/
[2] http://www.robots.ox.ac.uk/
[3] http://webdiis.unizar.es/GRPTR/
[4] http://www.csail.mit.edu/index.php
[5] http://www.acfr.usyd.edu.au/about/index.shtml

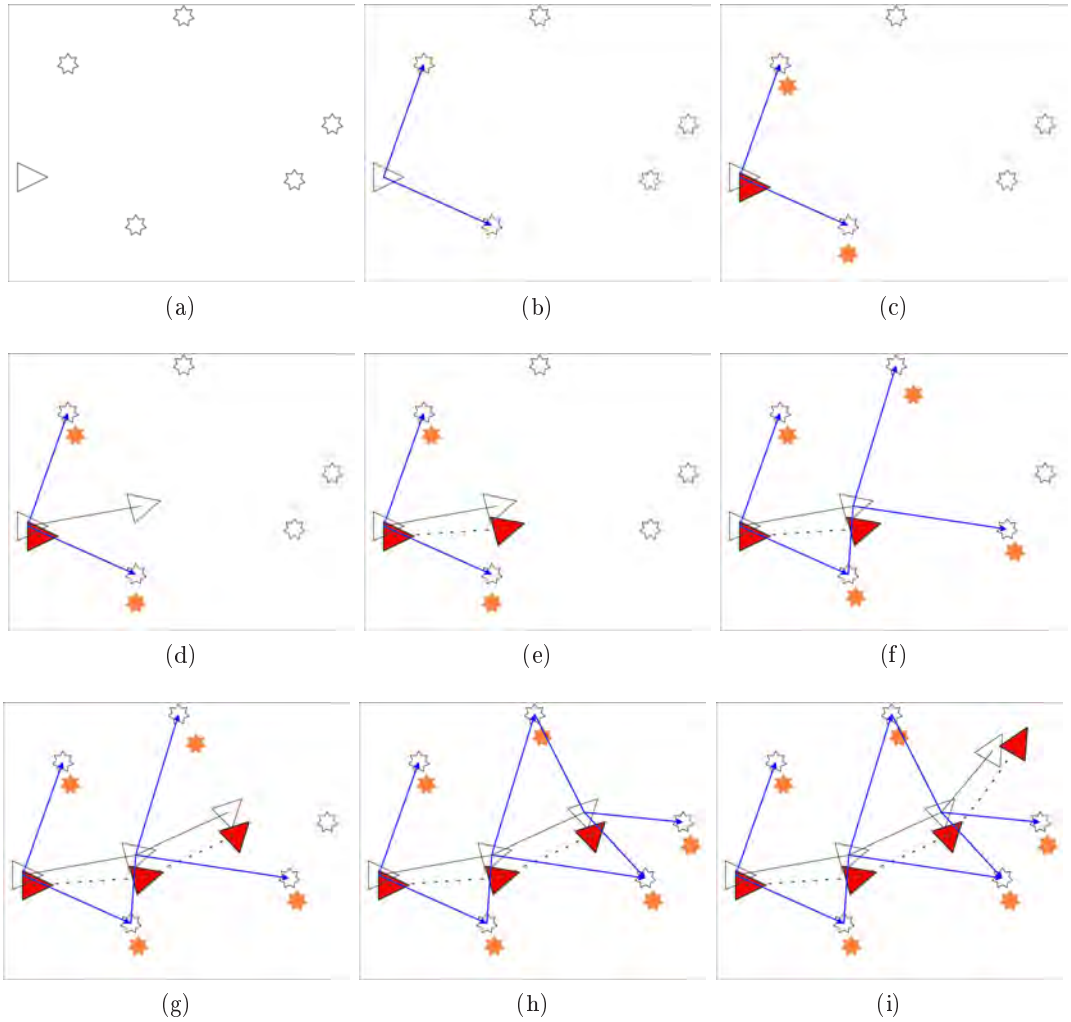


Figure 2.1: Conceptual basis of SLAM. Triangles are the vehicle's position and stars are the landmarks. In red and orange the estimates, while the true positions are left empty. In a) the vehicle starts a mission in an unknown scenario populated by several landmarks. b) On-board sensors produce noise measurements of the scenario. c) The relative position between the landmarks and the vehicle is estimated from the observation. d) The vehicle starts moving and e) an estimate of its new position is computed. f-g) The vehicle enters into a continuous operation of moving through the scene and observing it. h). If a known landmark is observed for the second time, i) both the map and the vehicle's position are corrected.

in a narrow beam towards the object and measuring the time taken by the pulse to be reflected off the target and returned to the sender. *Sonar-based systems* are fast and provide measurements and recognition capacities similar to vision [Ribas 2008b, Mahon 2004]. However, their dependence on inertial sensors such as odometers implies that a small error can have a great effect on later position estimates [Thrun 2002]. Finally, *Vision sys-*

tems are passive [Se 2002, Davison 2007, Jensfelt 2006]. They have a long range and high resolution, but the computation cost is prohibitively high and good visual features are more difficult to extract and match. Vision is used to estimate the 3D structure, feature location and the robot position, by means of stereo camera pairs for instance [Salvi 2008, Mei 2009, Johnson-Roberson 2010] or monocular cameras with the structure from the motion recovery [Nicosevici 2008, Civera 2010]. All these sensors produce large amounts of data, making the use of data mining and computer vision algorithms to visualize and interpret the data necessary.

Further classification can be made in terms of the working environment, for instance, indoor ground, outdoor ground, airborne or underwater. Most of the work done so far has focused mainly on inddor ground environments [Leonard 2003, Estrada 2005, Newman 2006, Davison 2007], with only a few papers dealing with airborne applications [Kim 2007, Artieda 2009] and even fewer that present underwater SLAM, and these generally work with acoustic data [Leonard 2001, Williams 2004, Fairfield 2007, Ribas 2008b]. In these latter cases, vision plays an important role in underwater SLAM approaches [Eustice 2008, Sáez 2006, Johnson-Roberson 2010], mostly combined with other sensory systems to acquire both depth and feature information of the scene, i.e. acoustic or inertial sensors.

This chapter presents a brief survey and lists some of the most representative work on SLAM. Afterwards, the most significant underwater SLAM approaches are summarized. Finally, computer vision algorithms commonly used in SLAM to visualize and interpret the data are surveyed.

2.2 A Review of SLAM Techniques

This section presents an overview of the most relevant contributions to SLAM filtering techniques. This overview does not intend to be a complete review of all the publications in the literature, it merely presents the main authors and approaches and finally a summary of the strengths and weaknesses of these projects in Table 2.1.

Robotic map-building can be traced back 25 years and since the 1990s, probabilistic approaches have become dominant. The representation and estimation of spatial uncertainty [Smith 1986, Smith 1988] and the creation of the statistical basis to describe spatial relationships between the vehicle and surrounding landmarks [Durrant-Whyte 1988] is considered to be the origin of SLAM. During the 1990s, many researchers started to consider the localization problem and the mapping problem as one. [Smith 1990] demonstrated that by joining both problems, a consistent solution is possible due to the correlations that exist between the vehicle and the landmarks, while [Leonard 1991] proved the joint solution to be not only consistent, but also convergent. The convergence is given by the correlations

between the vehicle and the landmarks, because every time a landmark is observed the relative position between this landmark and the vehicle is updated. These correlations make the SLAM problem computationally expensive when dealing with a large number of landmarks, i.e., large scenarios and long missions. In addition, filtering algorithms are necessary to deal with noisy measurements. Kalman Filter (KF) Particle Filter (PF) and Expectation Maximization (EM) are the main probabilistic approaches being used in SLAM. The three techniques are mathematical derivations of the recursive Bayes rule. The main reason for this probabilistic technique's popularity is the fact that robot mapping is characterized by uncertainty and sensor noise. Probabilistic algorithms undertake the problem by explicitly modeling different sources of noise and their effects on the measurements [Thrun 2002].

General SLAM terms are first introduced in order to ease the comprehension of this review.

- **State vector** is a $n \times 1$ column vector that describes the position, orientation and other properties of the vehicle. This vector also includes landmark information. The size of the vector grows with the number of landmarks.
- **Covariance matrix** is a $n \times n$ matrix that contains the uncertainty associated to each of the state's vector components.
- **Data association** is the process by which the system has to determine whether a detected landmark corresponds to a previously seen landmark or to a new one.
- **Loop closing** is the task of deciding whether or not a vehicle has, after an excursion of arbitrary length, returned to a previously visited area.
- **Motion model**, also called process model, is the mathematical formulation that describes the dynamics of the vehicle.
- **Observation model** is the mathematial representation of a sensor measruement.

2.2.1 Kalman Filter and its Variations (KF)

KF are Bayes filters that represent posteriors using Gaussian distributions, i.e. unimodal multivariate distributions that can be represented compactly by a small number of parameters. KF SLAM relies on the assumption that the state transition and the measurement functions are linear with added Gaussian noise. The initial posteriors are also Gaussian. There are two main variations of KF in state of the art SLAM: Extended Kalman Filter (EKF) and its related Information Filter (IF) or Extended Information Filter (EIF). EKF accommodates the non-linearities from the real world by approximating the robot's motion model using linear functions. These approximations make EKF a suboptimal solution [Castellanos 2007] increasing the chance of having some divergences. Several existing SLAM approaches use

EKF [Tardós 2002, Leonard 2003, Jensfelt 2006, Davison 2007, Ribas 2008b, Piniés 2009]. An IF is implemented by propagating the inverse of the state error covariance matrix. There are several advantages of an IF filter over a KF. Firstly, the data is filtered by simply summing information matrices and vectors, providing more accurate estimates [Thrun 2003]. Secondly, an IF is more stable than a KF [Thrun 2004b]. Finally, an EKF is relatively slow when estimating high dimensional maps, because every single vehicle measurement generally effects all the parameters of the Gaussian, therefore updates require prohibitive times when dealing with environments with many landmarks [Thrun 2004a].

However, IFs have some important limitations. One primary disadvantage is the need of recovering a state estimate in the update step when applied to non-linear systems. This step requires the inversion of the information matrix. Further matrix inversions are required for the prediction step of these information filters. For high dimensional state spaces the need of computing all these inversions is generally believed to make an IF computationally inferior to a KF. In fact, this is one of the reasons why EKFs have been vastly more popular than EIFs [Thrun 2005]. These limitations do not necessarily apply to problems in which the information matrix possesses structure. In many robotics problems, the interaction of state variables is local; as a result, the information matrix may be sparse. Such sparseness does not translate to sparseness of the covariance. Information filters can be thought of as graphs where states are connected whenever the corresponding off-diagonal element in the information matrix is non-zero. Sparse information matrices correspond to sparse graphs. Some algorithms exist to perform basic updates and estimation equations efficiently for such fields [Walter 2005, Walter 2007], in which the information matrix is (approximately) sparse and allow the development of an extended information filter that is significantly more efficient than both a KF and a non sparse IF.

An Unscented Kalman Filter (UKF) [Wan 2001] addresses the approximation issues of an EKF and the linearity assumptions of a KF. A KF performs properly in linear cases and is accepted as an efficient method for analytically propagating a Gaussian Random Variable (GRV) through a linear system dynamics. For non-linear models, an EKF approximates the optimal terms by linearising the dynamic equations. An EKF can be viewed as a first-order approximation to the optimal solution. In these approximations the state distribution is approximated by a GRV, which then is propagated analytically through the first-order linearization of the non-linear system. These approximations can produce important errors in the true a-posteriori mean and covariance, which may sometimes lead to divergence of the filter. In an UKF the state distribution is again represented by a GRV, but is now specified using a minimal set of carefully chosen sample points. These sample points capture the true mean and covariance of GRVcompletely, and when propagated through the true non-linear system, capture the a-posteriori mean and covariance accurately to the third order for any non-linearity. In order to do that, the unscented transform is used.

One of the main drawbacks of EKF and KF implementations is the fact that for long duration missions, the number of landmarks will increase and, eventually, computer resources will not be sufficient to update the map in real-time. This scaling problem arises because each landmark is correlated to all other landmarks. The correlation appears since the observation of a new landmark is obtained with a sensor mounted on the mobile robot and thus the landmark location error will be correlated with the error in the vehicle location and the errors in other landmarks on the map. This correlation is of fundamental importance for the long-term convergence of the algorithm and needs to be maintained for the full duration of the mission. The Compressed Extended Kalman Filter (CEKF) algorithm [Guivant 2001] significantly reduces the computational requirement without introducing any penalties in the accuracy of the results. A CEKF stores and maintains all the information gathered in a local area with a cost proportional to the square of the number of landmarks in the area. This information is then transferred to the rest of the global map with a cost that is similar to full SLAM but in only one iteration.

The advantage of a KF and its variants is that they provide optimal Minimum mean-square Error (MMSE) estimates of the state (robot and landmark positions) and its covariance matrix seems to converge strongly. However, the Gaussian noise assumption restricts the adaptability of KFs for data association and number of landmarks.

2.2.2 Particle Filter (PF) Based Methods

PFs, also called Sequential Monte-Carlo (SMC), are recursive Bayesian filters used in Monte Carlo simulations. They execute SMC estimation with a set of random point clusters (particles) representing the Bayesian posterior. In contrast to parametric filters (e.g., a KF), a PF represents the distribution by a set of samples drawn from this distribution, what makes it capable of handling highly non-linear sensors and non-Gaussian noise. However, this ability produces a growth in computational complexity on the state dimension as new landmarks are detected, becoming unsuitable for real time applications [Montemerlo 2002]. For this reason, PFs have only been successfully applied to localization, i.e. determining position and orientation of the robot, but not to map-building, i.e. landmark position and orientation; therefore, there are no important papers using PFs for the whole SLAM framework, but there are a few projects that deal with the SLAM problem using a combination of PFs with other techniques; for instance, the FastSLAM [Montemerlo 2002] and the fastSLAM2.0 [Montemerlo 2003]. FastSLAM takes advantage of an important characteristic of the SLAM problem (with known data association): landmark estimates are conditionally independent given the robot's path [Montemerlo 2007]. The FastSLAM algorithm breaks the SLAM problem down into a robot localization problem and a collection of landmark estimation problems that are conditioned on the robot position estimate. A key characteristic of FastSLAM is that each particle makes its own local data association. In contrast,

EKF techniques must commit to a single data association hypothesis for the entire filter. In addition, a FastSLAM uses a particle filter to sample various robot paths which requires less memory usage and computational time than a standard EKF or KF.

2.2.3 Expectation Maximization Based Methods

An EM estimation is a statistical algorithm developed in the context of Maximum Likelihood (ML) estimation and offers an optimal solution resulting in an ideal option for map-building. An EM algorithm is able to build a map when the robot's position is known, for example, by means of expectation [Burgard 1999]. An EM iterates two steps: an expectation step (E-step), where the a-posteriori probability over robot position is calculated for a given map, and a maximization step (M-step), in which the most likely map is calculated given this position expectations, resulting in a series of increasingly accurate maps. The main advantage of EMs with respect to KFs is that they can deal with the correspondence problem (data association problem) surprisingly well [Thrun 2002]. This is possible thanks to the fact that they repeatedly localize the robot relative to the present map in the E-step, generating various hypotheses as to where the robot might have been (different possible correspondences). In the latter M-step, these correspondences are translated into features in the map, which then get reinforced in the next E-step or gradually disappear as the case may be. However, the need to process the same data several times to obtain the most likely map makes it inefficient, non-incremental and unsuitable for real-time applications [Chen 2007]. Even using discrete approximations, when estimating the robot's position, the cost grows exponentially with the size of the map and the error is not bounded; hence the resulting map becomes unstable after long cycles. These problems could be avoided if the data association was known [Thrun 2001] and if the E-step were simplified or eliminated. For this reason, an EM usually is combined with a PF, which represents the a-posteriori probabilities by a set of particles (samples) that represent a guess of where the robot might be. For instance, some practical applications use EMs to construct the map (only the M-step), while the localization is done by different means, i.e. using a PF-based localizer to estimate positions from odometer readings [Thrun 2002].

2.2.4 Classification: Pros and Cons

Table 2.1 provides a list of advantages and disadvantages of different SLAM strategies in terms of the method used to deal with uncertainties. EKF produces a convergent solution and handles uncertainties. However, it produces a suboptimal solution due to gaussianity assumption and linearization approximations. In addition, it suffers a significant increment in computational cost as the map grows. A couple of variations of KF are CEKF and UKF. The former addresses computational cost issues, but it still suffers from linearization errors;

while the later solves the linearization problem but at a higher cost. An IF has shown itself to be fast for long missions, but it does not carry any uncertainty information which means that data association becomes much more complex. PFs and EMs seem to address some of the issues in EKFs. However, the former seem to be only successful for localization problems, while the later are only optimal in map building. From this survey, the decision towards using an EKF was made. EKF has been selected, because despite its weaknesses, it has been proved to converge and handles uncertainties, i.e. data association becomes easier. Furthermore, several approaches address the linearization errors and the computational cost by dividing the map into submaps. These techniques are surveyed in the following section.

Table 2.1: List of strengths and weaknesses of filtering approaches used in SLAM.

Pros	*Cons*
Kalman Filter and Extended KF (KF/EKF) [Davison 2002, Leonard 2003, Jensfelt 2006, Ribas 2008b]	
- high convergence	- Gaussian assumption
- handles uncertainty	- computationally expensive in high-D
	- linearization might cause divergences
Compressed Extended KF (CEKF) [Guivant 2001]	
- reduces uncertainty	- requires very robust features
- reduces memory usage	- data association issues
- handles large areas	- requires multiple map merging
- increases map consistency	
Unscented Kalman Filter (UKF) [Wan 2001]	
- robust and accurate estimation	- complexity dependent on number
- no need for Jacobians	of samples
Information Filters (IF) [Thrun 2003, Thrun 2004b]	
- stable and simple	- data association issues
- accurate	- cubic complexity to recover covariances
- fast for large maps	- in high-D is computationally expensive
Particle Filter (PF) [Montemerlo 2002, Montemerlo 2003]	
- handles non-linearities	- exponential growth in complexity
- handles non-Gaussian noise	- only successful in localization
- multiple hypotheses	
Expectation Maximization (EM) [Burgard 1999, Thrun 2004b]	
- optimal to map building	- inefficient, cost growth
- solves data association	- unstable for large scenarios
	- only successful in map building

2.3 A Review of Submapping SLAM Approaches

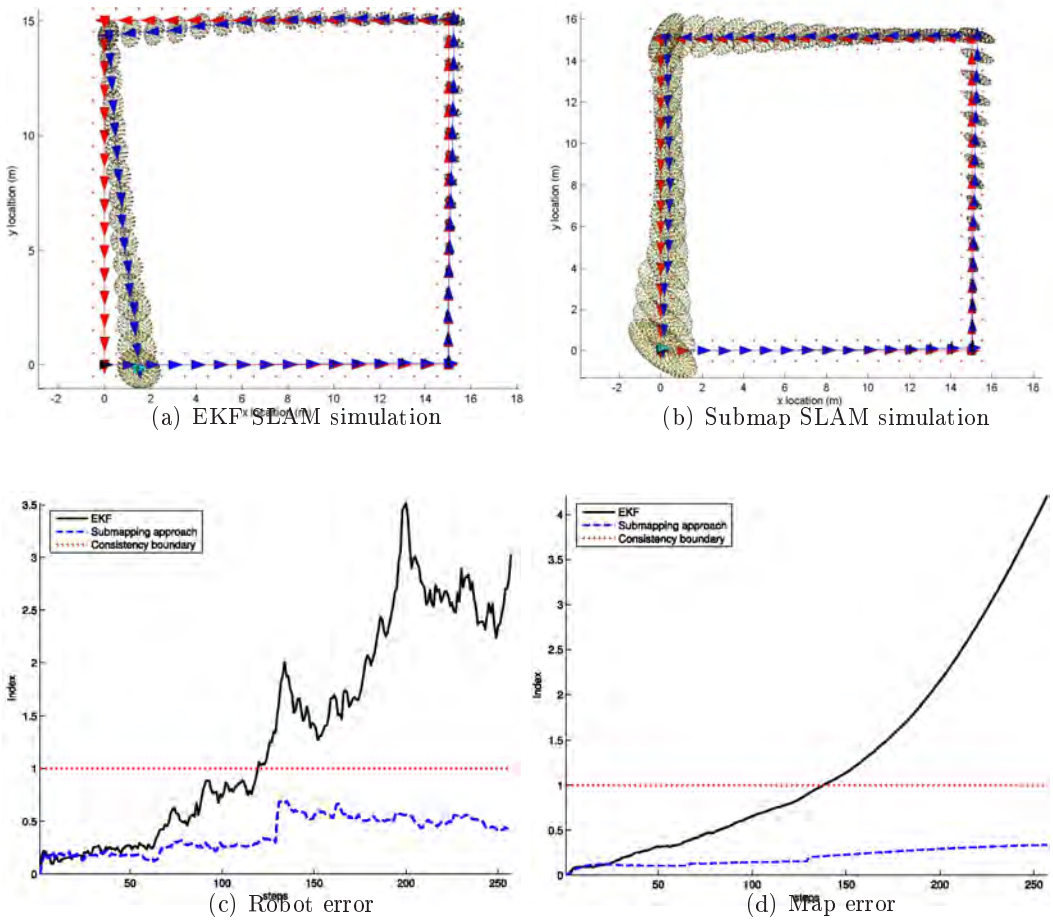
This section presents an overview of the most relevant submapping SLAM approaches. This review does not intend to list all publications in the field, but will present the main authors and approaches. Finally a summary of these works is given in Table 2.2.

The cost of updating the covariance matrix at each step limits the use of an EKF SLAM in large environments and the effect of linearization in the final vehicle and landmark estimates reduces the consistency. These considerations can be easily observed in Figure 2.2, in which the inconsistency rate keeps growing during the process in the EKF case, while it stays under a certain boundary in the multi-map case. In large areas, the EKF complexity grows with the number of landmarks, because each landmark is correlated to all other landmarks. Thus, the EKF memory complexity is $O(n^2)$ with a time complexity of $O(n^2)$ per step, where n is the total number of features stored in the map.

Several researchers have proposed ways of tackling the issues associated with EKF based SLAMs in large areas [Pizarro 2004]. In terms of computational complexity, in [Guivant 2001, Knight 2001, Guivant 2002b, Folkesson 2003] the authors propose delaying the global update stage after several observations, thereby significantly reducing the costs. Regarding map consistency, the UKF[Wan 2001] achieves better consistency when addressing EKF approximation and assumption issues because it does not need linearization. However, the UKF is computationally expensive. Other approaches reduce computational costs by taking advantage of the sparsity structure of the inverse of the covariance matrix. These are the so-called IF techniques [Thrun 2004a, Eustice 2006b, Walter 2007]. However, IF techniques have problems with data association since no covariance matrix is involved in the process.

The use of submaps has been shown to address both linearization errors and computational costs at the same time, thereby improving the consistency of EKF based SLAMs [Castellanos 2007]. An early example of this strategy is the Decoupled Stochastic Mapping (DSM) [Leonard 2001], which uses non-statistically independent submaps. As a result, correlations are broken and inconsistency is introduced into the map. The Constant Time SLAM (CTS) [Newman 2003] uses multi overlapping local submaps with the frame referenced to one of the features in the submap. This technique maintains a single active map and computes a partial solution independently. However, in non-linear cases the consistency is not proven. Different techniques, such as the Constrained Local Submap Filter (CLSF) [Williams 2002] or Local Map Joining (LMJ) [Tardós 2002] produce efficient global maps by consistently combining completely independent local maps. The main idea behind LMJ is to build maps of limited size and then, once completed, merge these small maps into a global one.

The Atlas SLAM [Bosse 2004] consists of a hierarchical strategy that achieves efficient



(a) EKF SLAM simulation (b) Submap SLAM simulation

(c) Robot error (d) Map error

Figure 2.2: Normal EKF SLAM solution vs. submapping approach [Paz 2007b]. On top: loop closing scenario, ground true trajectory in red, estimated trajectory in blue, and ellipses representing uncertainty. Bottom: mean consistency index EKF SLAM (black) and submapping SLAM (blue).

mapping of large-scale environments. They use a graph of coordinate frames with each vertex in the graph representing a local frame and each edge representing the transformation between adjacent frames. In each frame, they build a map that captures the local environment and the current robot position along with the associated uncertainties.

The Divide and Conquer SLAM (DCS) [Paz 2008] is capable of recovering a global map in approximately $O(n)$ time, by using the Divide and Conquer strategy from fundamental graph theory. An example of its working principle is illustrated in Figure 2.3.

The Hierarchical SLAM (HS) [Estrada 2005] consists of the lower (or local) map level, which is composed of a set of local maps that are guaranteed to be statistically independent, and the upper (or global) level, which is an adjacency graph whose arcs are labeled with the

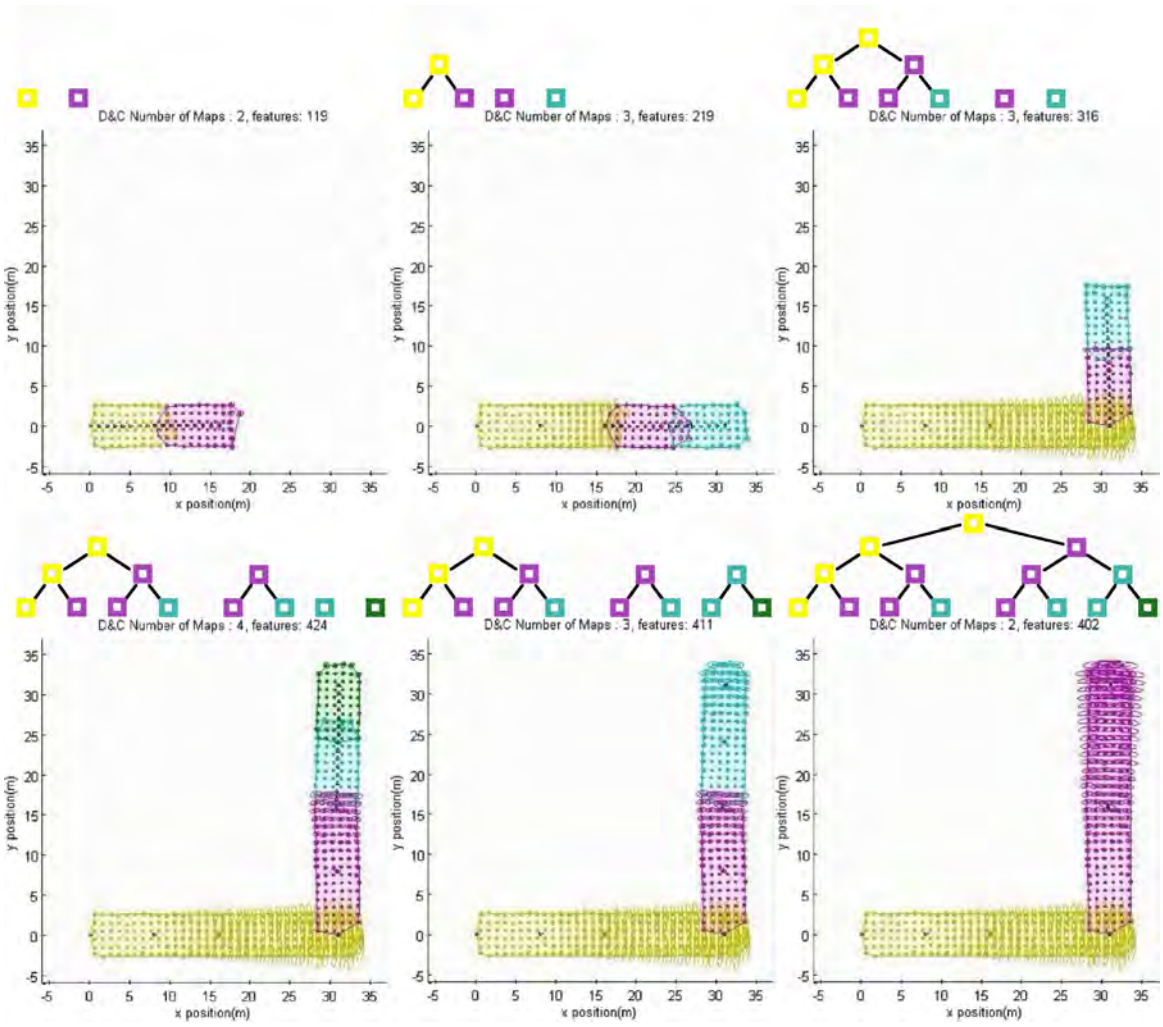


Figure 2.3: A Divide and Conquer SLAM example. From top-left to bottom-right: submaps are sequentially built and joined following the three structures plotted on top of each step.

relative location between local maps. An estimate of these relative locations is maintained at this level in a relative stochastic map. Every time the vehicle closes a loop, a global level optimization is performed, producing a better estimate of the whole map, as shown in Figure 2.4.

A Conditionally Independent SLAM (CIS) [Piniés 2008, Piniés 2009] is based on sharing information between consecutive submaps so that, a new local map is initialised with *a-priori* knowledge. The Bayesian Network that describes the probabilistic dependencies between submap variables is shown in Figure 2.5.

Table 2.2 compares memory complexity, computational time during local map building, and computational time during loop closing. The main conclusion one can extract from this

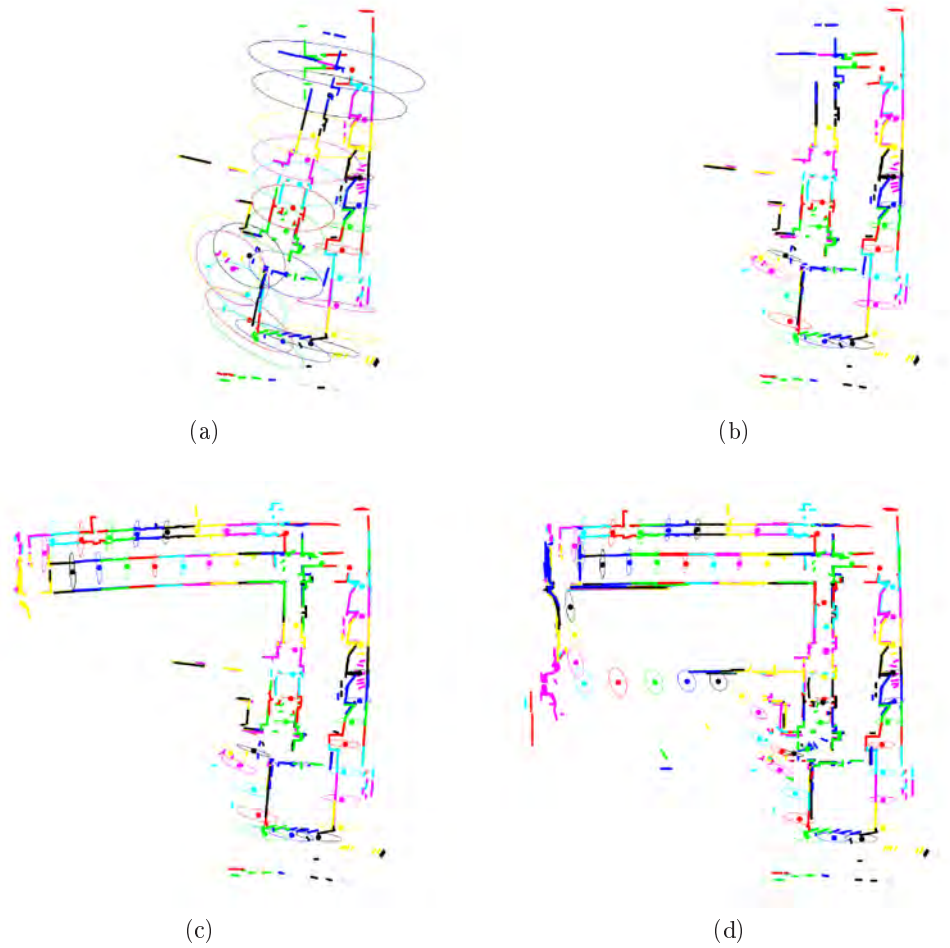


Figure 2.4: A Hierarchical SLAM example [Estrada 2005]. This figure depicts the evolution of the whole map during the mission. Submaps are painted in different colours. Every ellipse corresponds to the uncertainty of the link between two consecutive submaps (i.e., the uncertainty in the global level). a) The vehicle will close a loop for the first time; b) the vehicle has just closed this first loop and the map has been corrected; c) after closing a second loop; d) after closing a third loop.

table is the significant difference between a standard EKF and submapping approaches in terms of time consumption and memory usage. It is worth mentioning that recent approaches [Estrada 2005, Piniés 2009] are those with higher performance, i.e. lower computational demand.

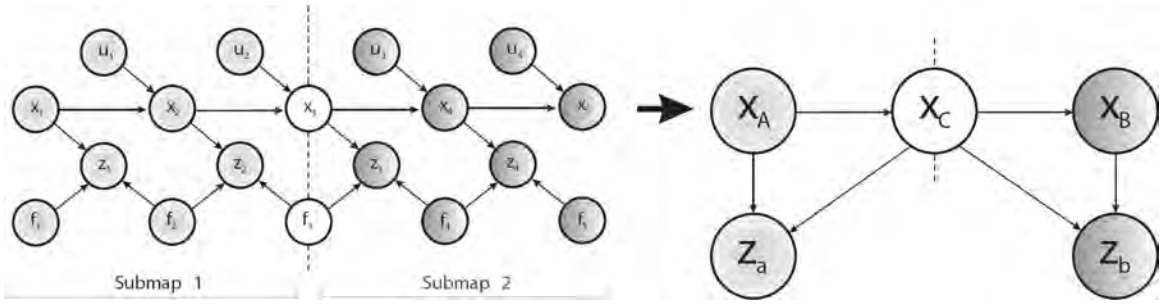


Figure 2.5: Conditional Independent SLAM [Piniés 2008]. The graph on the left describes a map in which the vehicle has moved from x_1 to x_5. Five features $f_{1:5}$ have been observed during the trajectory. Control inputs $u_{1:5}$ and observations $z_{1:4}$ are given. In this example, submap 1 starts at x_1 and finishes at x_3 and has observed features $f_{1:3}$ through measurements $z_{1:2}$. Afterwards, the submap 2 is initialized with a common feature f_3 and the last known vehicle position x_3. The vehicle then moves and traverses two new positions $x_{4:5}$, re-observes feature f_3 and indirectly x_3 through z_3 and observes two new features $f_{4:5}$ through measurements $z_{3:4}$. The graph on the right shows a simplification of the one on the left that makes it easier to see that the only connection between the set of nodes (x_A, z_a) and (x_B, z_b) is through node x_C, i.e. given x_C submap A and submap B are conditionally independent.

2.4 A Review of Underwater SLAM

This section presents an overview of the most relevant underwater SLAM implementations. This review does not enumerate all existing publications in underwater SLAM, but it gives an idea of the main authors and approaches. Afterwards, a summary is given in Table 2.3. A key issue for an AUV to navigate fully autonomously is the ability to localize itself. The localization problem has been analysed from different viewpoints: 1) assuming an a-priori known environment map, and 2) assuming an a-priori unknown environment.

Several approaches tackle the localization problem on known scenarios. For instance, some approaches use GPS-aided localization [Caiti 2005, Erol 2007], but the attenuation of electromagnetic waves through the medium of water limits the application of GPS to near surface activities, otherwise the vehicle is forced to pay frequent visits to the surface to recover its position. A standard for bounded xyz navigational position measurements for underwater vehicles is the Long-BaseLine (LBL) acoustic transponder system [Hunt 1974, Olson 2006]. The equivalent to GPS underwater are the acoustic transponders, such as an LBL or a Short-BaseLine (SBL). These positioning systems have limited range but high accuracy and an associated cost of deployment. An LBL operates on the principle of time-of-flight and has been proven to operate up to a range of 10 km [Whitcomb 1999]. The main drawback of an LBL is that it requires two or more acoustic transponder beacons to be tethered to the sea floor. SBL systems provide more accurate positioning information, but suffer from

Table 2.2: Summary of submapping SLAM approaches.

Method	Memory	T local	T Loop
EKF [Smith 1986] + High convergence – Computationally expensive	$O(n^2)$	$O(n^2)$	$O(n^2)$
CEKF [Guivant 2001] + Faster than EKF – Computationally expensive	$O(n^2)$	$O(n)$	$O(n^2)$
CLSF [Williams 2002] + Allows delayed map fusion – Computationally expensive	$O(n^2)$	$O(1)$	$O(n^2)$
LMJ [Tardós 2002] + Consistent map fusion – Computationally expensive	$O(n^2)$	$O(1)$	$O(n^2)$
CTS [Newman 2003] – Not tested in non linear problems – Suboptimal	$O(n)$	$O(1)$	–
Atlas [Bosse 2004] + Modular – Strong constrains	$O(n)$	$O(1)$	–
HS [Estrada 2005] + Global convergence – Optimization complexity	$O(n)$	$O(1)$	$O(n)$
DCS [Paz 2008] + Fast data association	$O(n^2)$	$O(1)$	$O(n)$
CIS [Piniés 2009] + Fast back-propagation + Share information between submaps	$O(n)$	$O(1)$	$O(n)$

the same drawbacks than the LBL. Recently, several AUVs have used Ultra Short-Baseline (USBL) technology, which consists of a transceiver, usually placed on the surface, a pole under the vessel, and a transponder mounted on the AUV. This technology is more accurate than LBLs and SBLs. Another set of approaches avoids the use of external devices by using computer algorithms. For instance, the use of particle filters for AUV localization presented in [Maurelli 2008]. This approach was shown to work with high performance, however, it only works when the map is known a-priori.

When the map is unknown, SLAM is conducted. Underwater scenarios are still one of the most challenging scenarios for SLAM because of the reduced sensory possibilities, the unstructured nature of the seabed and the difficulty of finding reliable features. Many underwater features are scale dependant, sensitive to viewing angle and look very small. A SLAM proposal attempts to solve the problem by using point features [Williams 2001]. This

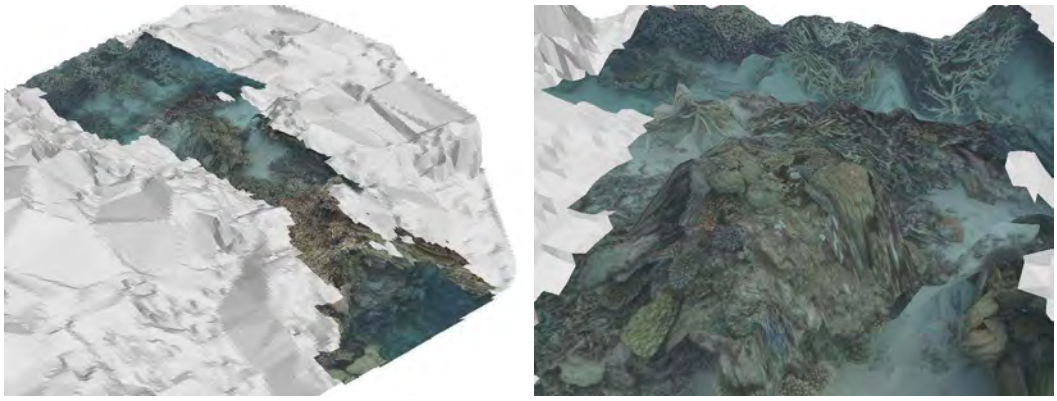


Figure 2.6: Terrain models built by projecting the texture of the visual images onto a surface model generated by the sonar data.

approach fuses information from the vehicle's on-board sonar and vision systems. They use highly textured information to estimate the vehicle's motion and to generate models of the sea floor structure. Later on [Williams 2004] proposed a similar approach using an EKF based SLAM and combining sonar and vision to obtain the 3D structure and texture (see Figure 2.6. [Leonard 2001, Newman 2003] also used point features. The former implemented the DCS and performed tests in a water tank, while the later proposed the CTS and used LBL information to help with the localization. A non-feature based approach to SLAM that utilizes a 2D grid structure to represent the map and a Distributed Particle Filter to track the uncertainty of the vehicle's state was presented in [Barkby 2009]. They named their method the bathymetric distributed Particle SLAM filter. This method does not need to explicitly identify features in the surrounding environment or apply complicated matching algorithms. However, they did require a prior low-resolution map generated by a surface vessel. Another approach using bathymetric sub-maps is the one presented by [Roman 2007].

Another approach uses a 3D occupancy grid map representation, efficiently managed with Deferred Reference Counting Octrees [Fairfield 2007]. A particle filter is used to handle the uncertainty of the navigation solution provided by the vehicle. This approach was successful in minimizing the navigation error during a deep sea mapping mission. However, map based localization was only available after the map building process had been carried out. This prohibited any corrections in navigation during the map building process. Later on, they proposed a similar but improved solution [Fairfield 2008], capable of providing real-time localization, with results comparable to those given by SBLs and USBLs.

A vision-based localization approach for an underwater robot in a structured environment was presented in [Carreras 2003]. This system was based on a coded pattern placed on the bottom of a water tank and an on-board down-looking camera. The system provided three-dimensional position and orientation of the vehicle along with its velocity. Another

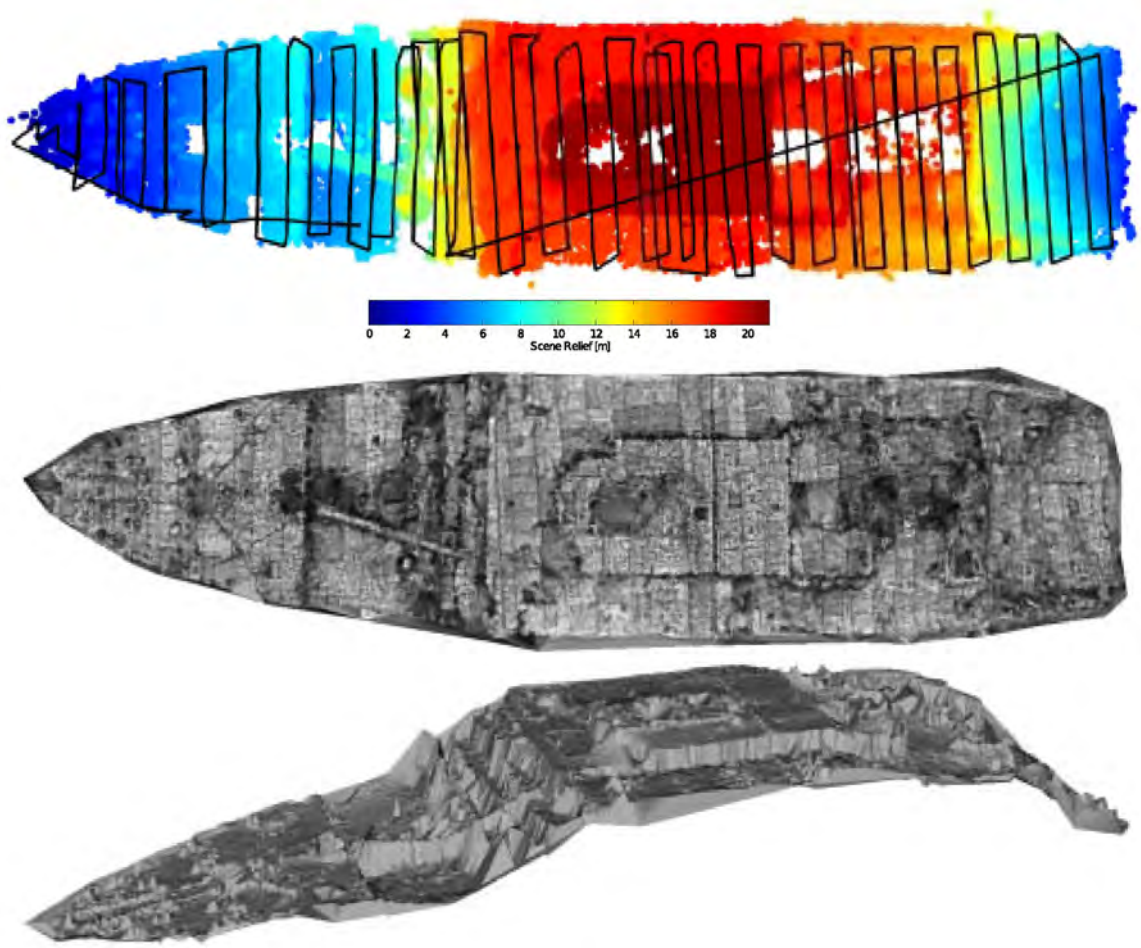


Figure 2.7: RMS Titanic reconstruction, using optical imaging [Eustice 2005a].

vision-based algorithm [Eustice 2008] used inertial sensors together with the typical low-overlap imagery constraints of underwater imagery. Their strategy consisted of solving a sparse system of linear equations in order to maintain consistent covariance bound within a SLAM information filter. The main limitation of vision-based techniques is that they are limited to near-field vision (1–5m). Deep water missions will require higher amounts of energy for lighting purposes. In previous project [Eustice 2005b, Eustice 2006a] presented the reconstruction of the RMS Titanic from a set of images using IF (see Figure 2.7). Using a Sparse Extended Unformation filter (SEIF) and a forward-looking sonar, [Walter 2008] presented a SLAM approach to inspect ship hull.

Instead of vision, in [Ribas 2008b] a mechanically scanned imaging sonar was used to obtain information about the location of vertical planar structures present in partially structured environments. In this approach, the authors extracted line features from sonar data

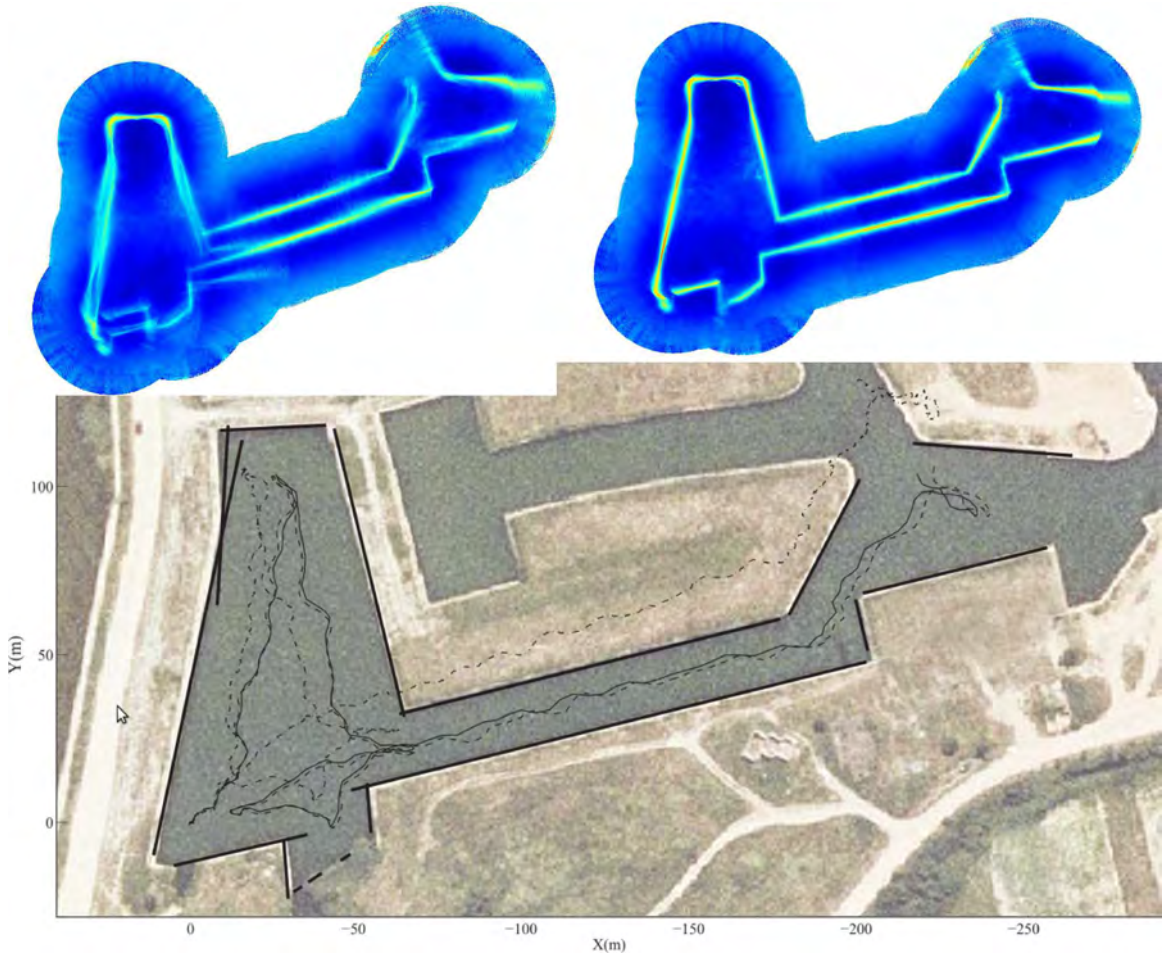


Figure 2.8: Abandoned marina SLAM example, using imaging sonar [Ribas 2008a].

by means of a robust voting algorithm (see Figure 2.8). These line features were used in the EKF base SLAM. In [Tena-Ruiz 2004] a side-scan sonar was used to sense the environment. The returns from the sonar were used to detect landmarks in the vehicle's vicinity. Reobserving these landmarks allows correction of the map and vehicle location. However, after long distances the drift is too large to allow the association of landmarks with current observations. For this reason, they proposed a method that combines a forward stochastic map in conjunction with a backward Rauch-Tung-Striebel (RTS) filter to smooth the trajectory.

Table 2.3 summarizes several aspects of existing underwater SLAM solutions. This information gives an idea of the remote sensing device used, the map representation and the filtering technique used. In addition, information about the vehicle and the research group is given. Most of these approaches have some points in common, for instance, the most commonly used sensor is an imaging sonar, the most common filtering technique is the EKF

Table 2.3: Summary of underwater SLAM approaches.

Method	Group	Vehicle	Sensor	Map	Filter
[Leonard 2001]	MIT	–	imaging sonar	point features	DSM
[Williams 2001]	ACFR	Oberon	imaging sonar	point features	EKF
[Newman 2003]	MRG/MIT	Caribou	imaging sonar	point features	CTS
[Tena-Ruiz 2004]	OSL	REMUS	side-scan	point features	EKF
[Williams 2004]	CAS	Oberon	camera/sonar	point features	EKF
[Fairfield 2007]	CMU	DEPTHX	sonar beams	evidence grid	PF
[Roman 2007]	WHOI	JASON	multibeam	bathymetry	EKF
[Eustice 2008]	WHOI	SeaBED	camera	vehicle poses	EIF
[Fairfield 2008]	CMU	MBAUV	sonar beams	evidence grid	PF
[Ribas 2008b]	ViCoRob	Ictineu	imaging sonar	line features	EKF
[Walter 2008]	MIT	HAUV	imaging sonar	point features	SEIF
[Barkby 2009]	CAS	Sirus	multibeam	bathymetry	PF
[Mallios 2009]	ViCoRob	Ictineu	imaging sonar	vehicle poses	EKF

Table legend:

ACFR	Australian Center for Field Robotics	Sydney, Australia
CAS	Centre of excellence for Autonomous Systems	Sydney, Australia
CMU	Carnegie Mellon University	Pittsburgh, PA, US
MRG	Mobile Robotics Group	Oxford, UK
MIT	Massachusetts Institute of Technology	Cambridge, MA, US
OSL	Ocean Systems Laboratory	Edinburgh, UK
ViCoRoB	Computer Vision and Robotics group	Girona, Spain
WHOI	Woods Hole Oceanographic Institution	Woods Hole, MA, US

and point features are commonly used to represent the map. Some approaches use side-scan sonar or optical cameras, which seem to be becoming more important as technology advances.

2.5 A Review of Underwater Imaging

This section presents an overview of the most relevant contributions to the underwater vision community. This review gives some insights on those existing approaches that work on side-scan sonar images and those working with underwater optical imaging. Later on in this section, Table 2.4 and Table 2.5 summarize these works. The choice for these two sensing devices (i.e., side-scan sonar and optical camera) is motivated by the fact that the technology behind these sensors has evolved up to the point of providing useful data.

2.5.1 Side-scan Sonar Imaging

Nowadays, side-scan sonar is widely used in industry and academic research programs to survey the sea floor [Chandran 2002] and as a Mine Counter Measures (MCM) tool. Several approaches tackle the issues related to object detection and classification, image segmentation and registration, and feature extraction. For instance, [Johnson 1994] proposed a threshold and clustering theory to segment a side-scan sonar image into bright spots, shadows and background. Another approach uses adaptive threshold techniques to detect and extract geometric features for both bright spots and shadows [Ciany 2000]. Similarly, [Ruiz 2001] used a double threshold to generate a binary image and then extract features. Their choice for feature matching was the Multiple Hypothesis Tracking Filter (MHTF) combined with topological information from SLAM. [Bell 2002] presented a detection approach based on the Co-operating Statistical Snake, which segmented the image into two regions: one for highlights and one for shadows. They proposed two classification techniques: 1) used several views of the same object to generate a 3D model, easyly identified by a human operator (i.e., non-automatic classification); and 2) used the Monte Carlo Markov Chain to compare segmented shadows with others from a database. In [Reed 2003] the authors presented an unsupervised model capable of extracting, detecting and classifying shadows automatically. A Markov Random Field (MRF) model was used to segment the image into regions, using the geometric signature of mines in side-scan sonar images. Later on, they presented a new approach to segment side-scan sonar images using pixel-based textural features and a classifier based on fusing a voting strategy and MRF. They proved the method by generating a mosaic of the surveyed area [Reed 2006]. A completely different strategy is to use machine learning techniques, such as neural networks used to detect man-made objects from sonar images [Perry 2004] and eigen analysis as a MCM tool [Saisan 2008]. A recent approach [Lianantonakis 2007], extracts texture features from side-scan images first and then a region based active contour model is applied to segment objects. A different approach using computer vision techniques to generate multiresolution 3D reconstruction from side-scan sonar images was presented by [Coiras 2007]. In this case, the side-scan image formation process is represented by a Lambertian diffuse model, which is then inverted by a multiresolution optimization procedure inspired by expectation-maximization to account for the characteristics of the sea floor (see Figure 2.9).

2.5.2 Optical Imaging

As existing hardware improves, the interest in using optical cameras under water increases. Optical cameras provide high resolution imaging of the sea floor easyly interpreted by operators and scientists. These images are useful for many applications such as inspection and maintenance of underwater man-made structures [Walter 2008], wreck

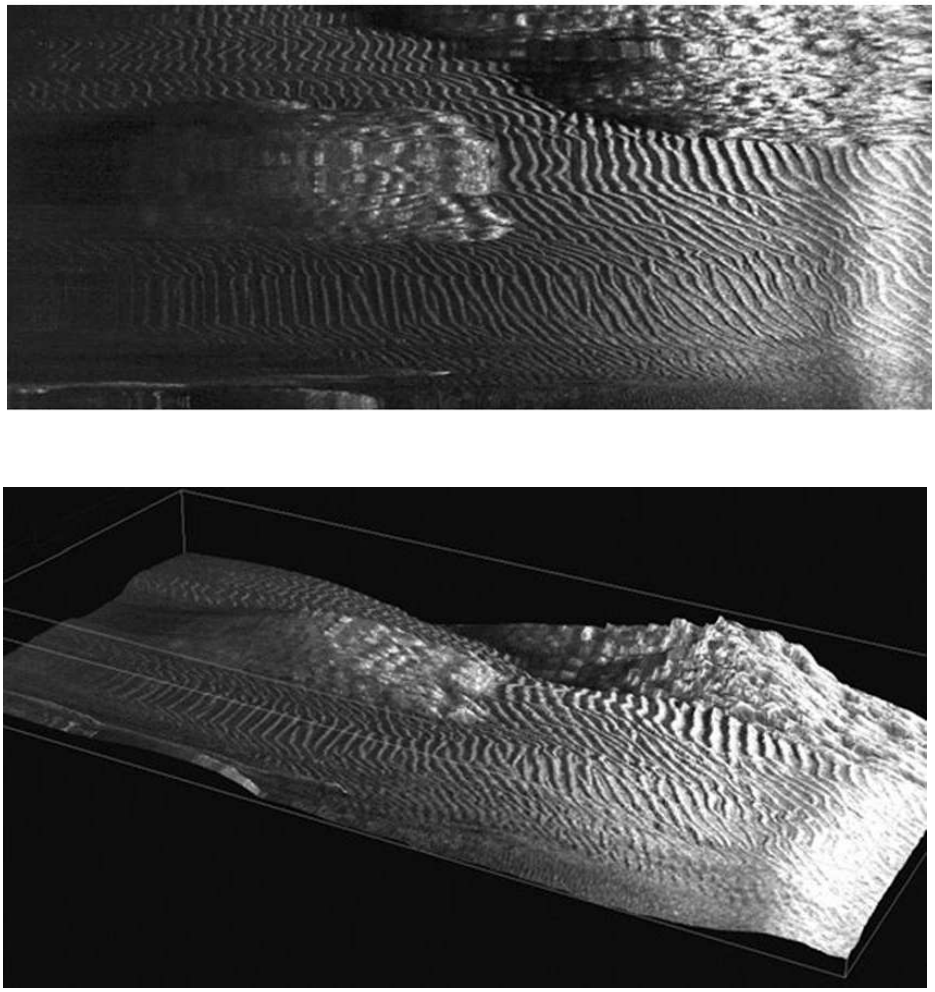


Figure 2.9: Multiresolution 3D reconstruction from a side-scan sonar image. Original image on the top and 3D model on the bottom [Coiras 2007].

localization [Kahanov 2001, Eustice 2005b], mine countermeasures and seabed surveying [Johnson-Roberson 2010]. In these applications, computer vision algorithms might be useful on station keeping [Cufi 2002, Negahdaripour 2006], cable tracking [Balasuriya 2002, Ortiz 2008, Wirth 2008], motion estimation (as a navigation aid) [Garcia 2006], localization [Garcia 2001] and/or mosaicking [Gracias 2003, Gracias 2005]. Mosaicking strategies normally assume planarity, which in large scale mosaicking is not very realistic. Large areas can contain very rugged terrain, therefore, it is necessary to account for any three-dimensional structures. [Hartley 2000] studies the theory to convert optical imagery to three-dimensional representations extensively. Recent projects use optical cameras to generate 3D underwater reconstruction of the scenario [Sáez 2006, Nicosevici 2008, Johnson-Roberson 2010]. In all

Table 2.4: Summary of side-scan sonar object detection approaches.

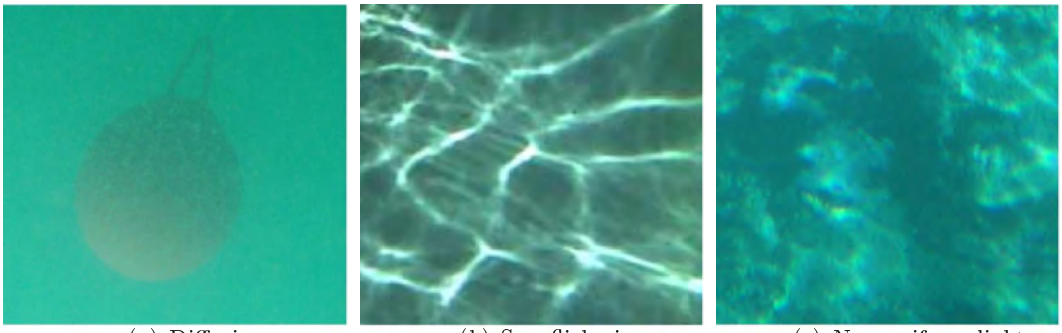
Method	Feature extraction	Feature matching
[Ciany 2000]	Threshold	–
[Ruiz 2001]	Threshold	MHTF
[Bell 2002]	CSS	MCMC
[Reed 2003]	MRF	–
[Perry 2004]	Geometry / Moments	Neural Network
[Reed 2006]	Texture	MRF / voting
[Coiras 2007]	Shape / Texture	–
[Lianantonakis 2007]	Texture / active contour	–
[Saisan 2008]	Texture	Eigen analysis

Table legend:	
CSS	Co-operating Statistical Snake
MCMC	Monte Carlo Markov Chain
MRF	Markov Random Field
MHTF	Multiple Hypothesis Tracking Filter

these approaches, computer vision algorithms are used to segment and interpret images, extract features, and to detect and classify objects.

Features are selected to provide robustness when face with a certain degree of distortion, so that the same point can be detected when observed from a different point. Underwater images are very challenging, because apart from changes caused by camera motion, they normally suffer from specific artifacts due to the medium (see Figure 2.10). These distortions are caused by diffusion which produces low contrast, scattering, blur and loss of colour (see Figure 2.10(a)), sun flickering which produces random patterns in all directions depending on the shape of the surface of the water (see Figure 2.10(b)), and non-uniform lighting (see Figure 2.10(c)). Several approaches propose image processing algorithms to address these issues, for instance [Garcia 2002] presents an approach to correct lighting effects and [Gracias 2008] presents a technique to filter flickering.

In the SLAM context, features must be distinguishable in order to ease the association of new observations to corresponding map features. In general, SLAM approaches use features that can be detected by their location; this is, features that are far apart within the map. However, in underwater environments, it is interesting to have as many features as possible and to observe them repeatedly, in order to reduce the uncertainty caused by any significant vehicle drift. In this sense, features from optical images are used either to estimate the motion on a frame to frame basis, but also as landmarks for the SLAM problem. These landmarks have to be very robust. Several methods exist to extract features from optical images. Edge, corner and contour detectors are commonly used in computer vision, for instance the well-known Canny edge detector [Canny 1986], or the Harris cor-



(a) Diffusion. (b) Sun flickering. (c) Non-uniform light.

Figure 2.10: Artifacts that appear on underwater images.

ner detector [Harris 1988]. These features are commonly used in cable tracking approaches [Balasuriya 2002, Ortiz 2002] and for mosaicking [Gracias 2003]. In addition, texture patches are used to provide more information on the points of interest, and improve the matching step [Gracias 2005]. However, image patches show poor robustness to viewpoint changes and scale. A different invariant approach is to use moment based descriptors [Mindru 1999], for instance, [Pizarro 2003] uses Zernike moments, which are robust to scale and rotation. More robust approaches are the Scale Invariant Feature Transform (SIFT) [Lowe 2004] and more recently the Speed Up Robust Features (SURF) [Bay 2006], which produce rotation and scale invariant features. SIFT and SURF features are becoming important features in recent approaches [Nicosevici 2007, Salvi 2008].

In most of these approaches, the output of the feature extraction step is a set of keypoints with features and descriptors for every image. Feature matching algorithms are necessary to allow proper data association. Traditionally, the cross correlation between two image patches was used, but this metric is weak when face with slight rotations or scale variations. A common practice is to match these keypoints between two images based on the similarity of their descriptors, i.e., the Euclidean distance between descriptor vectors. This approach is prone to find correct pairings, however, many features will not have a match, because they either belong to the background or they were not detected in the second image. For this reason, SIFT and SURF matching algorithms use the same distance together with a comparison with neighbouring features, making the match more robust [Lowe 2004].

2.6 Discussion

This chapter introduced the SLAM problem. First a review of the most relevant filtering techniques in the literature was given. Afterwards, special attention was focused on those solutions capable of dealing with large maps and long missions. Then, the most relevant

Table 2.5: Summary of underwater SLAM approaches.

Method	System	Features	Matching
[Balasuriya 2002]	Monocular	line features	Hough transform
[Cufi 2002]	Monocular	corners	correlation texture similarity
[Ortiz 2002]	Monocular	contours / threshold	correlation
[Gracias 2003]	Monocular	Harris corners	correlation Lucas Kanade
[Pizarro 2003]	Monocular	multiscale Harris Zernike moments	descriptor vector distance
[Gracias 2005]	Monocular	Harris / texture	cross-correlation
[Sáez 2006]	Trinocular	disparity / texture	correlation
[Nicosevici 2007]	Monocular	geometric features SIFT	descriptor vector distance
[Eustice 2008]	Monocular	Harris / SIFT	SIFT match / pose epipolar geometry
[Ortiz 2008]	Monocular	line features	temporal correlation
[Salvi 2008]	Stereo	SIFT / SURF 3D structure	descriptor vector distance
[Wirth 2008]	Monocular	line features	particle filter

underwater SLAM approaches were summarized. Finally, underwater computer vision techniques were analysed, first those techniques that deal with side-scan sonar images and then those techniques using optical images.

As has been seen in Section 2.2, several filtering techniques can be used in SLAM to deal with uncertainties and noise. It seems clear that an EKF based SLAM is the most common option. However, the most interesting outcome of the survey is that for large scenarios, or maps with a high population of landmarks, an EKF suffers from several limitations:

- gaussianity assumption, which in the real world might not be true
- linearization approximations may cause divergences on long missions and large maps.
- memory demand increases with the size of the map
- computational time rises with the size of the map

An IF produces a fast stable solution, however there is no covariance matrix involved in the process, which means important data association issues. The UKF solves the non-linearities by using the Unscented transform, however its computation is very costly. On the other hand, particle filters handle linearization and gaussianity problems, but the cost grows exponentially. To the best of the author's knowledge at the time of writing, particle filters

have only been shown successful on localization problems, while expectation maximization proposals seem to be optimal for map building, but unstable for large scenarios.

Section 2.3 presented submapping SLAM solutions. The core for most of these submapping approaches is the EKF. These approaches show significant improvements in terms of computational demand reduction and map consistency. Although these solutions show promising results on structured indoor environment and outdoor terrestrial applications, none of them have been used in underwater applications.

Section 2.4 demonstrated that underwater technologies, especially, underwater autonomous technologies, are gaining importance. Most of these projects use EKF based solutions with point features as landmarks. These features are obtained from acoustic sensors such as imaging sonar and side-scan sonar. Recent approaches use optical cameras as the main sensor which provides rich visual information for operators and scientists. Extracting features from underwater data, either acoustic or optical, is a difficult task due to the artifacts from underwater environments and the vehicle's motion.

Section 2.5 summarized the most relevant projects that focused on feature extraction and matching from side-scan sonar images and from underwater optical images. Initial proposals were based on using thresholds to separate bright spots and shadows from side-scan sonar images. In recent projects, there is a trend to machine learning algorithms which basically means that a system is first trained with specific features which renders it capable of detecting them in real missions. On the other hand, extracting features from optical images has been traditionally done by means of corner and edge detectors. SIFT and SURF seem to be a common choice nowadays because they are robust and fast. However, it seems clear that there are still many difficulties in defining robust enough features under water.

From this survey, several points stand out: 1) an EKF based SLAM seems to be an appropriate choice because it handles uncertainties and produces convergent solutions, 2) using EKF based submapping strategies is even better because they address the issues of standard EKF approaches, i.e. linearization errors and high computational demand, 3) most existing underwater SLAM solutions use an EKF with acoustic remote sensing devices, however none of them apply submapping strategies, and 4) the detection of robust features is a very complex task in an underwater scenario because they have an added difficulty arising from artifacts and noise due to the medium.

Therefore, the use of EKF submapping based SLAM combined with underwater imaging techniques will be the starting point for the approach presented in this work.

Chapter 3

Selective Submap Joining SLAM

Contents

Simultaneous *Localization and Mapping (SLAM) does not result in consistent maps of large areas because of a gradual increase in uncertainty for long term missions. In addition, as the map size grows the computational cost increases, making SLAM solutions unsuitable for on-line applications. The SSJS is a novel solution based on the use of independent local maps together with a global stochastic map. The global map contains the relative transformations between local maps which are updated once a new loop is detected. The information within local maps is also corrected. Thus, maps sharing a certain number of features are fused, maintaining the correlation between landmarks and the vehicle's position. The use of local maps reduces computational costs and improves map consistency. Synthetic and*

experimental results show that the proposed approach is able to map large areas consistently with lower computational cost compared to state of the art methods.

3.1 Overview

This Chapter presents the probabilistic background for the well known KF and its extension to the EKF. The EKF is the core of the proposed map building approach as well as other state of the art submapping approaches which are briefly introduced in this chapter to put the reader in context. Afterwards, the Selective Submap Joining algorithm is detailed. Finally, synthetic and real experiments and their results are presented.

3.2 Probabilistic Framework

3.2.1 Notation

In the following sections a common notation is used. These common variables and their meaning are listed in Table 3.1.

Table 3.1: Notation

Variable	Description
x_k	actual state vector at the current time step
P_k	actual covariance matrix at the current time step
z_{k+1}	next measurement vector for the current time step
$\widehat{x}_{k+1}$	next state vector estimate
$\widehat{P}_{k+1}$	next covariance matrix estimate
$\widehat{z}_{k+1}$	next measurement vector estimate
x_{k+1}	corrected state vector
P_{k+1}	corrected covariance matrix
F_k	transition matrix (process model)
H_k	measurement matrix (observation model)
u_k	control vector
Q_k	process uncertainty matrix
R_k	observation uncertainty matrix
ν_{k+1}	innovation vector
S_{k+1}	innovation associated matrix
W_{k+1}	Kalman filter gain

3.2.2 Kalman Filter

The KF was invented in the 1950s by Rudolph Emil Kalman [Kalman 1960] as a technique for filtering and predicting in linear systems. The KF implements belief computation for continuous states. It is not applicable to discrete or hybrid state spaces. The KF estimates the belief of the state of a dynamic system at a certain moment in the time t from a series of incomplete and noisy measurements through their mean μ_t and the covariance Σ_t.

3.2.3 Into the Maths: Assumptions

In order to obtain the correct a-posteriori probability, three assumptions have to be fulfilled. First, the initial belief must be Gaussian (3.1). Second, the state transition probability must be composed of a function that is linear in its argument with added independent Gaussian noise (3.2). Third, the same applies to the measurement probability (3.3). Systems that meet these assumptions are called linear Gaussian systems.

$$bel(x_0) = p(x_0) = \frac{1}{(|2\pi\Sigma_0|)^{\frac{1}{2}}} \exp\{-\frac{1}{2}(x_0 - \mu_0)^T \Sigma_0^{-1}(x_0 - \mu_0)\} \tag{3.1}$$

$$x_t = A_t x_{t-1} + B_t u_t + \varepsilon_t \tag{3.2}$$

$$z_t = C_t x_t + \delta_t \tag{3.3}$$

In (3.2) and (3.3) A_t, B_t and C_t are matrices that multiply the state, control and measurement vectors, allowing us to assimilate the transition and measurement functions as linear systems. ε_t and δ_t are Gaussian random vector and matrix respectively, with zero mean and covariances Q_t and R_t.

3.2.4 The Kalman Filter Algorithm

The KF algorithm adapted to the SLAM context is presented in Algorithm 1. The algorithm is composed of three main steps, prediction, observation and update[1], which keep repeating along time. In the prediction stage, the process model is used to obtain an estimate for the state vector $\widehat{x}_{k+1}$, the covariance matrix $\widehat{P}_{k+1}$ and the measurement $\widehat{z}_{k+1}$ for instant $k+1$ from the state vector x_k and the covariance matrix P_k of the previous instant k. The observation stage computes the innovation vector ν as a difference between real and estimated measurements and the innovation associated matrix S. This matrix S is then used in the update stage to compute the Kalman gain W. Both the ν and the W are then

[1] Various authors define only two steps: prediction and update. They consider the observation as a part of the update stage

Algorithm 1 (x_{k+1}, P_{k+1}) = Kalman Filter(x_k, P_k, z_{k+1})

Prediction (estimate)

1: $\widehat{x}_{k+1} = F_k x_k + u_k$

2: $\widehat{P}_{k+1} = F_k P_k F_k^{\mathrm{T}} + Q_k$

3: $\widehat{z}_{k+1} = H_k \widehat{x}_{k+1}$

Observation (innovation vector and matrix)

4: $\nu_{k+1} = z_{k+1} - \widehat{z}_{k+1}$

5: $S_{k+1} = H_k \widehat{P}_{k+1} H_k^{\mathrm{T}} + R_k$

Update (correction, Kalman Gain)

6: $W_{k+1} = \widehat{P}_{k+1} H_k^{\mathrm{T}} S_{k+1}^{-1}$

7: $x_{k+1} = \widehat{x}_{k+1} + W_{k+1} \nu_{k+1}$

8: $P_{k+1} = \widehat{P}_{k+1} - W_{k+1} S_{k+1} \widehat{P}_{k+1}$

used to correct the state vector x_{k+1} and the covariance matrix P_{k+1}, which are used as input for the next iteration.

3.2.5 The Extended Kalman Filter

The assumptions of linear state transitions and linear measurements with added Gaussian noise are rarely fulfilled in practice. For example, a robot that moves with a constant translational and rotational velocity typically moves in a circular trajectory which cannot be described by linear next state transitions. This observation, along with the assumption of unimodal beliefs, makes plain a KF inapplicable to all but the most trivial robotics problems. The EKF overcomes this linearity assumption.

3.2.6 Into the Maths: Non-linear Systems

In the EKF, the assumption is that the next state probability (3.4) and the measurement probabilities (3.5) are non-linear functions g and h which are essentially the equivalent to (3.2) and (3.3) with g instead of A_t and B_t, and h instead of C_t.

$$x_t = g(u_t, x_{t-1}) + \varepsilon_t \tag{3.4}$$

$$z_t = h(x_t) + \delta_t \tag{3.5}$$

Unfortunately, with arbitrary functions g and h, the belief is no longer Gaussian. For this reason the EKF calculates an approximation to the true belief. It represents this approximation with a Gaussian. Thus, the EKF inherits the basic belief representation from the KF, but it differs in that this belief is only approximate, not exact as was the case in the KF.

Algorithm 2 Extended Kalman Filter(x_k, P_k, z_{k+1})

Prediction (estimate)

1: $\widehat{x}_{k+1} = g(x_k, u_{k+1})$

2: $\widehat{P}_{k+1} = G_k P_k G_k^{\mathrm{T}} + Q_k$

3: $\widehat{z}_{k+1} = h(\widehat{x}_{k+1})$

Observation (innovation vector and matrix)

4: $\nu_{k+1} = z_{k+1} - \widehat{z}_{k+1}$

5: $S_{k+1} = H_k \widehat{P}_{k+1} H_k^{\mathrm{T}} + R_k$

Update (correction, Kalman Gain)

6: $W_{k+1} = \widehat{P}_{k+1} H_k^{\mathrm{T}} S_{k+1}^{-1}$

7: $x_{k+1} = \widehat{x}_{k+1} + W_{k+1}\nu_{k+1}$

8: $P_{k+1} = \widehat{P}_{k+1} - W_{k+1} S_{k+1} \widehat{P}_{k+1}$

The technique for linearizing non-linear functions used by an EKF is called *Taylor Expansion*, which constructs linear approximations taking advantage of the partial derivative of the non-linear function (Equation 3.6). To do this, the *Jacobian* of a non-linear function is computed. Within the EKF framework, g and h become G and H, being the Jacobian of g and h.

$$g'(u_t, x_{t-1}) = \frac{\partial g(u_t, x_{t-1})}{\partial x_{t-1}} \tag{3.6}$$

3.2.7 The Extended Kalman Filter Algorithm

The EKF algorithm applied to SLAM is presented in Algorithm 2. The iterative process is similar to that in the KF algorithm (Algorithm 1), but now instead of the transition matrix F and the measurement matrix H, the corresponding non-linear functions g and h and their Jacobian G and H are used. Note that now the H matrix is the Jacobian of the non-linear function h.

3.3 The Selective Submap Joining Basis

The main novelty in this approach is the loop closing strategy which involves a decision on whether to fuse local maps when closing loops and is the inspiration behind the name of this approach, SSJS (see Algorithm I). The basis of this approach is the EKF based SLAM. A sequence on which EKF based submaps are built. The size of these submaps is predefined by the total number of features per map and by the uncertainty boundaries. The links between local maps are stored in a global stochastic map. This information can be used to check the possibility of being confronted with a loop closing event. A loop closure is performed when the vehicle revisits a certain number of previous observations and two maps are joined

Algorithm 3 Selective Submap Joining SLAM.

begin mission
while navigating **do**
  $\widehat{\mathbf{x}}_i, \widehat{\mathbf{P}}_i$ = EKF SLAM() $\leftarrow$ (*Build submap* $\mathcal{M}_i$)
  $\widehat{\mathbf{x}}_G, \widehat{\mathbf{P}}_G$ = build global map($\widehat{\mathbf{x}}_i$, $\widehat{\mathbf{P}}_i$)
  $\mathcal{H}_{Loop}$ = check possible loops($\widehat{\mathbf{x}}_G$, $\widehat{\mathbf{P}}_G$)
  for $j = \mathcal{H}_{Loop}$ **do**
    refer $\mathcal{M}_i$ and $\mathcal{M}_j$ to a common base reference
    $\mathcal{H}_{ij}$ = data association($\widehat{\mathbf{x}}_i, \widehat{\mathbf{x}}_j, \widehat{\mathbf{P}}_i, \widehat{\mathbf{P}}_j$)
    if $\mathcal{H}_{ij} > threshold$ **then**
      $\widehat{\mathbf{x}}_{ij}, \widehat{\mathbf{P}}_{ij}$ = map fusion($\widehat{\mathbf{x}}_i$, $\widehat{\mathbf{P}}_i$, $\widehat{\mathbf{x}}_j$, $\widehat{\mathbf{P}}_j$, $\mathcal{H}_{ij}$)
      $\widehat{\mathbf{x}}_G, \widehat{\mathbf{P}}_G$ = update global map($\widehat{\mathbf{x}}_{ij}, \widehat{\mathbf{P}}_{ij}$)
    endif
  endfor
endwhile

and fused. Deciding whether to fuse two maps or keep them independent, depending on the number of landmarks they have in common, is the main difference between this method and other approaches that fuse maps regardless of the information they share.

3.3.1 Map Building

The SSJS builds a sequence of submaps. A submap is built using a standard EKF algorithm. Algorithm II summarises the adaptation of the EKF to the SLAM problem.

The EKF estimates belief about the state at a certain step at time k of a dynamic non-linear system through the mean x_k and the covariance P_k of a series of incomplete and noisy measurements. The algorithm is composed of three steps: prediction, observation and update, which repeat in a sequence over time.

The **prediction** stage uses the state estimate from the previous time step $\mathbf{x}_{k-1}$ to produce an estimate of the state at the current time step $\widehat{\mathbf{x}}_k$ (see Equation 3.7).

$$\widehat{\mathbf{x}}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_k) \qquad \widehat{\mathbf{P}}_k = \mathbf{F}_k \mathbf{P}_{k-1} \mathbf{F}_k^{\mathrm{T}} + \mathbf{G}_k \mathbf{Q}_k \mathbf{G}_k^{\mathrm{T}} \tag{3.7}$$

If control inputs $\mathbf{u}_k$ are available, they are used in the prediction stage. The **motion model** f is used to estimate the state at a given step k from the state at a step $k-1$. The linearized function $\mathbf{F}_k$ in Equation 3.8 is used to estimate the changes in the covariance matrix from time $k-1$ to k.

Algorithm 4 EKF based SLAM.

map initialisation()
$\mathbf{z}_0, \mathbf{R}_0$ = get measurements()
$\mathbf{x}_0, \mathbf{P}_0$ = add features()
for $k = 1$ until end of map **do**
  $\mathbf{x}_{od}, \mathbf{Q}_{od}$ = get odometry()
  $\widehat{\mathbf{x}}_{k|k-1}, \widehat{\mathbf{P}}_{k|k-1}$ = EKF prediction($\widehat{\mathbf{x}}_{k-1}$,$\widehat{\mathbf{P}}_{k-1}$,$\mathbf{x}_{od}$,$\mathbf{Q}_{od}$)
  $\mathbf{z}_k, \mathbf{R}_k$ = get measurements()
  $\mathcal{H}_k$ = data association($\widehat{\mathbf{x}}_{k|k-1}$,$\mathbf{z}_k$,$\widehat{\mathbf{P}}_{k|k-1}$,$\mathbf{R}_k$)
  $\widehat{\mathbf{x}}_k, \widehat{\mathbf{P}}_k$ = EKF update($\widehat{\mathbf{x}}_{k|k-1}$,$\widehat{\mathbf{P}}_{k|k-1}$,$\mathbf{z}_k$,$\mathbf{R}_k$,$\mathcal{H}_k$)
  $\widehat{\mathbf{x}}_k, \widehat{\mathbf{P}}_k$ = add features($\widehat{\mathbf{x}}_k$,$\widehat{\mathbf{P}}_k$,$\mathbf{z}_k$,$\mathbf{R}_k$,$\mathcal{H}_k$)
endfor
return: $\mathcal{M}_i = \{\widehat{\mathbf{x}}_k, \widehat{\mathbf{P}}_k\}$

$$\mathbf{F}_k = \left.\frac{\partial f}{\partial \mathbf{x}}\right|_{\mathbf{x}_{k-1}} \tag{3.8}$$

During the **observation** stage, one can obtain measurements for the vehicle's orientation, linear speeds, altitude, depth and other information about the world, through the vehicle's on-board sensors. These measurements are stored in an observation vector $\mathbf{z}_k$.

The **observation model** gives the predicted sensor measurement from the last known position. It is represented by the non-linear function $\widehat{\mathbf{z}}_k = h(\widehat{\mathbf{x}}_k)$. Notice that each sensor has its own observation model. This model needs to be linearized using Jacobian computation (see Equation (3.9)), giving the linear function $\mathbf{H}_k$.

$$\mathbf{H}_k = \left.\frac{\partial h}{\partial \mathbf{x}}\right|_{\mathbf{x}_k} \tag{3.9}$$

The observation model $\mathbf{H}_k$ is used to compute the innovation vector ν_k as a difference between real and estimated measurements and the innovation associated matrix $\mathbf{S}_k$ (see Equation (3.10)). $\mathbf{S}_k$ and ν_k are then used in the EKF update stage. Note that the term $\mathbf{R}_k$ represents zero mean white Gaussian observation noise is ($\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$).

$$\begin{aligned} \widehat{\mathbf{z}}_k &= f(\widehat{\mathbf{x}}_k) \\ \nu_k &= \mathbf{z}_k - \widehat{\mathbf{z}}_k \\ \mathbf{S}_k &= \mathbf{H}_k \widehat{\mathbf{P}}_{k-1} \mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k \end{aligned} \tag{3.10}$$

Some of these observations are now being seen for the first time, but some of them might correspond to features already observed in the past. It is, therefore, necessary to identify the correspondences between existing features and observations, this is done by means of **data association** algorithms. One of the main issues with the EKF based SLAM is the lack of robustness to errors in the data association process. Robust data association algorithms are therefore critical. Given a set of n features $F_1, ..., F_n$, which have already been observed in the past and are now part of the state vector $\mathbf{x}_k$, and given a new set of m measurements $E_1, ..., E_m$, the data association algorithm must be able to generate a hypothesis $\mathcal{H}_k = j_i, ..., j_m$ that pairs each measurement E_i with a map feature F_j. Notice that for new or spurious measurements there is $j_i = 0$. Data association techniques and matching algorithms will be detailed in Chapter 4.

The prediction gives an *a-priori* state estimate $\widehat{\mathbf{x}}_{k|k-1}$ because, although it is an estimate of the state at the current time step, it does not include observation information from the current time step. In the **update stage** (see Equation (3.11)), the current *a-priori* prediction is combined with current observation information $\mathbf{z}_k$. By knowing the data association $\mathcal{H}_k$, one can compute the estimates $\widehat{\mathbf{z}}_k$ for the features that correspond to new observations $\mathbf{z}_k$ and obtain the innovation vector ν_k and matrix $\mathbf{S}_k$. This matrix $\mathbf{S}_k$ is then used to compute the Kalman gain $\mathbf{W}_k$. Both, ν_k and $\mathbf{W}_k$ are necessary to produce an improved estimate $\mathbf{x}_k$ and $\mathbf{P}_k$ (the *a-posteriori* state estimate).

$$\begin{aligned} \mathbf{W}_k &= \widehat{\mathbf{P}}_{k|k-1}\mathbf{H}_k^{\mathrm{T}}\mathbf{S}_k^{-1} \\ \mathbf{x}_k &= \widehat{\mathbf{x}}_{k|k-1} + \mathbf{W}_k\nu_k \\ \mathbf{P}_k &= \widehat{\mathbf{P}}_{k|k-1} - \mathbf{W}_k\mathbf{S}_k\widehat{\mathbf{P}}_{k|k-1} \end{aligned} \tag{3.11}$$

Finally, after updating the states and their corresponding uncertainties, the observations that were not associated with any existing landmark on the map are considered new map features. They are added to the vector state and the associated covariance matrix is also augmented.

3.3.2 Submapping and Global Map Building

Local maps $\mathcal{M}_i$, $\mathcal{M}_{i+1}$, ..., $\mathcal{M}_j$ are built sequentially (see Figure 3.1). The reference frame of a submap is at the vehicle's starting point. This starting point of a local map $\mathcal{M}_{i+1}$ coincides with the last position of the previous map $\mathcal{M}_i$. Therefore, the relative transformation between two consecutive maps' ${}^{\mathcal{M}_i}_{\mathcal{M}_{i+1}}\mathcal{T}$ is the vehicle's pose at the last position of $\mathcal{M}_i$. This link is stored in a global map $\mathbf{x}_G$ together with its uncertainty $\mathbf{P}_G$ (see Equation (3.12)). The information contained at this global level is very important in detecting loop events, since a local map can be referred to the frame reference of any other local map.

$$\mathbf{x}_G = \begin{bmatrix} . \\ {}^{\mathcal{M}_{i-1}}\mathcal{T} \\ {}^{\mathcal{M}_i}\mathcal{T} \\ {}^{\mathcal{M}_{i+1}}\mathcal{T} \\ . \end{bmatrix} \quad \mathbf{P}_G = \begin{bmatrix} . & . & . & . \\ . & \sigma^2_{{}^{\mathcal{M}_{i-1}}\mathcal{T}_{\mathcal{M}_i}} & 0 & . \\ . & 0 & \sigma^2_{{}^{\mathcal{M}_i}\mathcal{T}_{\mathcal{M}_{i+1}}} & . \\ . & . & . & . \end{bmatrix} \tag{3.12}$$

3.3.3 Map Joining and Fusion

A map building technique in which a sequence of local maps is built and then consecutive local maps are joined was suggested by Tardós et al. [Tardós 2002]. In this case, local maps do not need to be consecutive to be joined, but the same joining and fusion algorithm is applied. Given two independent local maps, $\mathcal{M}_i^{\mathcal{B}} = (\mathbf{x}_i^{\mathcal{B}}, \mathbf{P}_i^{\mathcal{B}})$, which contains a set of n features $F_1, ..., F_n$, and $\mathcal{M}_j^{\mathcal{B}'} = (\mathbf{x}_j^{\mathcal{B}'}, \mathbf{P}_j^{\mathcal{B}'})$, which contains a set of m features $E_1, ..., E_m$, one can use the transformations in the global stochastic map to refer them both to the same frame (using the formulation in Appendix A). Features from both maps are therefore expressed relative to the same base B forming a joint state vector $\mathbf{x}_{i+j}^{\mathcal{B}}$ and covariance matrix $\mathbf{P}_{i+j}^{\mathcal{B}}$ (see Equation (3.13)).

$$\mathbf{x}_{i+j}^{\mathcal{B}} = \begin{bmatrix} \mathbf{x}_i^{\mathcal{B}} \\ \mathbf{x}_j^{\mathcal{B}} \end{bmatrix} \qquad \mathbf{P}_{i+j}^{\mathcal{B}} = \mathbf{J}_i \mathbf{P}_i^{\mathcal{B}} \mathbf{J}_i^{\mathrm{T}} + \mathbf{J}_j \mathbf{P}_j^{\mathcal{B}} \mathbf{J}_j^{\mathrm{T}} \tag{3.13}$$

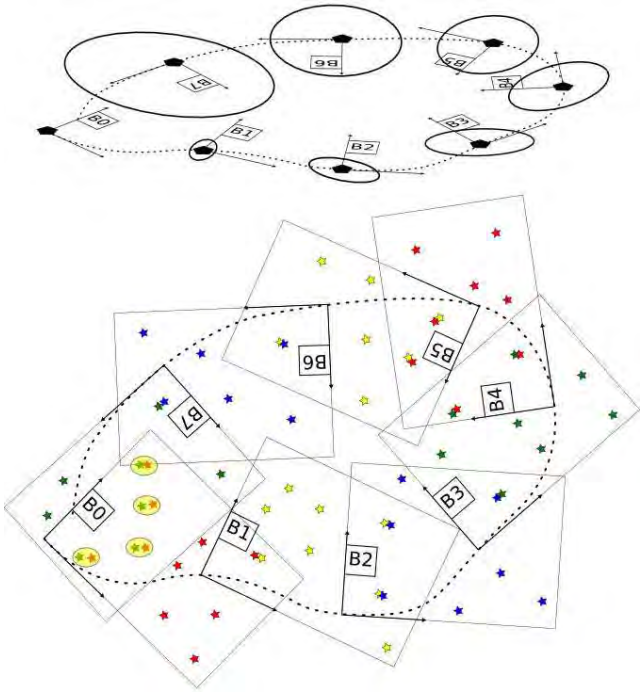
Where the matrices for the Jacobian $\mathbf{J}_i$ and $\mathbf{J}_j$ are in Equation (3.14).

$$\mathbf{J}_i = \frac{\partial \mathbf{x}_{i+j}^{\mathcal{B}}}{\partial \mathbf{x}_i^B} \qquad \mathbf{J}_j = \frac{\partial \mathbf{x}_{i+j}^{\mathcal{B}}}{\partial \mathbf{x}_j^B} \tag{3.14}$$

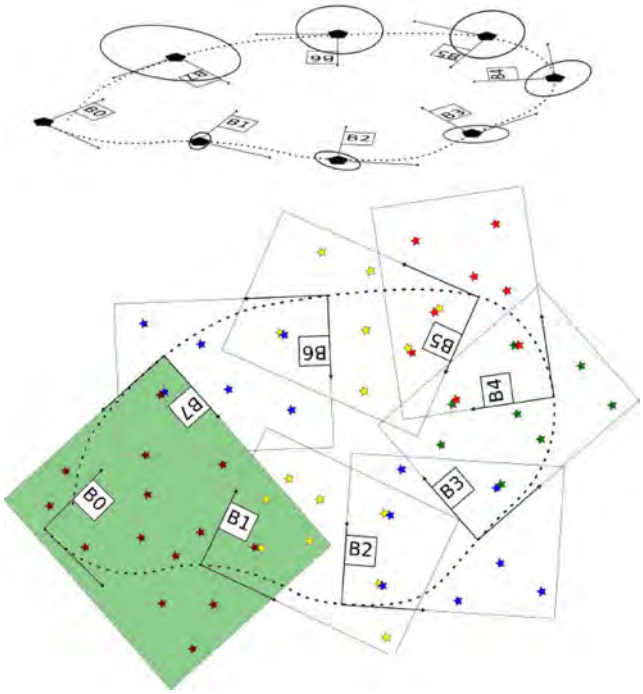
Assuming that $\mathcal{M}_i^{\mathcal{B}}$ and $\mathcal{M}_j^{\mathcal{B}}$ share a certain number of features f, a data association algorithm is carried out. Features from $\mathcal{M}_j^{\mathcal{B}}$ are understood as new measurements for the features from $\mathcal{M}_i^{\mathcal{B}}$, so a non-linear measurement $\widehat{\mathbf{z}}_{ij_f} = \mathbf{h}_{ij_f}$ mapping a feature $E_{j_f}^{\mathcal{B}}$ corresponding to a feature $F_i^{\mathcal{B}}$ needs to be linearized by means of Jacobian computation (see Equation (3.15)).

$$\mathbf{H}_{ij_f} = \begin{bmatrix} \frac{\partial \mathbf{h}_{ij_f}}{\partial \mathbf{x}_{v_i}^B} & \mathbf{0} & ... & \mathbf{0} & \frac{\partial \mathbf{h}_{ij_f}}{\partial \mathbf{x}_{F_i}^B} & \mathbf{0} & ... & \mathbf{0} & \frac{\partial \mathbf{h}_{ij_f}}{\partial \mathbf{x}_{E_{j_f}}^{B'}} & \mathbf{0} & ... & \mathbf{0} \end{bmatrix} \tag{3.15}$$

Next, the local map information and links between maps are improved using the EKF update equations (see Equation (3.11)). Once the joint maps have been updated, the rows and columns corresponding to common landmarks from $\mathcal{M}_j$ are removed from the joint state to avoid repetitions.



(a)



(b)

Figure 3.1: A Schematic representation of a submap sequence. The top plots show the global level map, while the bottom ones depict a sequence of submaps. B_0 ... B_7 are the submap base references, B_0 being the world reference frame. In a) the vehicle has just finished a submap and this is closing a loop with the first map. In this situation, several landmarks are common to both maps, they are coupled inside a circle. In b) the map joining and fusion step has taken place, updating both the local maps and the global level map.

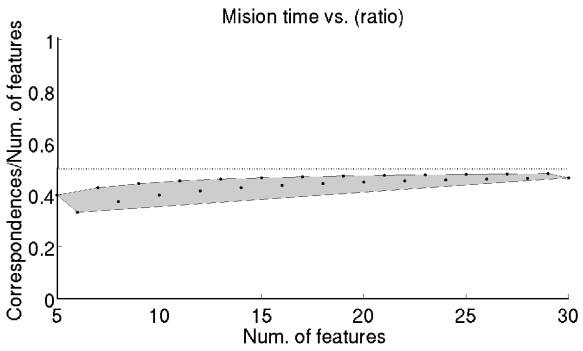


Figure 3.2: The lowest mission times occur when the *threshold* for the number of correspondences is approximately 40% and 50% the number of features per submap. The region in grey corresponds to configurations that produced minimum mission time.

3.3.4 Loop Closing Strategy

In the previous section it was shown how a sequence of submaps and a global stochastic map can be built. To improve the global consistency of the method it is necessary to navigate by identifying the loops and using the newest information to update past information. Closing a loop means revisiting a region, where some parts of the scenario with a higher level of certainty are now visited again, as depicted in Figure 3.1. A loop closing procedure begins with a search at the global map level each time a submap is finished. According to the proximity of the various submaps at the global level, loop closing hypotheses are formed. Afterwards, the data association between those maps defined as loop closing candidates is computed. If the correspondences between maps are higher than a *threshold*, they are joined and fused in a single map, as explained in Section 3.3.3. The value for this *threshold* is analysed in Section 3.4. This value is set to about fifty per cent of the size of the submap because it is the configuration with lowest computatinal cost (see Figure 3.2). Together with the map fusion, the corresponding link at the global level is corrected. This correction is directly obtained from the map fusion since the links within the fused maps are understood as new observations for the global level. These new observations are then run through the EKF update, resulting in a corrected and correlated global level map.

3.4 Experiments and Results

The SSJS algorithm is tested using synthetic and real data. In this section, the synthetic set-up and the synthetic results are described. Afterwards, the real dataset on the Victoria Park and the experimental results are presented.

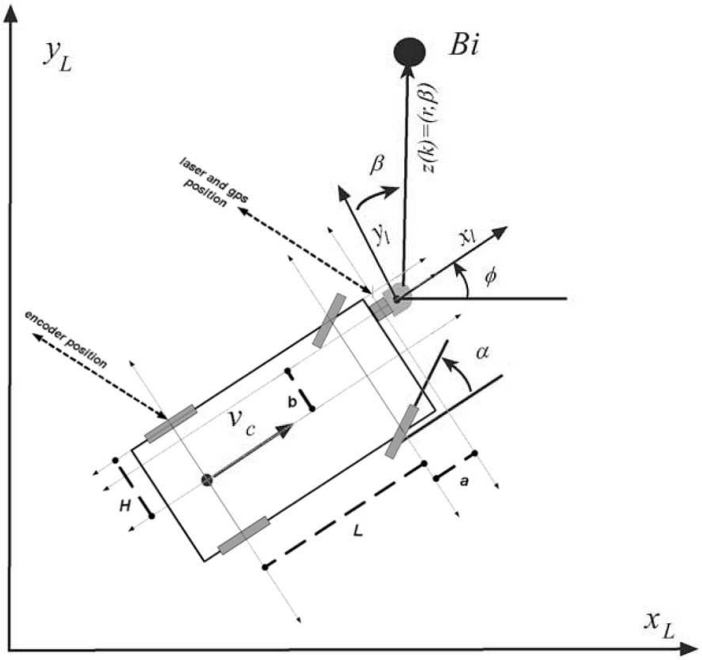


Figure 3.3: Vehicle kinematics. B_i represents a landmark. The sensors on-board the vehicle are also shown; the laser range finder on the front of the vehicle and the velocity encoder in the left rear wheel.

3.4.1 Synthetic Experiments

The synthetic experiment is composed of a 3-DOF vehicle equipped with velocity odometer, steering angle encoders, and laser range finder. The vehicle state is given by $(x\ y\ \psi)^{\mathrm{T}}$, where $(x,\ y)$ is the vehicle's position and ψ the vehicle's orientation on a plane. The map is composed of point-based features whose state is defined by the feature position $(x_{l_i}\ y_{l_i})^{\mathrm{T}}$. Features are detected by a simulated laser range finder that scans 360°. The **joint state vector estimate** $\widehat{\mathbf{x}}$ contains both vehicle and map information, as in Equation 3.16. The motion and observation models used in this simulation are detailed in the following paragraph (also used by [Guivant 2001]).

$$\widehat{\mathbf{x}} = (x\ \ y\ \ \psi\ \ x_{l_1}\ y_{l_1}\ \ldots\ x_{l_i}\ y_{l_i}\ \ldots\ x_{l_n}\ y_{l_n})^{\mathrm{T}} \tag{3.16}$$

A typical kinematic model for a land vehicle is the Ackerman Model (see Figure 3.3). This model is described by Equation 3.17. This equation is a discrete approximation of the **motion model** $f(\mathbf{x}_{k-1}, \mathbf{u}_k)$ and is used to predict the position of the vehicle from steering α and speed v_c control inputs. Notice, that L, H, b and a are vehicle parameters.

$$\begin{bmatrix} x_k \\ y_k \\ \phi_k \end{bmatrix} = \begin{bmatrix} x_{k-1} + \Delta T(v_c\cos(\phi) - \frac{v_c}{\mathbf{L}}\tan(\phi)(\mathbf{a}\sin\phi + \mathbf{b}\cos(\phi))) \\ y_{k-1} + \Delta T(v_c\sin(\phi) + \frac{v_c}{\mathbf{L}}\tan(\phi)(\mathbf{a}\cos\phi - \mathbf{b}\sin(\phi))) \\ \phi_{k-1} + \Delta T\frac{v_c}{L}\tan(\alpha) \end{bmatrix} \tag{3.17}$$

The linearized version of Equation 3.17 is given by its Jacobian $\mathbf{F}$ as in Equation 3.18.

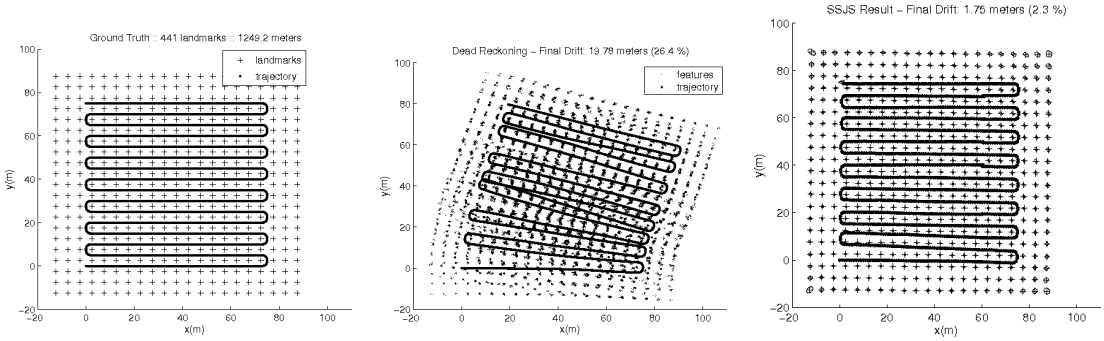
$$\frac{\partial f}{\partial \widehat{\mathbf{x}}} = \begin{bmatrix} 1 & 0 & -\Delta T(v_c\sin(\phi) + \frac{v_c}{\mathbf{L}}an(\alpha)(\mathbf{a}\cos\phi - \mathbf{b}\sin(\phi))) \\ 0 & 1 & \Delta T(v_c\cos(\phi) - \frac{v_c}{\mathbf{L}}an(\alpha)(\mathbf{a}\sin\phi + \mathbf{b}\cos(\phi))) \\ 0 & 0 & 1 \end{bmatrix} \tag{3.18}$$

A laser range finder is located on the front of the vehicle giving relative range z_r and bearing z_β. The **observation model** $\mathbf{h}_k$ in Equation 3.19 is then used to refer the features B_i from the environment with respect to the vehicle state. Its linearization $\mathbf{H}_k$ is given by Equation 3.20. Notice that $\Delta x = x_L - x_v$, $\Delta y = y_L - y_v$ and $\Delta = \sqrt{\Delta x^2 + \Delta y^2}$.

$$\begin{bmatrix} z_r \\ z_\beta \end{bmatrix} = \begin{bmatrix} \sqrt{(x_L - x_v)^2 + (y_L - y_v)^2} \\ atan\left(\frac{y_L - y_v}{x_L - x_v}\right) - \phi + \frac{\pi}{2} \end{bmatrix} \tag{3.19}$$

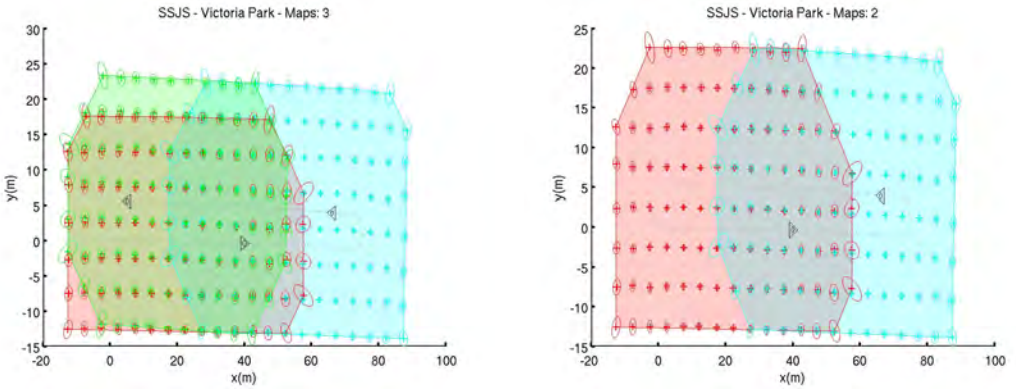
$$\frac{\partial h}{\partial \widehat{\mathbf{x}}} = \begin{bmatrix} \frac{-\Delta x}{\Delta} & \frac{-\Delta y}{\Delta} & 0 & 0 & 0 & \ldots & \frac{-\Delta x}{\Delta} & \frac{\Delta y}{\Delta} & 0 & 0 & \ldots \\ \frac{\Delta y}{\Delta^2} & \frac{-\Delta x}{\Delta^2} & -1 & 0 & 0 & \ldots & \frac{-\Delta y}{\Delta^2} & \frac{\Delta x}{\Delta^2} & 0 & 0 & \ldots \end{bmatrix} \tag{3.20}$$

A scene simulator that allows the user to define the position of the landmarks and the vehicle's trajectory on a plane was implemented. The simulator begin then sensor readings from the odometry and the laser range finder come in. This is, the speed of the vehicle and the steering wheel angle for each time-step of the odometer, as well as the range and bearing of the observation with respect to the vehicle at each time-step of the laser range finder. In addition, the user was prompted to define the amount of zero mean Gaussian noise to be added to every sensor reading. This set-up consisted of a scenario of 100 m^2 populated with 441 landmarks. The vehicle performed a survey trajectory. Figure 3.4(a) shows the ground truth, Figure 3.4(b) shows the dead reckoning with added Gaussian noise and the output of SSJS is shown in Figure 3.4(c). Notice that for the noise levels used in this specific experiment (i.e. laser range finder noises are 10 cm in the range and 1 degree in the bearing and the odometer noises are 1 m/s in speed readings and 7 degrees in steering), the final



(a) Ground truth (b) Dead reckoning (c) SSJS Final Result

Figure 3.4: A synthetic scenario generated using a simulator.



(a) Before joining two maps (b) After joining two maps

Figure 3.5: An example of a map fusion step where two submaps are joined and fused after closing the loop.

drift of the vehicle on the dead reckoning was around 25%, while for the SSJS the drift was less than 2.5%. These results were obtained after tuning the submap size and the *threshold*, that indicates whether to fuse maps as a trade-off between computational cost and accuracy. The map size was set to 15 landmarks per map and the *threshold* was set to 40% of the map size. These sorts of synthetic experiments allowed us to check the precision, consistency and computational cost of SSJS compared to other SLAM techniques.

The overall consistency of SSJS was checked qualitatively and quantitatively. Figure 3.5(a) and Figure 3.5(b) show a loop closing event. Quantitative results where obtained via the Normalized Estimation Error Squared (NEES) statistical test depicted in Equation (3.21).

$$NEES = (\mathbf{x}_k - \widehat{\mathbf{x}}_k)^{\mathrm{T}} \mathbf{P}_k^{-1} (\mathbf{x}_k - \widehat{\mathbf{x}}_k) < \chi^2_{r,1-\alpha} \tag{3.21}$$

Table 3.2: Comparison of time consumption (in seconds and % over the whole mission time.

method	total (s)	local mapping (%)	global mapping (%)	others (%)
EKF	2028.21	99.0	–.-	1.0
LMJ	92.48	76.9	22.9	0.2
DCS	83.73	88.2	11.7	0.1
HS	78.37	91.0	8.6	0.4
CIS	85.26	89.3	7.3	3.4
SSJS	**78.88**	**81.0**	**18.7**	**0.3**

where $r = dim(\mathbf{x}_k)$ is the degree of freedom and α is the desired significance level (usually 0.05) [Paz 2008]. Given the ground truth, the state $(\widehat{\mathbf{x}}, \mathbf{P})$ estimation is consistent when $NEES < \chi^2_{r,1-\alpha}$, otherwise the estimation is too optimistic and becomes inconsistent. Figure 3.6 shows the SSJS algorithm performing within the consistency boundaries.

Regarding time complexity, results in Table 3.2 show that SSJS is faster than other submapping techniques. In LMJ the cost grows exponentially but decreases as the mission gets longer. DCS is faster than LMJ in the submapping stage and even in the joining step, however, the complexity corresponding to the joining step varies with the size of the maps to be joined. This complexity increases the cost considerably and can be very high after several submaps. In fact, Joint Compatibility Branch and Bound (JCBB) does not perform efficiently in these situations because the exploration tree is too large. For this reason, JCBB is substituted for Randomized Joint Compatibility (RJC) to address data association in large maps, such as in DCS in the upper levels of the global level tree. The use of RJC in DCS significantly reduces the computational cost. HS is even faster than LMJ and DCS because the local maps are considerably smaller. However, as the mission gets longer, some of the maps become bigger and the optimization of the global level becomes more complex producing a slight increase in the computational cost. CI-Graph SLAM is slightly slower due to the extra cost caused by the back propagation step and the additional reobserving and revisiting stages. Overall, SSJS and HS have similar computational costs and are faster than the other SLAM techniques surveyed. The time spent during local map building is similar in all these methods, but there is no extra cost arising from global level optimization or from the map fusion stage and because the maps are only fused selectively. They are only fused if they really share enough information to be considered a single map, otherwise they are kept independent and small.

The precision (error) with respect to the ground truth is given in Table 3.3. Table 3.3 shows the maximum, mean and standard deviation of the final map and vehicle position error for every technique. SSJS is one of the most precise thanks to the selective submap

Table 3.3: Mapping and vehicle position precision compared to ground truth (in meters).

method	map error			localization error		
	maximum	**mean**	**std**	**maximum**	**mean**	**std**
EKF	7.9213	0.5891	0.9875	1.5185	0.7272	0.3461
LMJ	4.8144	0.4857	0.6822	1.4635	0.7605	0.3268
DCS	6.2612	0.1260	0.5391	1.5003	0.9006	0.3332
HS	5.5558	0.5761	0.5761	1.4563	0.7477	0.3428
CIS	4.3284	0.1870	0.4568	1.2352	0.5267	0.3156
SSJS	**4.5268**	**0.1960**	**0.3468**	**1.3214**	**0.5372**	**0.3201**

fusion approach. Fusing only those maps that overlap significantly is shown to be positive since the amount of information used in the update step is higher. In addition, the data association between maps becomes more robust since the matches are more constrained for the rest of the features. CIS error is slightly lower compared to SSJS because it uses the conditional independence between submaps to correlate their information, thus all available information is taken into account. However, this conditional independence produces a higher computational cost as reported in Table 3.2.

To sum up, the SSJS algorithm is shown to be computationally efficient and produces a convergent solution. These two properties are very important in long term missions or large scale scenarios on-line solutions.

3.4.2 Real Experiments: the Victoria Park Dataset

Real experiments were conducted using the well-known Victoria Park dataset which uses a 3-DOF terrestrial vehicle. The Victoria Park dataset was recorded by [Guivant 2000a] at the Australian Center for Field Robotics (ACFR) (see Figure 3.7). The Utility vehicle (UTL) shown in Figure 3.8 was used to gather the data. This dataset describes a path through an area of around 197m x 93m. The sequence consists of 7247 frames along a trajectory of 4 kilometres recorded over a total time of 26 minutes. The dataset contains sensor readings from a steering and rear axle wheel (odometry) and a laser range finder (one 360-degrees scan per second), along with the ground truth position data from a GPS. For the laser range data, a tree detector approach is described in Section 4.3.1. The detected trees are used as point features. The motion and observation models for this vehicle and its sensors are presented in the synthetic experiments section. All the experiments were conducted on a Pentium Core Duo 1.77-GHz. The purpose of this experiment was twofold: 1) to evaluate the consistency of the SSJS approach, and 2) to analyse its computational cost. Both were compared with state of the art SLAM techniques.

The final result of the SSJS for the Victoria Park dataset is shown in Figure 3.7, which indicates the level of correction of the approach in qualitative terms. The final map (black trajectory) is almost the same as the one generated using GPS data (grey trajectory). Note that the GPS readings were not used in the SSJS. However, it is not possible to extract any further conclusions such as the consistency of the method or the time consumption. In order to perform a quantitative evaluation, the consistency of the SSJS approach is checked via map consistency analysis. When the ground truth for the state variable is not known, the Normalized Innovation Squared (NIS) (Equation 3.22), can be used as an alternative.

$$NIS = \nu_k^{\mathrm{T}} \mathbf{S}_k^{-1} \nu_k < \chi^2_{r,1-\alpha} \tag{3.22}$$

Given the estimation of the innovation vector ν and the innovation matrix $\mathbf{S}$, the state $(\widehat{\mathbf{x}}, \mathbf{P})$ estimation is consistent when $NIS < \chi^2_{r,1-\alpha}$, otherwise the estimation is too optimistic and becomes inconsistent. Figure 3.9 shows the SSJS constantly performing within the consistency boundaries. The rest of the submapping techniques also perform inside these consistency boundaries, while an standard EKF based SLAM with no submaps does not. It can therefore be concluded that the use of submaps is a must to obtain consistent maps in large missions.

Regarding computational cost, Figure 3.10 shows computing time evolution during the Victoria Park mission, demonstrating that SSJS has a reduction on computing time compared to the other techniques surveyed. This slight improvement can be explained by the fact that these approaches only differe on the strategy to build the global map, which is directly dependent on the association of the data from two maps. There is a significant increase on the data association cost for the MJ and DCS because maps' size grow with the mission. HS, DCS or SSJS solve this issue by keeping maps' size small. SSJS only pays the cost of a single EKF update step during the global mapping, while DCS has an additional back propagation cost and HS has additional cost caused by the minimization process.

3.5 Chapter Summary

The suitability of the SSJS for mapping large scale scenarios has consistently been demonstrated. The main contribution of SSJS is the local map fusion strategy. This strategy is used to determine whether to fuse two local maps based on the amount of information they share. SSJS produced consistent maps using synthetic data and the Victoria Park dataset. In addition, a comparison between the SSJS approach and the state of the art SLAM techniques showing their consistency, computational cost and precision was given.

The computational cost of SSJS outperformed those of other submapping methods. Comparisons between the LMJ, DCS, HS, CIS and SSJS methods were made. Although similar results are obtained in terms of consistency, all the techniques have a higher computational

cost compared to SSJS. This is due to the map joining strategy and the global graph optimization. These steps increase the computing time considerably in LMJ, DCS and HS, while SSJS does not require a global optimization, with a consequent reduction in complexity. The precision of SSJS is similar to that of other submapping strategies. The main reason for this improvement is that not all maps are always fused, as happens with LMJ, DCS and HS. LMJ fuses each submap with the global map, DCS fuses maps following a tree strategy, and HS fuses maps when a loop is detected. All of them fuse maps without considering the number of matches between them, which might not be enough to produce a proper update. Sometimes only very few features overlap and they may not be the most reliable ones. In addition, data association between maps may not be correctly solved due to the lack of jointly compatible constraints. SSJS addresses this weakness by only fusing maps when the amount of common information is large, thereby improving the final update and correctness of the fused map. Only CIS performs slightly better that the proposed solution in terms of consistency, but at a higher computational cost.

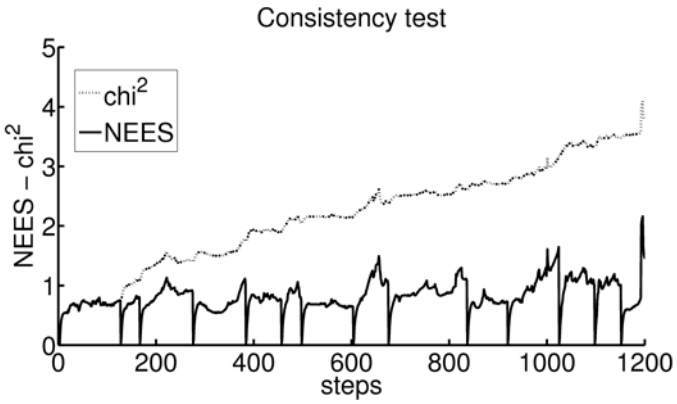


Figure 3.6: Consistency test using the NEES statistical test.

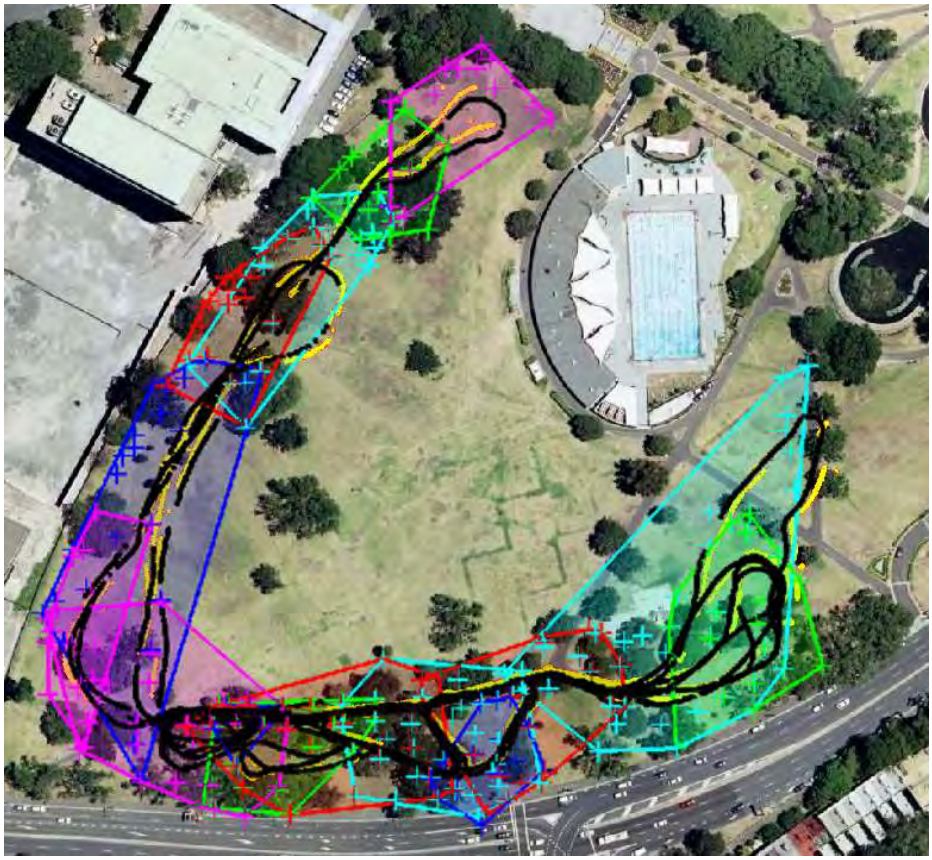


Figure 3.7: Satellite image of the Victoria Park dataset (source: *Google Earth*). The GPS data captured during the mission is drawn in grey, while the vehicle's estimated trajectory is represented in black. The final submaps obtained with SSJS are also shown.

Figure 3.8: Utility vehicle used to gather the Victoria Park dataset. The vehicle is carrying the scanning laser sensor (vehicle left front) and the optical speedometer (left back wheel) (source: [Guivant 2002b]).

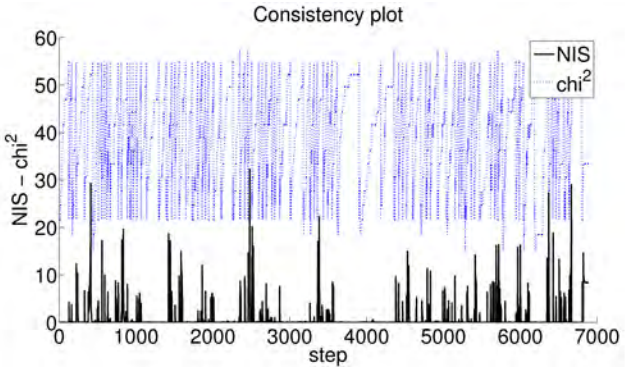


Figure 3.9: Consistency test, where the dashed line represents the χ^2 corresponding to each step and the continuous line corresponds to the NIS.

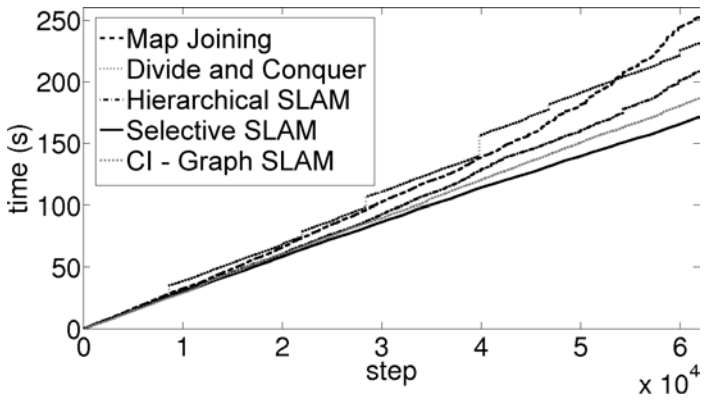


Figure 3.10: Computation demand comparison. Time consumption throughout the Victoria Park dataset is compared for the MJ, DC, HS, CIS and SSJS methods.

Chapter 4

Feature Extraction and Matching

Contents

Feature base SLAM solutions only work properly when new observations can be correctly matched with past information. Once a landmark is observed, it is necessary to determine whether it is observed for the first time or is already inside the map information. The first time a landmark is observed, it is added to the map and will serve as a landmark to relocate the vehicle and correct map information in the event of a second observation, in a similar way humans relocate themselves when they are lost and find a known building, sign or street. Therefore, being able to spot these associations is a key factor in solving the SLAM problem. To correlate these observations it is necessary to characterise a landmark using robust features. Therefore, not only matching, but also feature extraction is very important for SLAM to work consistently. In this chapter, approaches to extract features from three different sensors are presented together with their matching approach. The performance of these approaches is analysed through experimental validation.

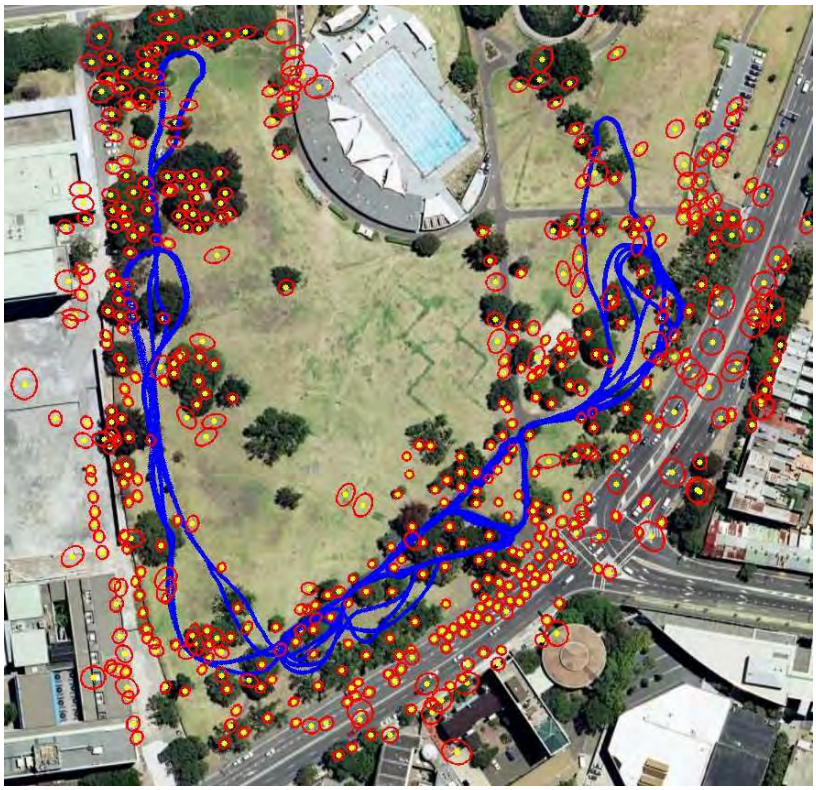


Figure 4.1: Example of a point feature based SLAM approach [Aulinas 2010c]. The image shows the resulting map composed of point features representing trees, together with the SLAM trajectory represented in a satellite image of the scenario.

4.1 Overview

The goal of SLAM is to build a map of an unknown environment while simultaneously determining the location of the robot within this map. This task is still one of the fundamental challenges of robotics and one of its main difficulties is the lack of robust and consistent landmarks. Several methods use features to describe these landmarks and achieve the SLAM purpose. These landmarks are used as references for the localization problem and as map elements on the mapping problem. For instance, a point feature based SLAM solution was presented in [Aulinas 2010c], as shown in Figure 4.1. In this specific experiment, a vehicle equipped with a laser range finder navigates a park full of trees, which are then represented as point features. Other approaches like the one in Figure 4.2 propose a line feature based SLAM approach [Ribas 2008b]. In this example, an AUV navigates an abandoned marina, in which line features are the best choice to represent the boundaries between water and land. Finding a proper representation for these features is a key factor in solving a feature based SLAM problem. These features must be very robust, in order to ensure that the same feature will be observed again once revisited. Associating a new observation with a

Figure 4.2: Example of a line feature based SLAM approach [Ribas 2008b]. The image shows the resulting map together with the dead-reckoning (dash-dotted line), GPS (dashed line) and SLAM (solid line) trajectories represented in a satellite image of the scenario.

previously seen feature is the key to improving the vehicle's localization and the final map. Instead, wrong associations would introduce divergences and inconsistencies in the results and the consequent loss of the vehicle.

The aim of this chapter is to describe the computer vision algorithms used to extract robust features and to detect objects using different sensors on-board different vehicles. For instance, the utility vehicle used to acquire the Victoria Park dataset carried a laser range finder, the REMUS AUV uses a side-scan sonar as its main remote sensing device, and the SPARUS is equiped with a down-looking optical camera.

4.2 Laser Range Finder Data

The Victoria Park dataset used to compare submapping methods in Chapter 3 contains laser range finder readings. This information explains the environment, which in this particular case, is populated with many trees. A laser range finder is similar to radar sensors in the sense that they both provide range and bearing measurements (see Figure 4.3). Following the same strategy presented in [Guivant 2002a], trees were described by point features in the centre of the trunk (see Figure 4.4(b)) and then they were used as landmarks inside the SLAM algorithm.

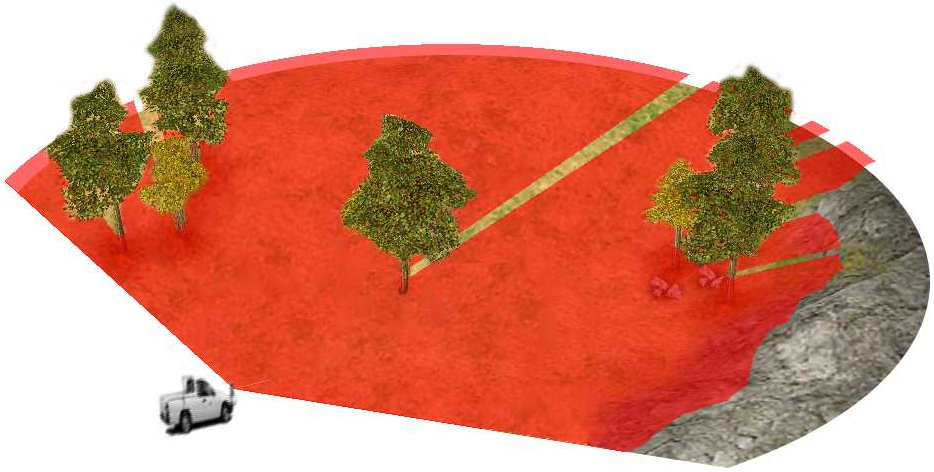


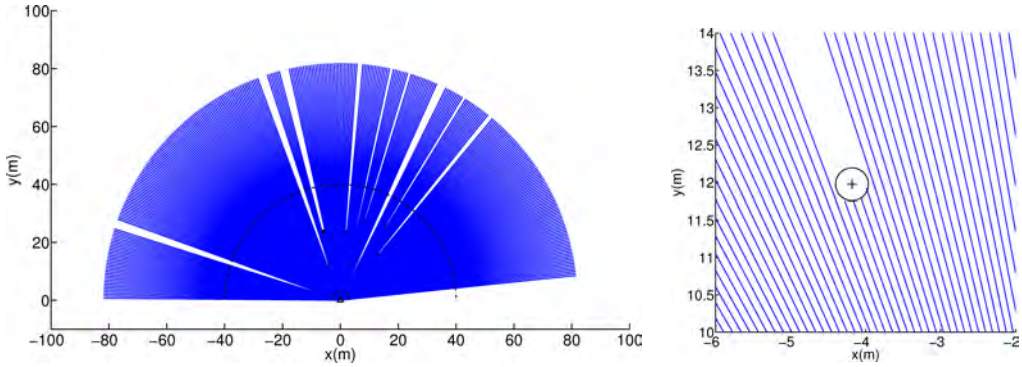
Figure 4.3: Laser range finder observation principle. The laser range finder reads a number of beams covering 180 degrees. In this example, the trunk of trees are within the laser's range.

4.2.1 Feature Extraction

The data provided in the Victoria Park dataset contains one laser reading per second. This per second reading is composed of 360 values corresponding to the range of a semicircular measurement, i.e. the bearing resolution is half of a degree. The maximum range in this data is around 80 meters, see Figure 4.4(a). The uncertainty of the measurement increases with distance due to bearing inaccuracies, therefore, only trees within a certain range are taken into account, i.e. further than 3 meters and no more than 40 meters. At this distance, more than one laser beam hits the same tree generating a semicircular shape, as shown in Figure 4.4(b). The strategy used here to define an estimation of the trunk's centre point is explained in Figure 4.5. This figure shows a zoom in the region of a tree. In this case, one can observe a sequence of laser beams, some reaching the range limit of 40 meters, while some stop before because there is an object in its way. The two laser beams at the boundaries of the object are emphasised in red. These are assumed to be tangent to the trunk's surface, i.e., the circle. Therefore, the centre point of the trunk should be on the bisectrix between these two lines (dashed black). The radius of the trunk is then estimated by finding the cord (dotted black) that connects the two outer most beams of the tree (in green). This cord is assumed to be a first approximation of the trunk's diameter. The radius is then added to the range of the tree's middle beam (in yellow).

4.2.2 Feature Matching

In order to associate those trees that are already part of the map with new observations, the only information used is their global position (x, y) in the map. The radius of the tree



(a) Top view of a laser scan (b) Zoom in on a object

Figure 4.4: Top view of a laser scan. a) This particular example contains several objects that cut the laser beam producing a circular shape, as shown in b).

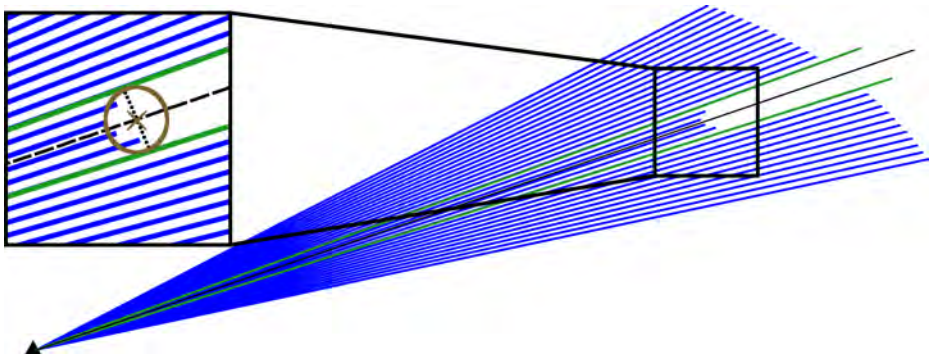
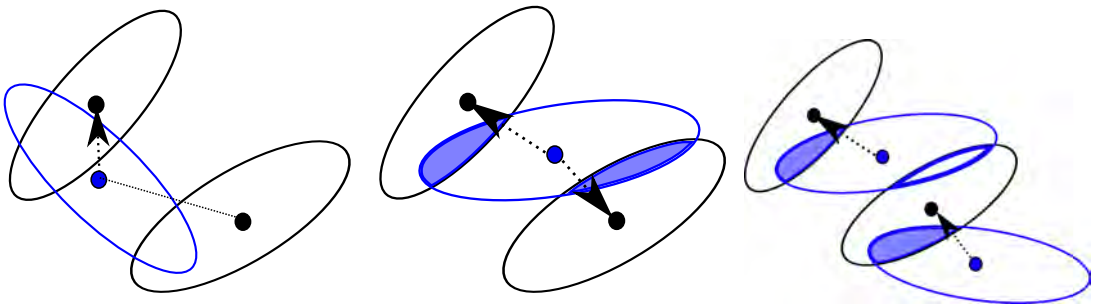


Figure 4.5: The trunk center estimate conceptual idea. The triangle on the bottom represents the car's orientation.

was initially thought as a distinctive feature for each tree. After several trials, the radius was discarded because it was not robust when observed from a different point of view or from different heights. For this reason, existing data association algorithms from the literature were tested and used on this dataset. The simplest criterion to find an association for a given measurement is the Nearest Neighbor (NN) method. NN is based on associating pairs with the smallest Mahalanobis distance (See Figure 4.6(a)), based on their covariances. The Mahalanobis distance normalises feature-landmark distances based on their covariance. This approach performs properly when the distance between features is large. However, as the vehicle keeps moving and the uncertainties grow (see Figure 4.6(b)), this method leads to misclassification because the most compatible pairing from a global point of view (i.e., taking into account all the features) is never recomputed. The JCBB algorithm [Neira 2001] addresses this issue by considering the compatibility of all the pairings globally (see Figure 4.6(c)). JCBB is very robust because it considers relative locations between features, but there is a higher computational cost associated with the number of pairings. This in-



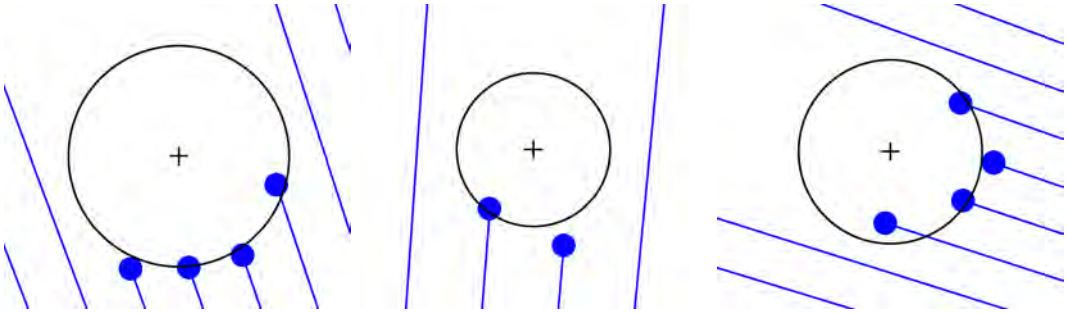
(a) NN (b) NN failure (c) JCBB

Figure 4.6: Data association examples: points represent landmarks, ellipses represent uncertainty and arrows represent the association between landmarks. a) example of easy association; the closest one is easy to identify using the NN algorithm; b) example of a ambiguous situation with two possible associations of a single new observation, in this example it is not clear which is the nearest neighbour; c) but if one zoom out and consider other points and associations from a global point of view, then it becomes clear which are the right associations. This is an example of JCBB, where the ambiguity in b) is now solved by considering global associations.

crease in cost is not a problem when building local maps because the number of features to be associated is always limited by the local map size. However, data association is a critical issue during map joining because map joining involves finding correspondences between two local maps. As the ambiguity increases the cost to solve the compatibility between the two maps becomes very expensive. In order to limit the computation cost of the data association between local maps, a RJC [Paz 2007a] algorithm was created. The RJC approach is a variant of the linear relocation used for global localization [Neira 2003]. RJC first identifies the overlap between two maps using individual compatibility (i.e., Nearest Neighbour). A set of b overlapping features is then randomly selected and associated through JCBB. Associations for the remaining features in the overlapping area are obtained using NN. This process is repeated several times so that the probability of missing a correct association is limited.

4.2.3 Experimental Validation

The performance of this method was tested on the Victoria Park dataset. Figure 4.19 shows several examples of circle approximation. In addition, Figure 4.1 shows a final map generated through SLAM automatically detecting the trees in the park. The data association algorithm employed in this particular example was the JCBB, producing consistent associations all along the path. In addition, spurious detections, such as moving elements or noisy measurements, were properly spoted by the JCBB due to its incompatibility from a global point of view.



(a) case 1 (b) case 2 (c) case 3

Figure 4.7: Examples of the performance of the extraction of the tree's center.

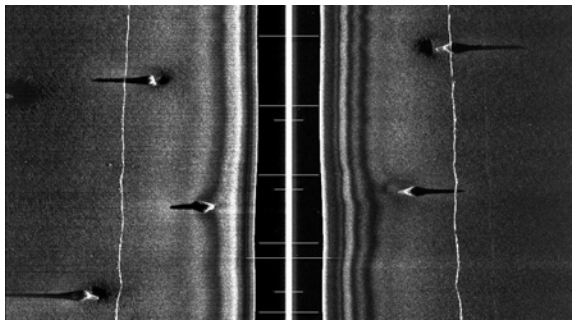


Figure 4.8: A side-scan sonar image example with five objects shown as bright spots with their corresponding shadows.

4.3 Side-scan Sonar Imaging

This section describes an approach able to detect objects in side-scan sonar images on-board an AUV. Side-scan sonar provides high resolution acoustic images in which an object appears as a bright spot with a dark shadow trail (see Figure 4.8). In order to have a fast and robust object detector, an approach which is based on the Viola and Jones solution [Viola 2001] is proposed. In this approach a cascade of classifiers is used to detect faces with high detection rates. Following the same idea but for the detection of specific objects observed from a side-scan sonar and with a proper feature selection, the detection is significantly improved, providing high detection rates and very low false positives. The performance of the detection method is shown to be a proper input for the SLAM algorithm, providing a consistent map and a correct vehicle localization.

The following sections describe the principles of the object detector approach for side-scan sonar images. First, a initial approach is described. Then, a discussion on the best suited features for side-scan sonar images is summarized. Afterwards, basic concepts related to the detection process are introduced. Finally, the training procedure is briefly described.

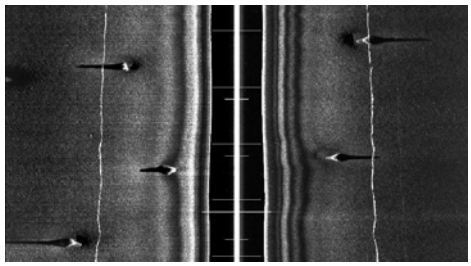
4.3.1 Feature Extraction

Initially, the approach to object detection is based on preprocessing the image followed by a threshold, as shown in Figure 4.9 [Aulinas 2010a]. First, a median filter is applied to remove the salt and pepper effect of the sea floor (Figure 4.9(b)). Second, a low intensity threshold is used to binarize the image and find shadows (Figure 4.9(c)). The resulting image is run through the 'erode' morphological operation in order to magnify shadow sizes, while at the same time joining small noise areas. In this way, only dark regions of a certain size, which depends on the image range, are accepted as object shadow candidates (Figure 4.9(e)). Third, the process is repeated for bright spots. A high intensity threshold is applied in order to binarize the image and find highly reflective metallic objects (Figure 4.9(d)). In order to magnify these small spots, a 'dilate' is applied and these spots are selected as possible object candidates (Figure 4.9(f)). Finally, only those areas with both shadow and metallic object candidates are accepted as real objects (Figure 4.9(g)).

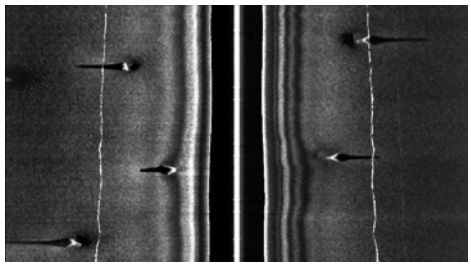
The Viola-Jones object detection framework is capable of providing competitive face detection rates [Viola 2001]. It can be trained to detect a variety of object classes. The Viola-Jones cascade of classifiers introduces a new algorithm to construct a robust classifier [Viola 2004]. They use Haar-like features which are really simple (see Figure 4.10), fast and cheap to compute using the integral image. All these features rely on more than one rectangular area. The value assigned to this rectangular area is the sum of the pixels within clear rectangles subtracted from the sum of pixels within shaded rectangles. The advantage of these features is that they are sensitive to vertical and horizontal changes. With the use of the so called integral image, rectangular features can be evaluated in constant time. The integral image at location x, y contains the sum of the pixels above and to the left of x, y (see Figure 4.11). Using the integral image any rectangular sum can be computed as shown in Figure 4.12. Therefore, properly selected Haar-like features encode the oriented contrasts between regions in the image and give a quantity for the presence or absence of contrast characteristics at a specific image location.

Features are usually more efficient to process than having to process the whole intensity image. In side-scan sonar images, objects are simple (see Figure 4.13), which means that they can be described by using only few features. Moreover, the performance of the classifier is effected by the number of features. In order to improve this performance, a feature selection is necessary. In this approach, two different types of features have been used: Haar-like features and the distance from the boundary of the object to its centroid. The advantage of these features is their invariance when face with rotations, scale changes, intensity shifts and translations.

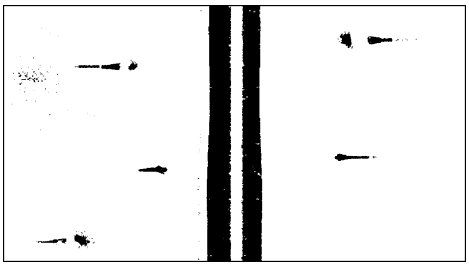
In this proposal, rectangular shaped features are used. Although rectangle features are of limited flexibility, they provide a rich image representation which makes them extremely computationally efficient. The rectangle is divided into two regions: a dark one on the left



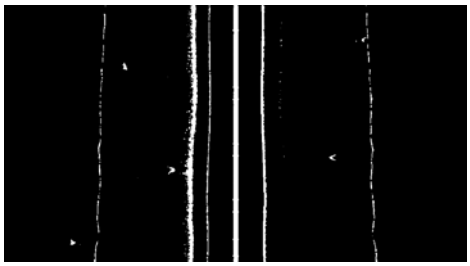
(a) original image



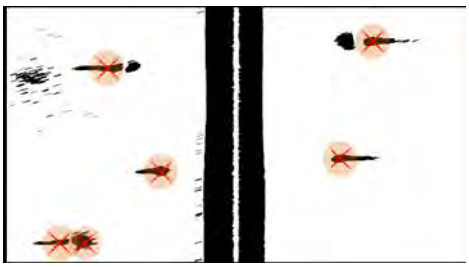
(b) image after median filter



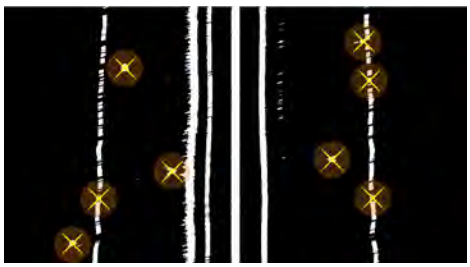
(c) high intensity threshold binary image



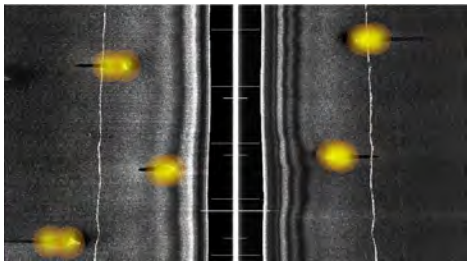
(d) low intensity threshold binary image



(e) shadow candidates detection



(f) bright spot candidates detection



(g) detection = shadows + bright spots

Figure 4.9: Example of automatic detection performance.

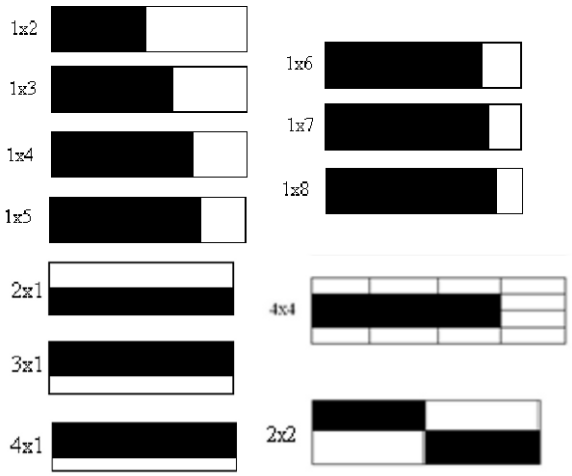


Figure 4.10: Haar-like feature examples.

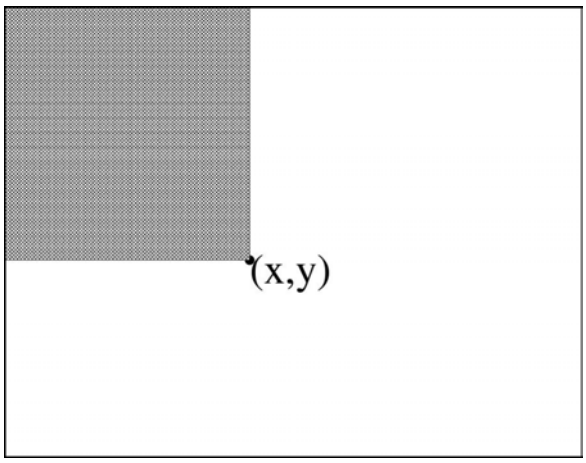


Figure 4.11: The value of the integral image at (x, y) is the sum of all pixels in the shades area [Viola 2001].

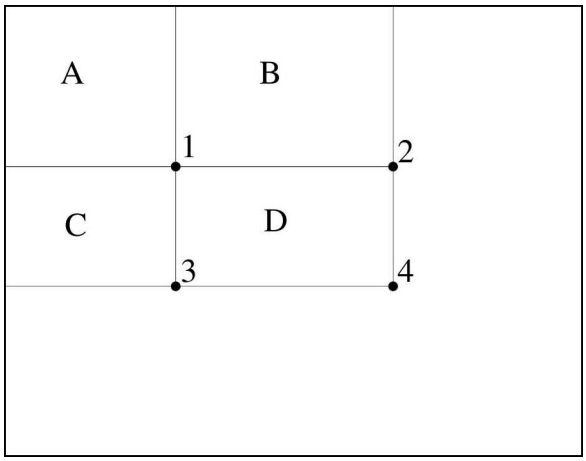


Figure 4.12: The sum of pixels within rectangle D can be computed as the combination $4+1-(2+3)$ of the integral images at points 1 to 4.

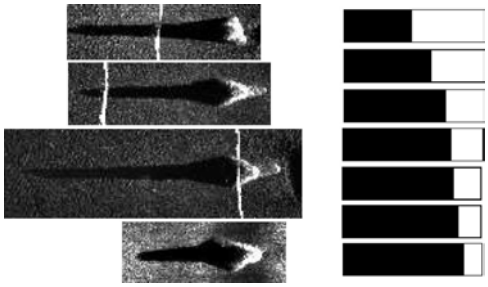


Figure 4.13: The left column shows real objects as seen in a side-scan sonar image. The right column shows those Haar-like features best suited to train the sort of real object from the left column.

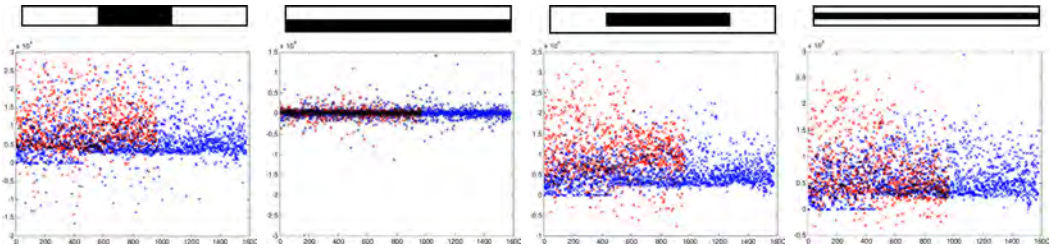


Figure 4.14: Comparison of the detection performance using different features not suited for side-scan sonar objects. These features do not distinguish between true and false positives, due to the fact that the orientation from which they discriminate is not the one from the objects in the side-scan sonar images. The integral image results for the positive training sample are in red, and in blue the negative ones.

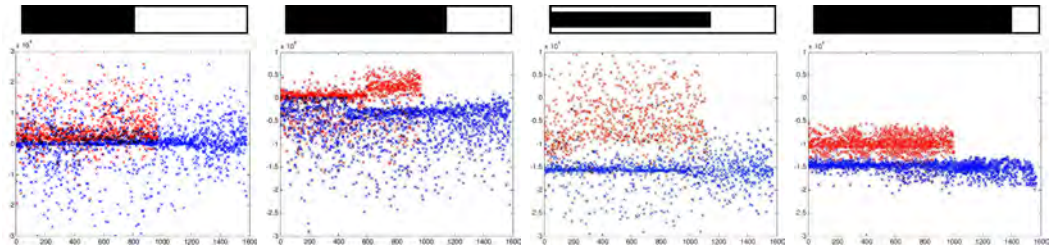


Figure 4.15: Comparison of the detection performance using different features well suited for side-scan sonar objects. These features produce proper distinction of true and false positives. They are very similar to real objects in side-scan sonar images, but with different proportions.

and a bright one on the right. The proportion of each region is different in each feature (see Figure 4.13). The value of a two-region feature is the difference between the sum of the pixels within the two regions. Figure 4.14 represents a set of features, some of them not suited for the problem presented in this work. In contrast, Figure 4.15 shows a set of features well suited for side-scan sonar objects. These plots represent the ability of the system to distinguish between true and false positives [Aulinas 2011]. The system used to run these tests is a single weak classifier analysed using a different feature in each case. Each plot is for a specific feature and shows the value obtained for each training sample after computing the integral image on the corresponding feature. After analysing different feature shapes and proportions, it seems that the most discriminative one is the one with the proportion 8 dark to 1 bright because the values obtained for the positive set (in red) are easily separable from the ones obtained for the negative set (in blue).

Another significant feature is the distance from the boundary of the object to its centroid, as in [Atallah 2005]. This feature not only improves a classification stage, but also the object detection.

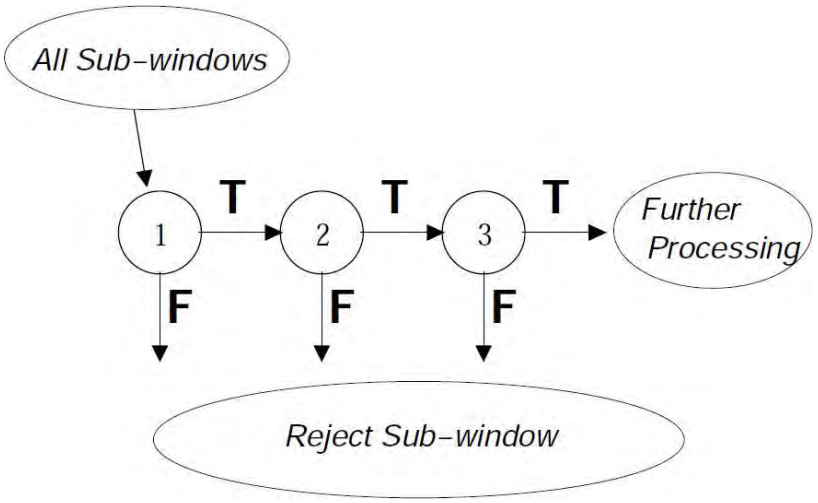


Figure 4.16: Schematic depiction of the detection cascade.

4.3.2 Feature Matching: the Cascade of Classifiers

A cascade classifier is a sequence of simple classifiers (see Figure 4.16 from [Viola 2004]). The main idea behind a cascade classifier is to detect and reject background information quickly. The initial classifier eliminates a large number of negative examples with very little processing. As the detection goes deeper in the cascade, a higher number of features are used to reject negative objects. Each layer has a strong classifier which is composed of one or more weak classifiers. It tries to keep high detection rate in all the layers by decreasing the threshold of the strong classifier. Classifiers and their relevant features are selected using AdaBoost [Freund 1995].

4.3.3 Training Stage

A classifier with more features produces higher detection rates and lower false positives. However, using more features means a higher computational cost. For this reason it is necessary to find a trade-off between computational cost and detection rates. The optimal solution for a SLAM problem would be to detect all existing objects in the scene, therefore a detection rate of a 100% would certainly be optimal. However, forcing the system to detect as many real objects as possible will introduce false detections. For the SLAM algorithm to work properly, this false positive rate must be kept to a minimum because detecting a false object could produce a wrong data association and its consequent inconsistencies in the map and the localization. In addition, on-line SLAM solutions demand fast detectors, therefore only a few features should be used.

From all these requirements, the main constraints to be met during the training process are summarized in Table 4.1. As explained in Table 4.1, each layer tries to keep a high

Table 4.1: Training constraints.

Cascade layer	1	2	3	4	...
Detection rate	99.9%	99.7%	99.6%	99.5%	...
False positive rate	50%	20%	10%	5%	...
Number of features	1	2	6	11	...

Cascade layer	...	5	6	7
Detection rate	...	99.5%	99.3%	99%
False positive rate	...	1%	0.5%	0.05%
Number of features	...	20	30	30

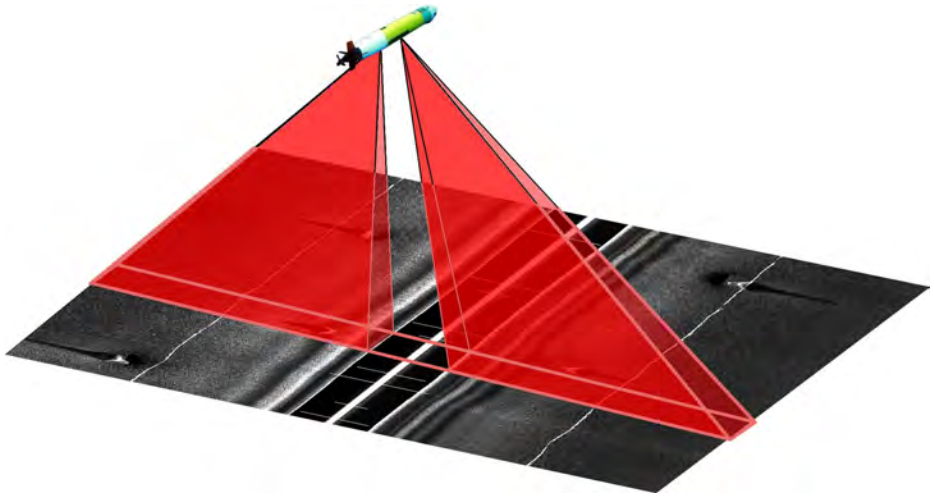


Figure 4.17: The side-scan sonar working principle.

detection rates while maintaining low false positive rates. In order to achieve this objective, it is necessary to add more features to obtain a stronger classifier. The detection rates are determined by testing the current detector on a validation set. If the overall target false positive rate is not yet met then another layer is added to the cascade.

4.3.4 Experimental Validation

The experiments were conducted on a real environment dataset. This dataset was acquired with a REMUS-100 AUV. The sea floor was populated with objects, rocks and other salient features. The vehicle was carrying a side-scan sonar pointing both ways, starboard and port (see Figure 4.17). This side-scan sonar acquired high resolution acoustic images, like the one shown in Figure 4.8.

The cascade of classifiers was trained with the feature type explained in Section 4.3.1. This cascade algorithm was the one in charge of detecting the objects, rocks and other salient features. For the training stage a data set of side-scan sonar image patches was used, some

Table 4.2: Cascade Performance.

Cascade layer	1	2	3	4	...
Detection rate	99.5%	99.3%	99.2%	99%	...
False positive rate	53%	6%	1%	> 1	...
Number of features	1	2	4	> 1	...

of them containing an object (positive set) and the rest without (negative set). According to the behaviour of the classifier, it is recommended to use twice the amount of negative samples than positive ones. In this case, 2000 positive objects and 5000 negative objects were used. Initially the dataset was much smaller, but the performance of the system was not satisfactory. The cascade only improved the false positive rate until a certain layer. Then there was no further improvement, not even after increasing the number of features. For this reason, it was necessary to generate more training images from the original ones by changing their scale, their intensity and adding synthetic noise. This augmented dataset produced better results. In addition, during the training stage, several false object detections were removed from the dataset, changing and improving the cascade structure.

Notice that a different set of side-scan sonar images was used to test the performance of the proposed approach, giving the results in Table 4.2. These results show that the obtained cascade is only useful until its fourth layer because further layers suffer an increase in the false positive rate and in computational cost. This means that there is a chance of detecting a false object every hundred observations, which could be a real issue for the SLAM problem. For this reason, the JCBB was used to perform a double check on spurious observations.

4.4 Down-looking Optical Camera

AUV are gaining importance on intervention missions. In order to conduct such tasks autonomously, it is necessary to have precise and accurate information about the scene. To achieve this goal, computer vision algorithms are necessary to enable 3D reconstructions from underwater imagery. These algorithms must extract robust features from underwater imagery and perform reliable feature matching.

AUVs typically use acoustic sensors to gather data from the environment: echo sounder or multi-beam echo-sounder, multi-beam imaging sonar, forward-looking imaging sonar and side-scan sonar. However, the use of such acoustic devices does not give any intensity information, which might be necessary on intervention missions to detect specific objects, or might be useful when navigating through shallow waters.

The SPARUS AUV is equipped with a down-looking camera, as shown in Figure 4.18. This camera acquires three frames per second, like the one shown in Figure 4.19(a). These images contain regions of interest with salient features, as shown in Figure 4.19(b). These

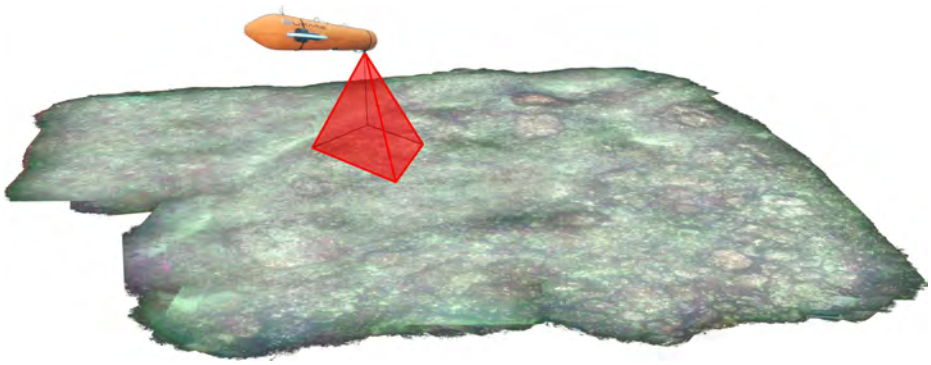


Figure 4.18: The working principle for the SPARUS down-looking camera.

(a) Original image (b) Salient features (c) SURF features

Figure 4.19: A underwater image and its salient features.

salient features will then be used as landmarks in SLAM. The idea behind the landmark detection used for the SPARUS dataset is based on using two sources of information: 1) context information to characterize the Region of Interest (RoI); and 2) features extracted from these RoIs. Contextual information is obtained by segmenting the image into background and RoI. Features extracted from RoI are computed by using common feature extractors such as SIFT and SURF 4.19(c). This information is then used together with the topological location of these landmarks to match new observations with known landmarks obtained from previous observations. The method is evaluated through experimental validation on a real unstructured underwater environment using the SPARUS AUV.

4.4.1 Feature Extraction

Features are selected to provide robustness when faced with a certain degree of distortion so that the same point can be detected when observed from a different point of view. The feature extraction procedure is shown in Figure 4.21. The process starts with an image preprocessing stage. Preprocessing consists of single channel selection, i.e., gray, followed by non-uniform light correction and a normalization. These preprocessing steps are done by common filtering techniques, in this particular case the ones presented in [Gracias 2008]

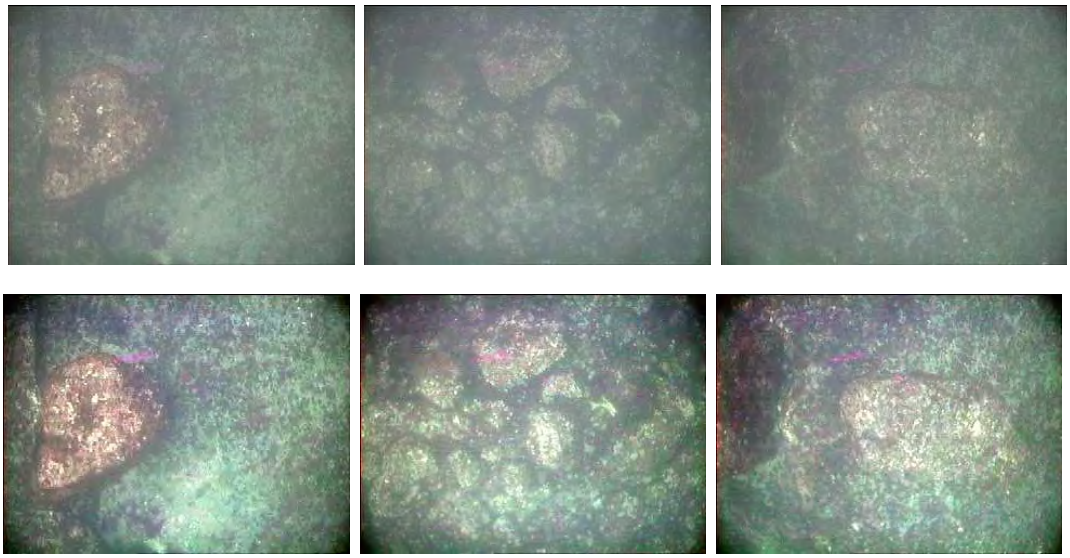


Figure 4.20: Three different examples showing original images taken by SPARUS' camera (top row) and its corresponding preprocessed image (bottom row).

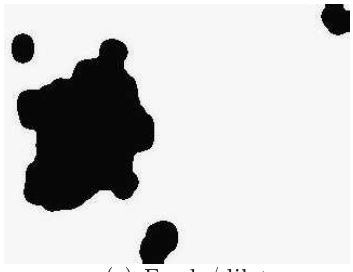
are used. Results from this preprocessing step are shown in Figure 4.20. In addition, lens distortion is corrected, using the calibration obtained with the well known Bouguet's calibration toolbox [Bouguet 2009].

The second stage is focused on detecting RoIs within these images, i.e. segmenting RoIs. In order to do so, two parallel segmentation lines are computed. Both are based on common image processing operations. The first line starts with edge detection, producing the binary image shown in Figure 4.21(b). Afterwards erode/dilate operations are conducted joining regions and eliminating insignificant spots (see Figure 4.21(c)). The next step is a region search within this black and white image producing the segmentation shown in Figure 4.21(d). On the other hand, the second line uses the Hue channel (see Figure 4.21(e)). The hue channel is a well known photometric invariant with respect to both shading and highlights. It is the most invariant channel against illumination changes [Momchilova 2007]. This channel is then blurred in order to smooth the whole image. Afterwards, a threshold is applied, giving the results shown in Figure 4.21(f)). This threshold is automatically selected according to the mean value of the Hue image. Afterwards, a region search is conducted producing the results shown in Figure 4.21(g). At this point both lines are fused, this is, a RoI is selected through the intersection of both segmentations (see Figure 4.21(h)).

The third stage uses SURF features (see Figure 4.21(i)). Depending on the previous stage, if a RoI exists then SURF features are extracted within this RoI and associated with it. Otherwise, if no RoI was segmented, SURF features are extracted from the whole image and stored according to the camera's pose from the moment they were extracted for further

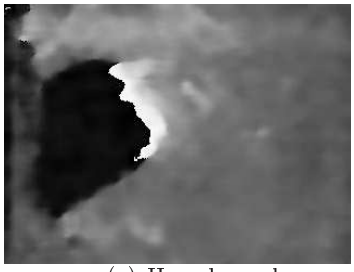
(a) Pre-processed image

(b) Edge detection



(c) Erode/dilate

(d) RoI-1

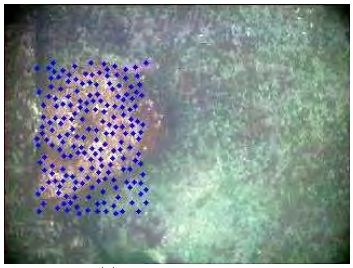


(e) Hue channel

(f) Binary image

(g) RoI-2

(h) RoI intersection



(i) SURF features

Figure 4.21: Procedure to extract regions of interest (RoI). The final selected RoI is the one shown in red in (h) and its SURF features are shown in blue in (i).

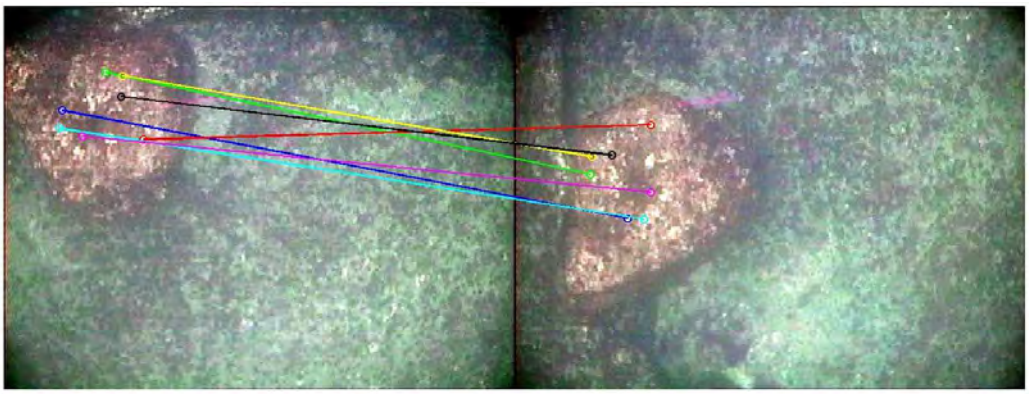


Figure 4.22: This example shows two different observations of a rock. SURF features extracted from the ROIs are matched. The ones that match in both images are connected with a line. The stone is observed from different angles, i.e, rotated almost 90 degrees clockwise.

matching when the camera revisits the same area.

4.4.2 Feature Matching

The output of the feature extraction step is a set of keypoints with their features and descriptors for every image. These features have to be matched. Traditionally, the cross correlation between two image patches was used, but this metric is weak when faced with slight rotations or scale variations. A common practice is to match these keypoints from two separate images based on the similarity of their descriptors, i.e., the Euclidean distance between descriptor vectors. This approach is inclined to find correct pairings, however, many features will not have a match because they either belong to the background or they were not detected in the second image. For this reason, SIFT and SURF matching algorithms use the same distance together with a comparison of neighbouring features, making the matching more robust.

The matching approach used in this work is as follows. First, map information is used to obtain a first approximation of pairable candidates, meaning, the three-dimensional position of a landmark and its uncertainty are the first constraint. Therefore, only new observations whose uncertainty intersects with the known landmark's uncertainty are checked as possible pairing candidates. Initially, only a few landmarks are in the map and their uncertainties might be small, producing only one candidate. However, as the mission continues, more landmarks are added to the map and uncertainties may be larger. At this point, more than one pairing candidate will be found and more information is necessary to find the correct match. Therefore, the SURF matching algorithm is used to discard false matchings (see Figure 4.22).

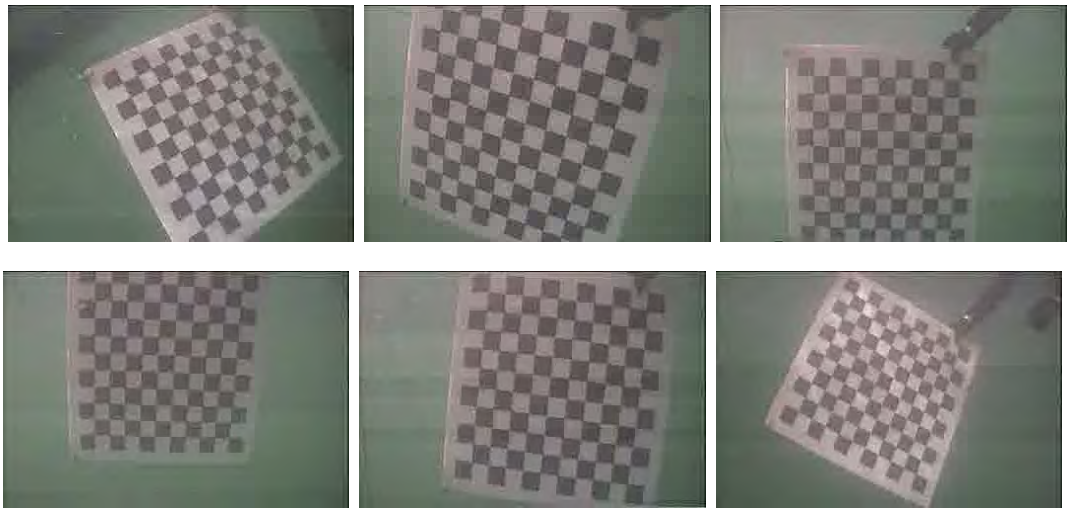


Figure 4.23: A subset images used to calibrate the camera on-board the SPARUS AUV.

4.4.3 Experimental Validation

Experiments were conducted on a sequence of images acquired by the down-looking camera on-board the SPARUS AUV. This sequence was composed of 3199 frames of size 320×240 pixels at a rate of 3 frames per second. Firstly, camera calibration parameters were obtained using the sequence of images shown in Figure 4.23. Available images for calibration were considerably noisy, producing large calibration uncertainty, such as about ten pixels uncertainty for the principal point location. The calibration was used to correct lens distortion.

Secondly, RoIs and features where extracted from the sequence of frames, in this case RoIs where found in 627 images (i.e., 19.6% of the whole set of images), which meant that the greater part of the scenario was either flat or without landmarks. Thirdly, the matching solution performance was evaluated through the SLAM performance; correct matching produced proper map estimates.

4.5 Chapter Summary

In this chapter, three different approaches to detect robust features were presented. The first approach was dedicated to solving the detection of trees from laser range finder readings. Then, different data association techniques were evaluated, reaching the conclusion that the JCBB and the RJC are the most suitable for point feature based SLAM problems. JCBB is very robust with a high computational cost when the number of parings is high, while the RJC is much faster without loosing accuracy. In the second approach, a selection of features that improved the detection of objects in side-scan sonar images was presented. The method used the distance to the centroid and an accurate selection of Haar-like features.

This object detection strategy was used to segment objects located on the sea floor observed through side-scan sonar. These objects were then used as landmarks for the SLAM problem. The third approach, presenting the idea behind feature extraction and matching on optical camera images. The core of this approach relied on SURF feature extraction and its corresponding matching algorithm. After analysing every approach separately, the results performed satisfactorily. The detection methods were shown to provide high detection rates, but notice that these tests were conducted off-line, therefore they need further improvement to become real-time solutions.

Chapter 5

Underwater SLAM

Contents

This chapter presents experimental results obtained with two different AUV: REMUS-100 and SPARUS. The former carries a side-scan sonar as its main remote sensing unit, while the later used a down-looking optical camera. Their corresponding motion and observation models are presented in this chapter. Later on, their validation is presented through the performance of the SSJS.

5.1 Overview

Submapping SLAM techniques demonstrate that using submaps both linearization errors and computational cost can be addressed at the same time, by improving the consistency of EKF SLAM [Castellanos 2007]. Only few of them have been tested in underwater scenarios where some extra constraints have to be taken into account. Firstly, seabed sensing is limited to either acoustics [Ribas 2008b] or near-field vision [Eustice 2008], because electromagnetic waves are strongly attenuated in the water. Secondly, underwater scenarios are in general unstructured and require 3D navigation (6-DOF motion), while most current SLAM solutions are used on man-made (geometrically simple) indoor spaces where a 2D map representation is sufficient. Therefore, the use of SLAM in AUV navigation requires further testing and

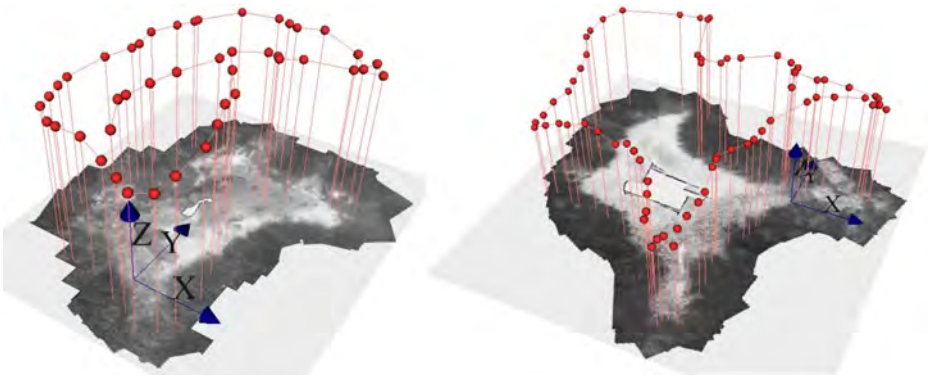


Figure 5.1: Underwater Mosaicking example. This figure shows the perspective view of two mosaics used for the underwater navigation tests with original camera path reconstruction. The small dots mark the 3D positions of the camera centres for the image set selected to create the mosaic. The world referential is represented by the 3 perpendicular axes in the upper right of the image [Gracias 2003].

improvement. On the other hand, one can find several projects that tackle the mapping problem or propose map aided navigation solutions through the use of 2D mosaicks (see Figure 5.1). Instead, the example given in Figure 5.2 produces 3D mapping and localization by the means of structure from Motion (SFM) algorithms.

In this chapter, two good examples of underwater SLAM implementations are described. The implementation of the SSJS on two different AUV is detailed and the results are described.

5.2 SSJS on a REMUS-100 AUV

The AUV REMUS-100 in Figure 5.3 was used to gather experimental data. REMUS-100 was equipped with a Doppler Velocity Log (DVL) and an Inertial Measuremet Unit (IMU), giving navigation data relative to the vehicle's reference frame such as velocities, orientations and depth. In addition, the vehicle was carrying a side-scan sonar pointing both ways, starboard and port. From the navigation information provided by the sensors, the vehicle's state can be defined by a 9-vector composed of the 6-DOF vehicle's pose $\mathbf{x}_V = (x\ y\ z\ \phi\ \theta\ \psi)^{\mathrm{T}}$ and the vehicle's frame linear velocities $(v_x\ v_y\ v_z)^{\mathrm{T}}$.

The map was composed of objects, rocks and other detectable features, especially manta type underwater mines. Therefore, this approach could be useful on MCM applications. The features' state is defined as a 3D points $\mathbf{x}_l = (x_{l_i}\ y_{l_i}\ z_{l_i})^{\mathrm{T}}$. Notice that the 3D point of an object represents the gravity centre of the object. These features are extracted from side-scan sonar images. In addition to feature information, side-scan sonar provides a measure of altitude, i.e. distance from sensor to seabed d_{s-s}. This distance is stored in the state

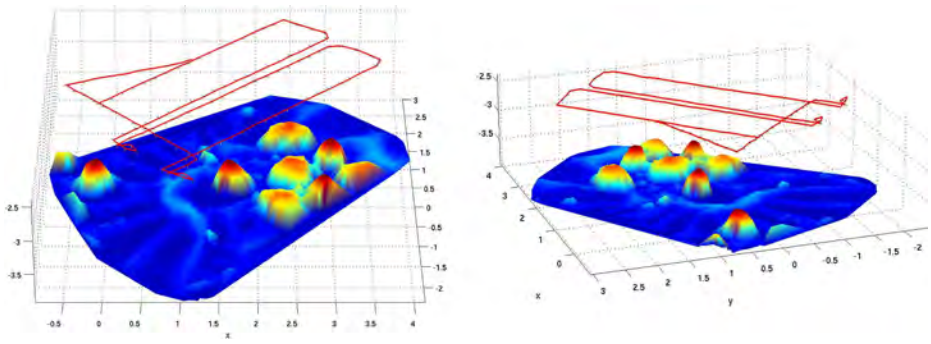


Figure 5.2: Example of 3D reconstruction by means of Structure from Motion algorithms. These plots show two different views of positions and structure for the JHU tank. The camera positions are connected by a red line. A Delaunay triangulation interpolates a surface between the 3D feature points. The structure is color-coded according to height. Units are in meters [Pizarro 2004].

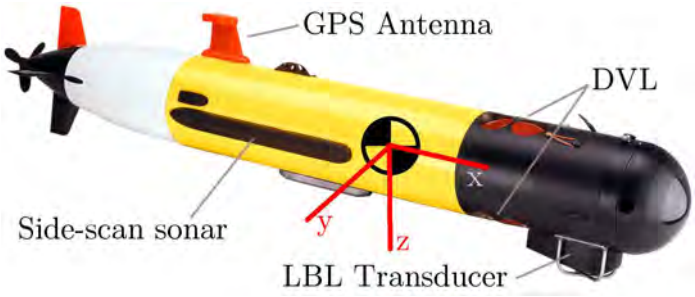


Figure 5.3: REMUS 100 and some of the on-board sensors. The vehicle's reference frame is also illustrated in the image.

vector and used to give a first estimate of the landmark's depth. Notice that the seabed where this mission took place was considerably flat, i.e. planarity assumption gives a first landmark's depth estimate. The joint state vector estimate $\widehat{\mathbf{x}}$ for this experiment contains both the vehicle's state and map information, as shown in Equation (5.1).

$$\begin{aligned}\widehat{\mathbf{x}} \;=\; & (x_V \;\; y_V \;\; z_V \;\; \phi_V \;\; \theta_V \;\; \psi_V \;\; v_x \;\; v_y \;\; v_z \;\; d_{s-s} ... \\ & ... \; x_{l_1} \; y_{l_1} \; z_{l_1} \; ... \; x_{l_i} \; y_{l_i} \; z_{l_i} \; ... \; x_{l_n} \; y_{l_n} \; z_{l_n})^{\mathrm{T}}\end{aligned} \tag{5.1}$$

5.2.1 Motion Model

The motion model for REMUS-100 is a 6-DOF constant velocity kinematics model as shown in Equation (5.2). Where ${}^{\mathbf{x}_{k-1}}\mathcal{R}_{\mathbf{x}_k}$ is the rotation matrix necessary to go from instant $k-1$ to instant k (as explained in Appendix A) and dt is the time increment from step $k-1$ to step k.

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_k) = \begin{bmatrix} x_k \\ y_k \\ z_k \\ \phi_k \\ \theta_k \\ \psi_k \\ v_{x,k} \\ v_{y,k} \\ v_{z,k} \end{bmatrix} = \begin{bmatrix} {}^{\mathbf{x}_{k-1}}\mathcal{R}_{\mathbf{x}_k} \begin{bmatrix} v_{x,k-1}dt \\ v_{y,k-1}dt \\ v_{z,k-1}dt \end{bmatrix} + \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ z_{k-1} \end{bmatrix} \\ \phi_{k-1} \\ \theta_{k-1} \\ \psi_{k-1} \\ v_{x,k-1} \\ v_{y,k-1} \\ v_{z,k-1} \end{bmatrix} \tag{5.2}$$

This motion model $f(\mathbf{x}_k, \mathbf{u}_k)$ is a non-linear function, thus it needs to be linearized in order to fit the linearization assumption of EKF. The linear version is denoted by $\mathbf{F}_k$ and is computed by taking the jacobian of $f(\mathbf{x}_k, \mathbf{u}_k)$, as in Equation (5.3).

$$\mathbf{F}_k = \left.\frac{\partial f}{\partial \mathbf{x}}\right|_{\mathbf{x}_{k-1}} \tag{5.3}$$

5.2.2 Observation Model

Through sensors on the REMUS-100, measurements for the vehicle's orientation, linear speeds, depth, altitude (seabed's depth), and salient feature positions are obtained (see Figure 5.4). Each sensor has its own measurement model. An observation model gives the predicted sensor measurement from the last known position. The observation model is represented by the non-linear function $\widehat{z}_k = h(\mathbf{x}_k)$ and its linearized version $\mathbf{H}_k$. Equation (5.4) shows a full observation model, i.e. all the sensors giving measurements at the same time.

$$\mathbf{H}_k = \left.\frac{\partial h}{\partial \mathbf{x}}\right|_{\mathbf{x}_k} = \begin{bmatrix} \mathbf{H}_{k,o} \\ \mathbf{H}_{k,v} \\ \mathbf{H}_{k,d} \\ \mathbf{H}_{k,d_{s-s}} \\ \mathbf{H}_{k,s} \end{bmatrix} \tag{5.4}$$

IMU gives the vehicle's rotation measurements in its three axis. The vehicle's heading, i.e., yaw angle ψ, is given with respect to magnetic north. Therefore, a heading offset is stored and subtracted from each heading measurement in order to avoid its bias. The observation model $\mathbf{H}_{k,o}$, Equation (5.5), is the one used to relate orientation observations with the state vector. Notice that the first three columns correspond to (`x y z`) of the vehicle, the second three columns are orientations ($\phi\ \theta\ \psi$) of the vehicle, then the next three columns are velocities ($v_x\ v_y\ v_z$) and the tenth column corresponds to the distance from the vehicle to the seabed d_{s-s}. After that, the observation matrix is dimension so that it contains as many zeros as landmark variables, i.e. three zero columns for each landmark n in the state vector.

$$\mathbf{H}_{k,o} = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times3n} \end{bmatrix} \tag{5.5}$$

IMU measurement noise $\mathbf{R}_o$ is represented by IMU's uncertainty, as in Equation (5.6).

$$\mathbf{R}_o = \begin{bmatrix} \sigma^2_{IMU_\phi} & 0 & 0 \\ 0 & \sigma^2_{IMU_\theta} & 0 \\ 0 & 0 & \sigma^2_{IMU_\psi} \end{bmatrix} \tag{5.6}$$

DVL produces velocity measurements in the three axis of the vehicle. The observation model $H_{k,v}$, Equation (5.7), is necessary to relate velocity measurments with the state vector. Again, the matrix is adapted to the dimension of the state vector.

$$\mathbf{H}_{k,v} = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3\times3} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times3n} \end{bmatrix} \tag{5.7}$$

DVL measurement noise $\mathbf{R}_v$ corresponds to DVL's uncertainty on each axis, as shown in Equation (5.8).

$$\mathbf{R}_v = \begin{bmatrix} \sigma^2_{DVL_x} & 0 & 0 \\ 0 & \sigma^2_{DVL_y} & 0 \\ 0 & 0 & \sigma^2_{DVL_z} \end{bmatrix} \tag{5.8}$$

Depth measurments are obtained through pressure sensor. This depth corresponds to the vehicle's z value and its corresponding observation model is given in Equation (5.9). Its associated variance is used as measurement noise, Equation (5.10).

$$\mathbf{H}_{k,d} = \begin{bmatrix} 0 & 0 & 1 & \mathbf{0}_1 & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 1} & \mathbf{0}_{1\times 3n} \end{bmatrix} \tag{5.9}$$

$$\mathbf{R}_d = \sigma^2_{pressure} \tag{5.10}$$

The side-scan sonar sensor provides information from which 2D points corresponding to a landmark's centroid are extracted. See Figure 5.4 for a better comprehension of these 2D points. A 2D point lays on the YZ plane from the vehicle's axis and is defined by the horizontal distance between the vehicle's reference frame and the object's centroid in the y component and by the vertical distance between the vehicle and the seabed. As the vehicle's reference frame is rotated and translated with respect to the world frame, the transformation ${}^{\mathbf{W}}\mathcal{T}_{\mathbf{V}}$ is used. Therefore, the observation model $\mathbf{H}_{k,s}$ is the linearization of this transformation, as shown in Equation (5.11). Notice that in this case, all columns corresponding to landmarks are filled with zeros, except for the three columns corresponding to the landmark in the state vector that is now being reobserved.

$$\mathbf{H}_{k,s} = \begin{bmatrix} \frac{\partial {}^{\mathbf{W}}\mathcal{T}_{\mathbf{V}}}{\partial \mathbf{x}_V} & \mathbf{0}_{3\times 3} & \mathbf{0} & \dots & \frac{\partial {}^{\mathbf{W}}\mathcal{T}_{\mathbf{V}}}{\partial \mathbf{x}_l} & \dots & \mathbf{0} \end{bmatrix} \tag{5.11}$$

The uncertainty of a side-scan sonar observation is used to define the landmark's measurement noise $\mathbf{R}_s$, as in Equation (5.12).

$$\mathbf{R}_v = \begin{bmatrix} \sigma^2_{sonar_x} & 0 & 0 \\ 0 & \sigma^2_{sonar_y} & 0 \\ 0 & 0 & \sigma^2_{sonar_z} \end{bmatrix} \tag{5.12}$$

Finally, the side-scan produces an estimate of the vehicle's altitude, i.e. the distance from the vehicle to the seabed. This distance is stored in the state vector in its tenth row, therefore, its observation model $\mathbf{H}_{k,d_{s-s}}$ corresponds to the one in Equation (5.13). Its measurement noise $\mathbf{R}_{d_{s-s}}$ is represented by the side-scan sonar uncertainty in its range,

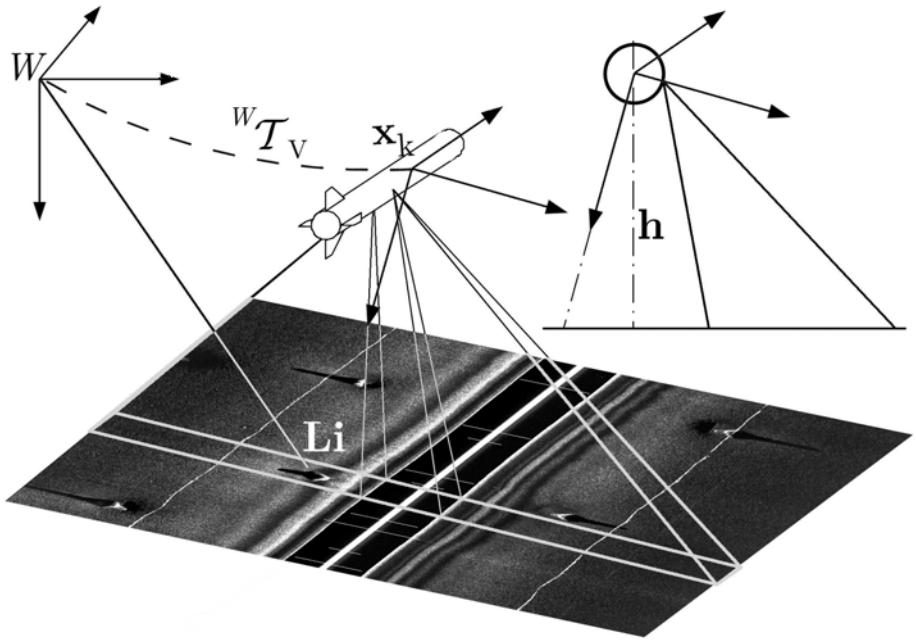


Figure 5.4: A schematic representation of the side-scan sonar measurement procedure.

Equation (5.14).

$$\mathbf{H}_{k,d_{s-s}} = \begin{bmatrix} 0 & 0 & -1 & \mathbf{0}_{1\times 3} & \mathbf{0}_{1\times 3} & 1 & \mathbf{0}_{1\times 3n} \end{bmatrix} \tag{5.13}$$

$$\mathbf{R}_{d_{s-s}} = \sigma^2_{sonar_z} \tag{5.14}$$

5.2.3 Experimental Validation

The vehicle was sent underwater to perform a recognition mission. During the mission, the vehicle navigated a large surface, about 300m × 400m. The whole navigation consisted of a large number of loops, i.e. revisiting the same area several times. The vehicle's depth was almost constantly around 12 meters first and around 14 meters later on, while the sea floor with respect to the water surface was slightly oscillating at a depth around 16 meters. This scenario's sea floor was considerably flat but with several salient objects. The total navigation time was almost 4 hours. The experiment was conducted to gather data but not to run the experiment on-line [1] during the mission. The data was post processed and run through the SSJS algorithm, producing the results shown in Figure 5.5. This figure presents

[1] Experiments were conducted off-line as the system was all implemented in Matlab. However, for real purposes the most critical issue should be the computational effort, which could be addressed by optimizing the code for any specific on-board computer.

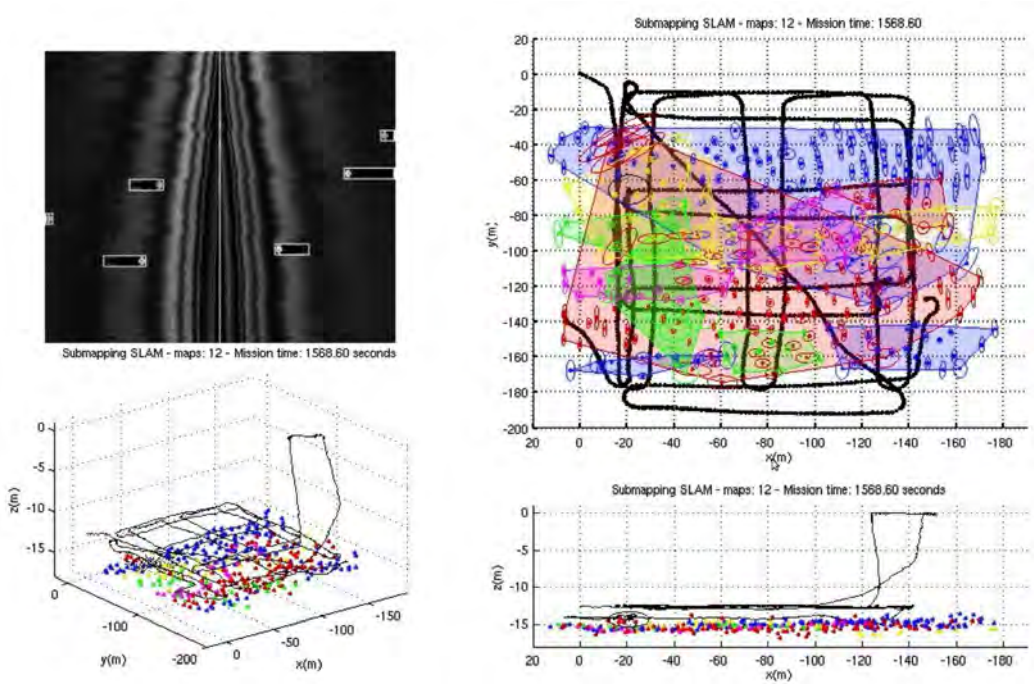


Figure 5.5: Plot of the execution at a certain step during the mission. Top-left: side-scan image with several detected objects. Bottom-left: 3D plot of the map and the trajectory of the vehicle. Top-right: `XY` top view of the map and trajectory. This plot shows the submaps in different colours, together with their corresponding landmarks and uncertainty ellipses. Bottom-right: `XZ` frontal view of the execution. In this plot, it is easy to see that the vehicle had been navigating at around 12 meters, then went to the surface and dived back to around 14 meters.

the object detection performance (top-left), while the other three plots show different views of vehicle's trajectory and submaps that were generated until that specific time step.

Map consistency is shown qualitatively in Figure 5.6. This figure compares the trajectory given by dead reckoning to the one obtained by SSJS and LBL. LBL provides an accurate positioning of the vehicle, therefore it can be considered as a reference. Notice that the SSJS SLAM approach clearly improved dead reckoning results.

Another example that shows the consistency of the approach is shown in Figure 5.7. Figure 5.7(a) shows that the discrepancy between LBL and SSJS is always kept inside the uncertainty boundaries, which means that the filter will not cause divergences due to overconfidence. Figure 5.7(b) shows an example of an overconfident estimation of this discrepancy, which will lead to inconsistencies [Aulinas 2010b]. This overconfidence appeared in simulations with larger submaps, and was an expected result, as with large submaps the approach tends to become a standard EKF.

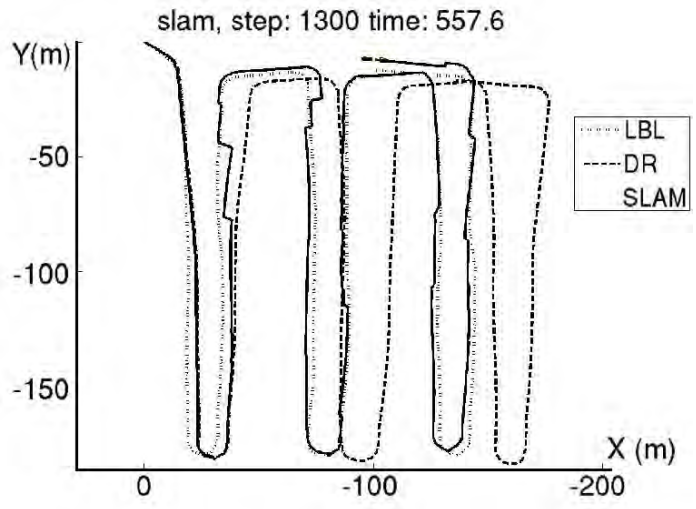


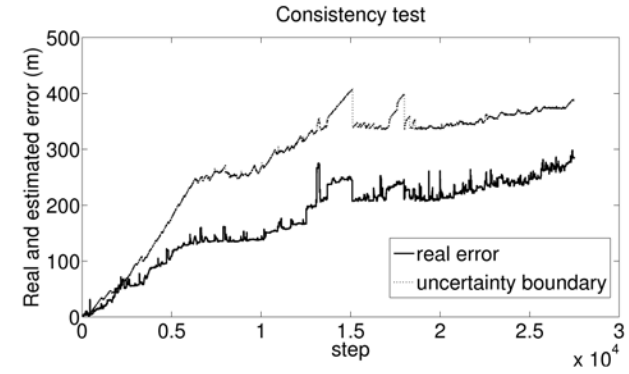
Figure 5.6: Short paths of the vehicle's trajectory are compared. These trajectories are obtained by means of an LBL, dead reckoning and our SSJS SLAM. The figure shows a clear improvement when using our SSJS SLAM compared to dead reckoning.

5.3 SSJS on the SPARUS AUV

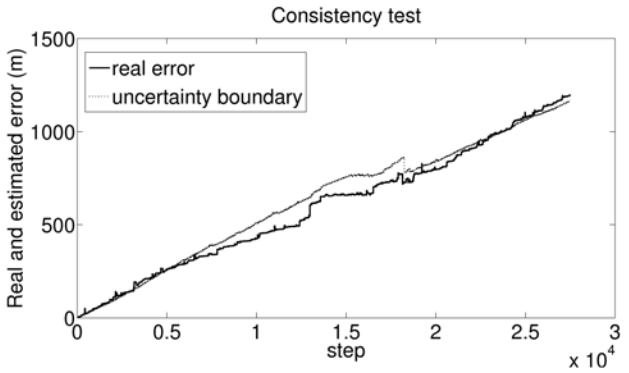
The second set of experiments presented in this chapter were conducted in a real unstructured underwater environment using the SPARUS AUV (see Figure 5.8). SPARUS was developed with the purpose of participating in the Student Autonomous Underwater Challenges – Europe (SAUC-E) competition. SPARUS won the 2010 SAUC-E edition. Since then, the vehicle has been on several mission. On of them was a survey mission, which was then used to validate the SLAM algorithm presented in this work.

SPARUS is equiped with several sensing devices: a DVL, an IMU, a down-looking camera, a forward-looking camera, an imaging sonar and a GPS (see Figure 5.9). In this work, only the DVL, the IMU and the down-looking camera are used, producing information about velocities, orientations and the sea floor. The vehicle's state vector for SPARUS is defined in the same way as presented for the REMUS-100, a 9-vector composed of the 6-DOF vehicle's pose $\mathbf{x}_V = (x\ y\ z\ \phi\ \theta\ \psi)^{\mathrm{T}}$ and the vehicle's frame linear velocities $(v_x\ v_y\ v_z)^{\mathrm{T}}$. However, in this case, the vehicle is very stable in roll and pitch thanks to the three thruster design, which means that the vehicle could be defined by using only 4-DOF, meaning that the vehicle's pose could be $\mathbf{x}_V = (x\ y\ z\ \psi)^{\mathrm{T}}$. Despite the possibility of simplifying the state vector, provided that the IMU gives measurements for the three orientations, the vehicle's state vector was finally chosen to be like the one for REMUS-100, as described in Section 5.2.

The vehicle's reference frame is set according to Figure 5.10 representation. Figure 5.9 shows the position of each sensor within the vehicle, giving an idea of rotations and translations necessary to bring all measurements to a common reference frame.



(a)



(b)

Figure 5.7: a) An example of configurations for the SSJS approach, where the discrepancy between the LBL and SSJS (solid line) and the uncertainty boundary (dotted line) are drawn. a) consistent example; b) inconsistent example.

Figure 5.8: SPARUS vehicle.

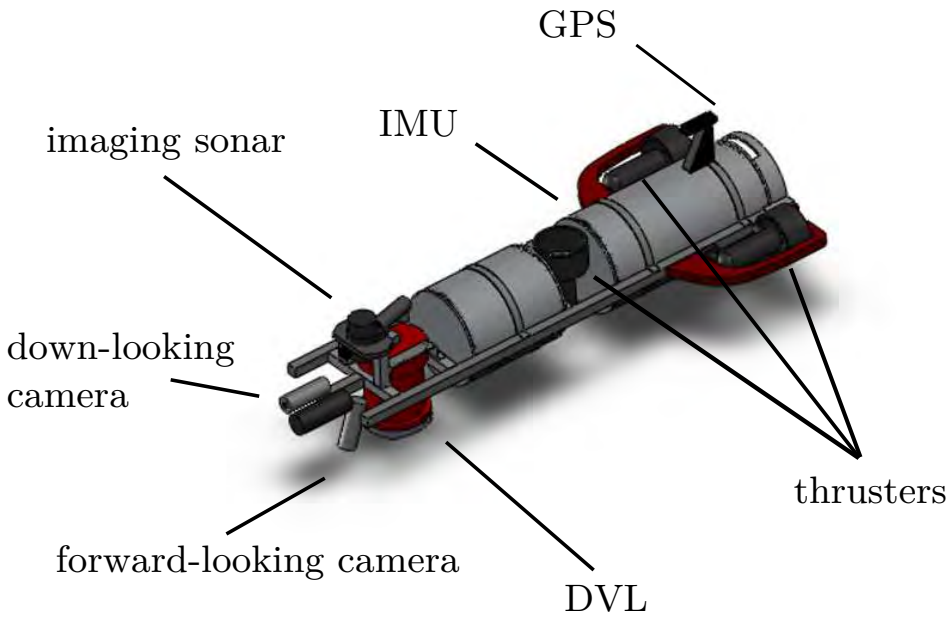


Figure 5.9: SPARUS 3D model with its sensors.

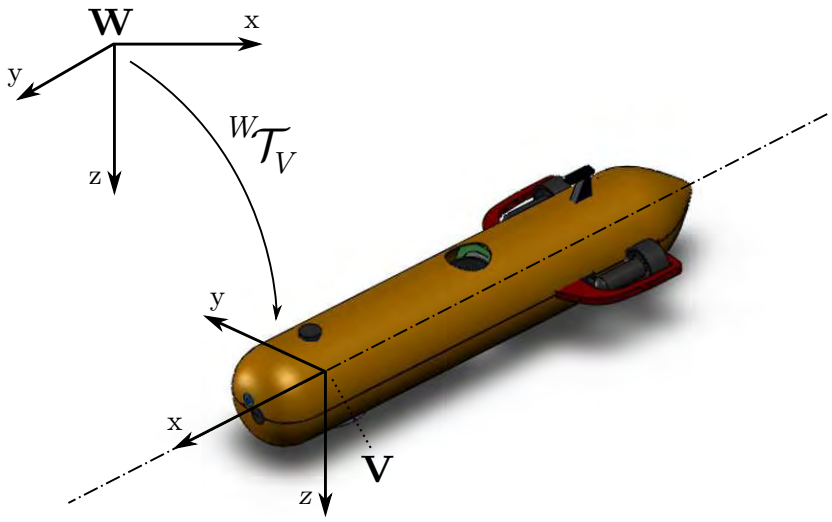


Figure 5.10: SPARUS 3D model with its reference frame, together with world frame (**W**).

5.3.1 Motion Model

A 6-DOF motion model for SPARUS corresponds to the one presented in Equation (5.2). For a simpler modelling of the vehicle's kinematics, hereafter a 4-DOF model is described in Equation (5.15). As this is a non-linear model, it is necessary to compute its Jacobian in order to linearize it, as in Equation (5.3).

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_k) = \begin{bmatrix} x_k \\ y_k \\ z_k \\ \psi_k \\ v_{x,k} \\ v_{y,k} \\ v_{z,k} \end{bmatrix} = \begin{bmatrix} x_{k-1} + v_{x,k-1} dt \cos(\psi_{k-1}) - v_{y,k-1} dt \sin(\psi_{k-1}) \\ y_{k-1} + v_{x,k-1} dt \sin(\psi_{k-1}) + v_{y,k-1} dt \cos(\psi_{k-1}) \\ z_{k-1} + v_{z,k-1} dt \psi_{k-1} \\ v_{x,k-1} \\ v_{y,k-1} \\ v_{z,k-1} \end{bmatrix} \tag{5.15}$$

5.3.2 Observation Model

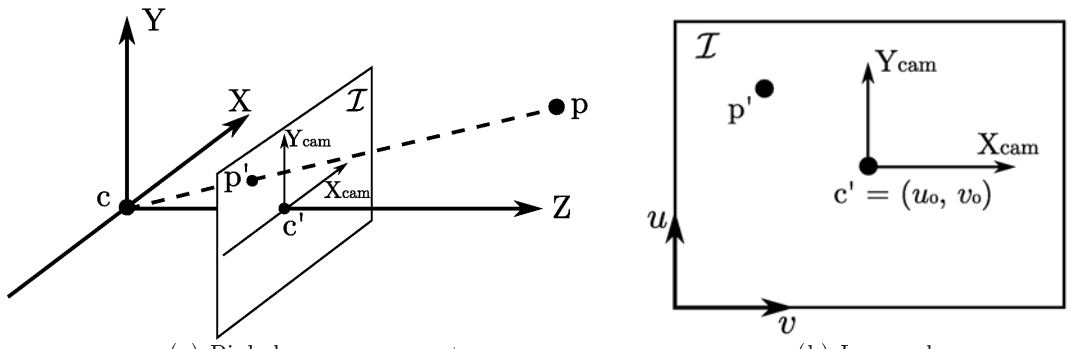
The map is represented by point features extracted from optical images. The pinhole model [Hartley 2000] is a common model used to project points in the scene onto the image plane. A Euclidean representation is commonly used to represent a 3D point $\mathbf{p}$ in the scene, i.e. $\mathbf{p} = (X,\ Y,\ Z)^{\mathrm{T}}$. The basic principle of this model is shown in Figure 5.11. In this figure, an image plane $\mathcal{I}$ is located at a distance f from the optical center $\mathbf{c} = (x_c,\ y_c,\ z_c)$ of the camera on its z axis. This distance f is known as focal length. The projection of $\mathbf{p}$ on the image plane $\mathbf{p'}$ is the intersection of plane $\mathcal{I}$ and the ray that connects the camera center $\mathbf{c}$ with point $\mathbf{p}$. From the point of view of the image plane (see Figure 5.11(b)), image axis are $(u,\ v)$ and the camera's axis are the ones corresponding to the principal point $\mathbf{c'}$, also referred as $(u_0,\ v_0)$. Mathematically speaking, the projection of $\mathbf{p}$ is obtained by using the projection matrix $\mathbf{K}$, as in Equation 5.16.

$$\mathbf{p'} = \mathbf{K}\mathbf{p} \tag{5.16}$$

$\mathbf{K}$ is the so-called calibration matrix, Equation (5.17), which contains information about focal length f, principal point coordinates $(u_0,\ v_0)$, distortion parameters k_u, k_v, and pixel skew s. The skew parameter generally equals zero.

$$\mathbf{K} = \begin{bmatrix} fk_u & s & u_0 \\ 0 & fk_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{5.17}$$

Therefore, mapping a 3D point onto the image plane in homogeneous coordinates is



(a) Pinhole camera geometry. (b) Image plane.

Figure 5.11: Pinhole camera model. **c** is the optical center and **c'** is its corresponding projection onto the image plane, known as principal point (u_0, v_0). **p** is a 3D point in the scene, whose projection onto the image plane is **p'** [Hartley 2000].

Equation (5.18):

$$\begin{bmatrix} x_{\mathbf{p}'} \\ y_{\mathbf{p}'} \\ z_{\mathbf{p}'} \end{bmatrix} = \begin{bmatrix} fk_u & 0 & u_0 & 0 \\ 0 & fk_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} = \begin{pmatrix} fk_u X + u_0 Z \\ fk_v Y + v_0 Z \\ Z \end{pmatrix} \tag{5.18}$$

Equation (5.18) presents the mathematical model used to project a 3D point onto the image plane, but in reality one might need to compute the inverse of this operation. A 2D point **p'** is known on the image plane, in pixel dimensions. Therefore, a parametrization able to encode **p'** into a 3D point **p** in metrics is necessary. In this case, instead of using a common Euclidean representation of a 3D point, i.e. $\mathbf{x}_l = (x_l, y_l, z_l)$, the inverse depth parametrization presented by [Civera 2008] is used. Hereafter, the inverse depth parametrization principle is introduced, as was done by the authors.

The inverse depth parametrization is defined by a 6-vector composed of $\mathbf{x}_l = (x_{c_1}, y_{c_1}, z_{c_1}, \alpha_{c_1}, \beta_{c_1}, \rho_{c_1})^{\mathrm{T}}$. This parametrization encodes the ray from the camera position from which the feature was first observed, this is, the optical center $(x_{c_1}, y_{c_1}, z_{c_1})$, and the azimuth α_{c_1} and elevation β_{c_1}, as shown in Figure 5.12. In this figure, point $\mathbf{p}_i$ in the scene can be computed from point $\mathbf{p}'_i$ on the image plane by using Equation (5.19).

$$\mathbf{p}_i = \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} = \begin{pmatrix} x_{c_1} \\ y_{c_1} \\ z_{c_1} \end{pmatrix} + \frac{1}{\rho_{c_1}} \mathbf{m}(\alpha_{c_1}, \beta_{c_1}) \tag{5.19}$$

where $\mathbf{m}(\alpha_{c_1}, \beta_{c_1})$ is a unit vector pointing azimuth α and elevation β with respect to the world, Equation (5.20).

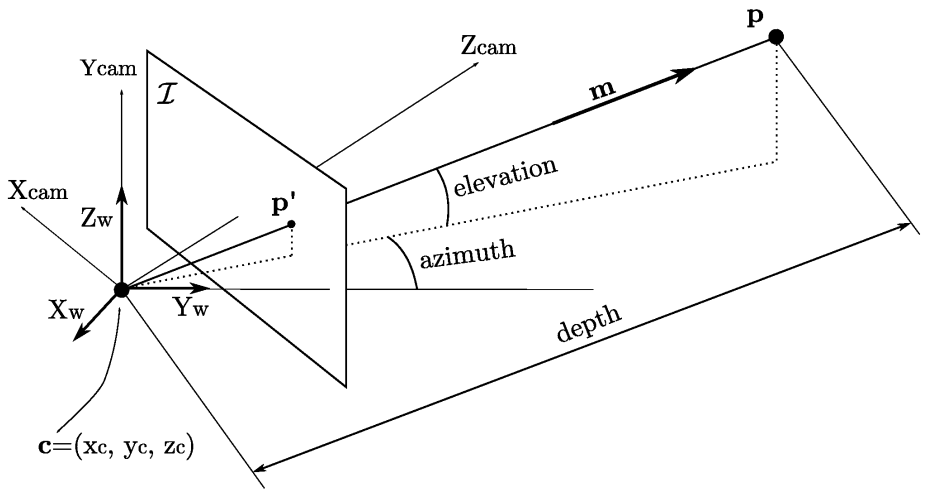


Figure 5.12: A schematic representation of the inverse depth parametrization. $\mathbf{m}$ is the unit vector that points from the optical center $\mathbf{c}$ to the 3d point $\mathbf{p}$, with a certain azimuth and elevation with respect to the world frame.

$$\mathbf{m}(\alpha_{c_1}, \beta c_1) = (\cos\alpha_{c_1}\sin\beta_{c_1} - \sin\alpha_{c_1},\ -\sin\alpha_{c_1},\ \cos\alpha_{c_1}\cos\beta c_1) \tag{5.20}$$

Being a monocular system, no depth information can be retrieved, therefore, one should expect that a 3D point $\mathbf{p}$ lays on the ray but without any knowledge of its distance d_{c_1} to the optical center $\mathbf{c}$. Using inverse depth parametrization, the inverse of this distance is first initialized with ρ_{c_1}, together with its uncertainty, and then, as the same point is reobserved, it is corrected.

Using this point parametrization, the observation model ${}^{C}\mathbf{h}$ can be defined as in Equation 5.21, where ${}^{C}\mathbf{R}_W$ and ${}^{C}\mathbf{r}_W$ are the rotation and translation to map components from world frame to camera frame.

$${}^{C}\mathbf{h} = \begin{pmatrix} X_i \\ Y_i \\ Z_i \end{pmatrix} = {}^{C}\mathbf{R}_W \left(\rho_{c_1} \left(\begin{pmatrix} x_{c_1} \\ y_{c_1} \\ z_{c_1} \end{pmatrix} - {}^{C}\mathbf{r}_W \right) + \mathbf{m}(\alpha_{c_1}, \beta_{c_1}) \right) \tag{5.21}$$

${}^{C}\mathbf{h}$ codes a point from the world to the camera in metric dimensions. In order to have this point projected on the image plane in pixel dimensions, it is necessary to use a normalized pinhole model as in Equation (5.22). With this equation, an estimate of a point that was in the map can be projected onto the current image plane, therefore it can be associated to current observations.

$$\mathbf{h} = \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} fk_u \frac{X_i}{Z_i} + u_0 \\ fk_v \frac{Y_i}{Z_i} + v_0 \end{pmatrix} \tag{5.22}$$

In the event of a new observation that is not associated with any existing landmark, it is necessary to initialize this observation in the state vector, meaning, a 2D point on the image plane $\mathbf{p'}_n = (u_n,\ v_n)$ must be transferred to world coordinates by means of inverse depth parametrization, i.e. $\mathbf{p}_n = ({}^W c_x\ {}^W c_y\ {}^W c_z\ {}^W\alpha_p\ {}^W\beta_p\ {}^W\rho_p)^{\mathrm{T}}$. This parametrization contains the current camera position with respect to the world frame $({}^W c_x\ {}^W c_y\ {}^W c_z)$. Azimuth and elevation are obtained following the idea shown in Figure 5.12. First, the camera frame is rotated using ${}^W\mathbf{R}_C$ to match the world frame, Equation (5.23).

$$\begin{pmatrix} {}^W p'_x \\ {}^W p'_y \\ {}^W p'_z \end{pmatrix} = {}^W\,\mathbf{R}_C \begin{pmatrix} u_n \\ v_n \\ 1 \end{pmatrix} \tag{5.23}$$

Afterwards, ${}^W\alpha_p$ and ${}^W\beta_p$ are computed as in Equation (5.24).

$$\begin{pmatrix} {}^W\alpha_p \\ \\ {}^W\beta_p \end{pmatrix} = \begin{pmatrix} \arctan\left({}^W p'_x,\ {}^W p'_z\right) \\ \\ \arctan\left(-{}^W p'_y,\ \sqrt{{}^W p'^2_x + {}^W p'^2_z}\right) \end{pmatrix} \tag{5.24}$$

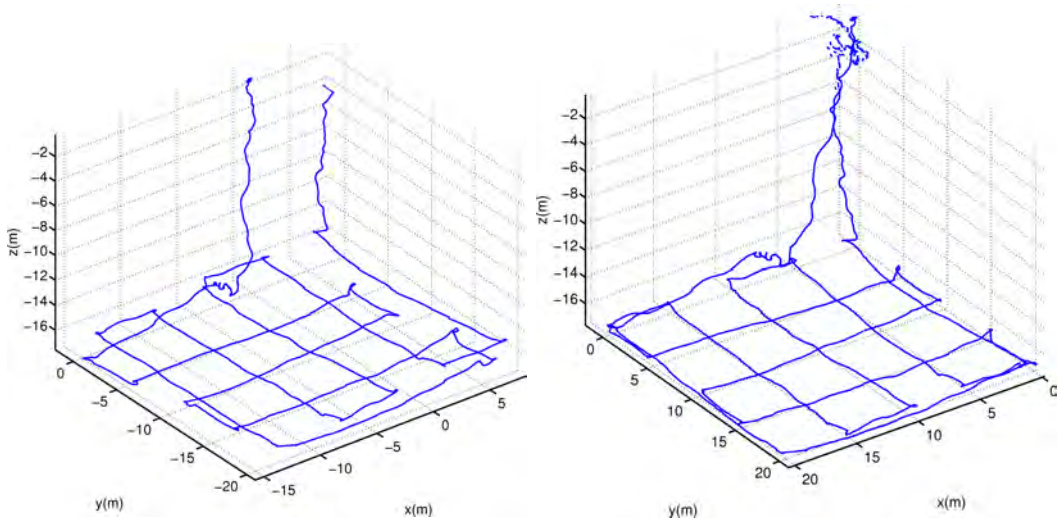
According to [Civera 2008], an initial value for $\rho_p = 0.1$ is appropriate. The linearization of this motion model is obtained by taking its Jacobian.

5.3.3 Experimental Validation

Experimental validation is done through the data acquired by SPARUS during a survey mission. The mission consisted of navigating an area of about $20m \times 20m$, in a grid of $5m \times 5m$. The vehicle's depth was almost constantly around 17 meters. The total navigation time was about 17 minutes. The vehicle carried a down-looking camera that acquired a total of 3199 images. Experimental results obtained with the SSJS approach show that there is a significant improvement for trajectory estimate using SSJS as compared to dead reckoning (see Figure 5.13).

Automatic detection of features provides robust features. Features are considered to be robust if they can be identified even when observed from different points of view and orientations. However, these features are very few and widely spaced, which is positive for the data association, but not interesting if one wants to test the performance of SSJS. If the scenario contains only a few landmarks, it is not worth using submaps. Figure 5.14 shows the trajectory and map information obtained using automatic feature extraction, as presented in Section 4.4. In this figure, every 3D point in the map has been associated with its texture patch in order to give an idea of extracted RoI.

In order to test the performance of SLAM using submaps, a subset of random 2D points where extracted from a mosaic of the scene (see Figure 5.15). These 2D points were then



(a) Dead reckoning (b) SSJS

Figure 5.13: A 3D view of the vehicle's trajectory. In a) one can observe the drift suffered during the mission, as the ending point is far from the starting point, while in b) this drift has been corrected by the means of SSJS.

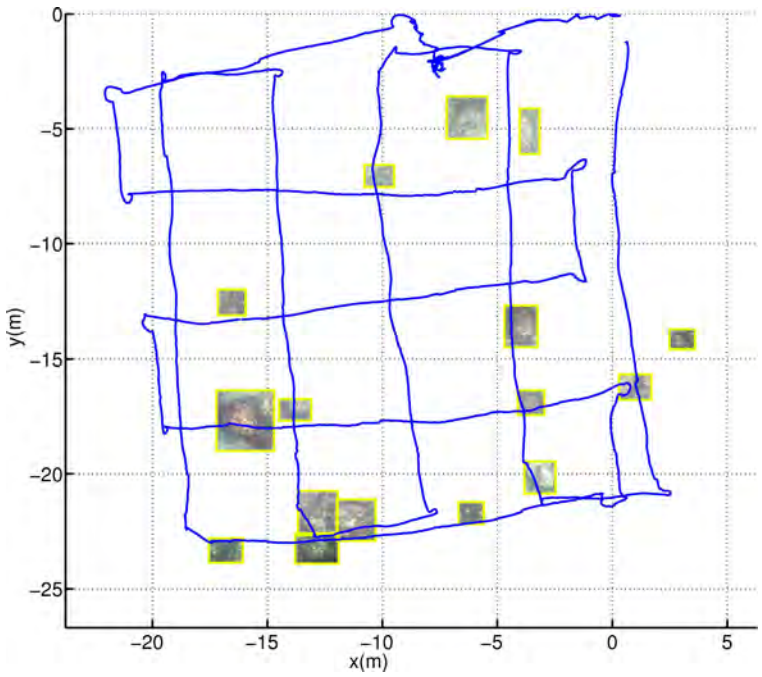


Figure 5.14: Top view of trajectory and texture patches of regions of interest.

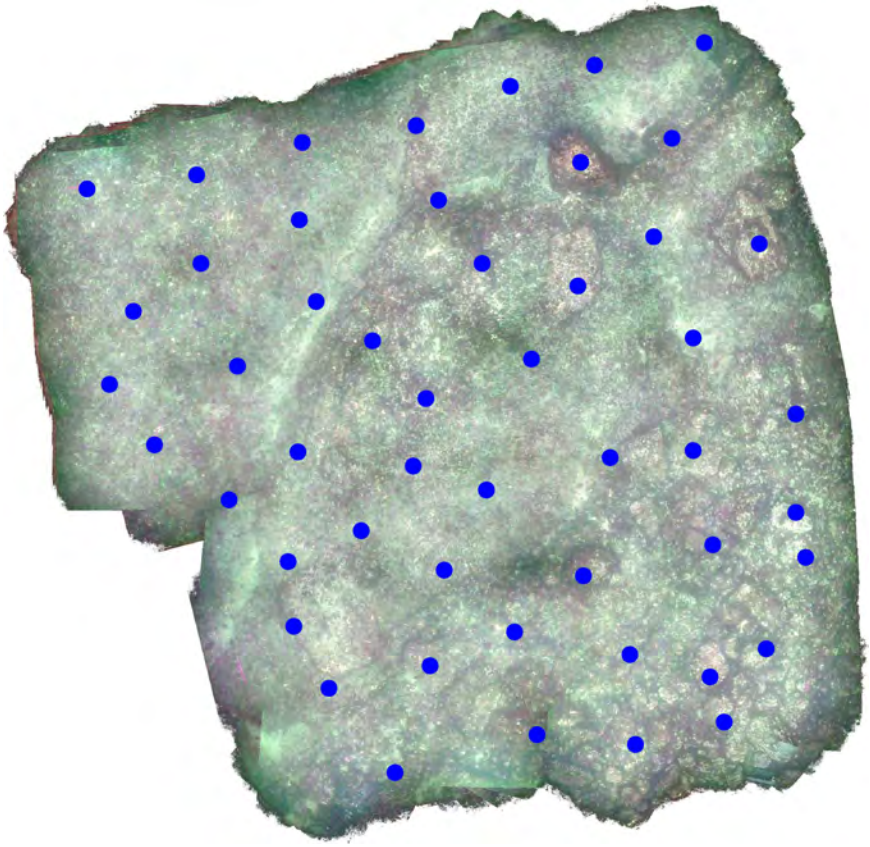


Figure 5.15: A 2D Mosaic of the scenario with 2D random points.

back referred to the image to which they belonged. This subset of points was used instead of automatically detecting RoI and features.

The performance of SSJS using this set of points is shown in Figure 5.16. Figure 5.16 shows a sequence of 5 frames containing first one landmark and later on two landmarks. In addition, the uncertainty projected on the image plane is drawn.

Figure 5.17 presents a top and a frontal view of the resulting map and trajectory. In these views, one can see the vehicle's trajectory corrected with SSJS and the landmark's location, as well as its associated uncertainties. Finally, figure 5.18 shows a 3D plot of these results.

5.4 Discussion

The main contribution of this chapter is a 6-DOF implementation of SSJS for underwater vehicles. In addition, motion and observation models are detailed in this chapter. First, the 6-DOF motion model for REMUS-100 AUV is provided, together with an observation model for side-scan sonar sensors. Experiments done with real data show a bounded effect of the

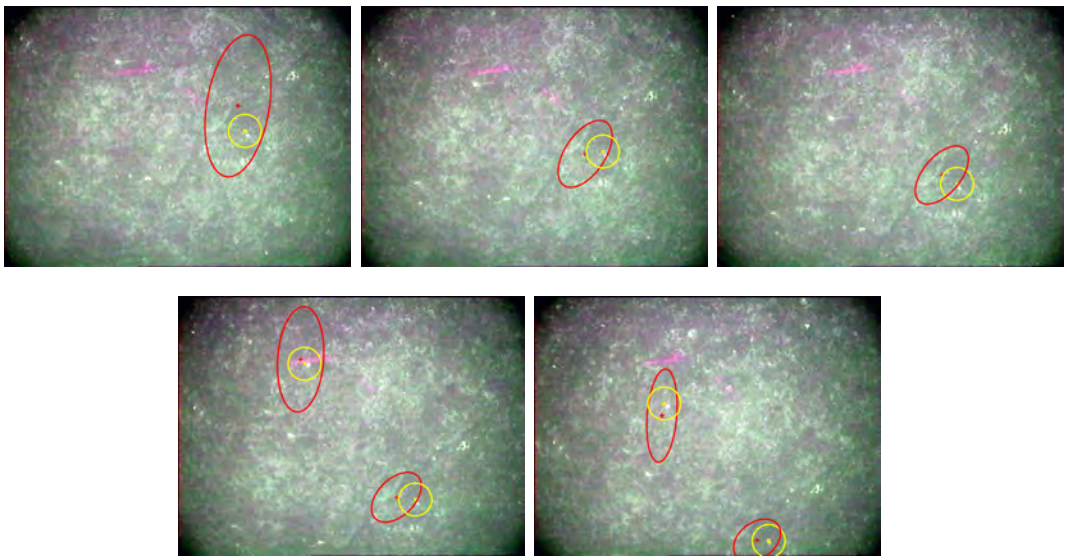


Figure 5.16: A sequence of down-looking camera frames. in yellow (x) a new observation and its associated measurement uncertainty. In red (+) the prediction of a landmark that was already in the map and the projection of its associated uncertainty onto the image plane. One can observe a reduction of uncertainty with the new observations.

linearisation error and also a precise reconstruction of the map since the drift suffered in shorter distances is smaller and the data association can be more robustly solved. The second part of this chapter, describes an alternative 4-DOF motion model that could be used on SPARUS AUV. However, after several tests and provided that measurements for 6-DOF are available, the final motion model for SPARUS was chosen to be like the one for REMUS-100. Also, the observation model for a down-looking optical camera was introduced. This model is based on inverse depth parametrization. Experiments conducted in a real unstructured environment demonstrated that SSJS improves the vehicle's trajectory as compared to dead reckoning. Moreover, SSJS combined with inverse depth parametrization was capable of producing a three-dimensional map.

The main benefit of the system is the capacity to produce consistent maps of larger size than the maps that could produce a standard EKF base method. This benefits arise from the fact that subdividing the whole problem into small regions reduce the computational cost and the uncertainty at the same time. Reducing the computational cost means being closer to real time solutions. Reducing the uncertainty means being able to navigate larger distances with a higher level of confidence. Therefore, the use of such systems on real vehicles shows a promising future.

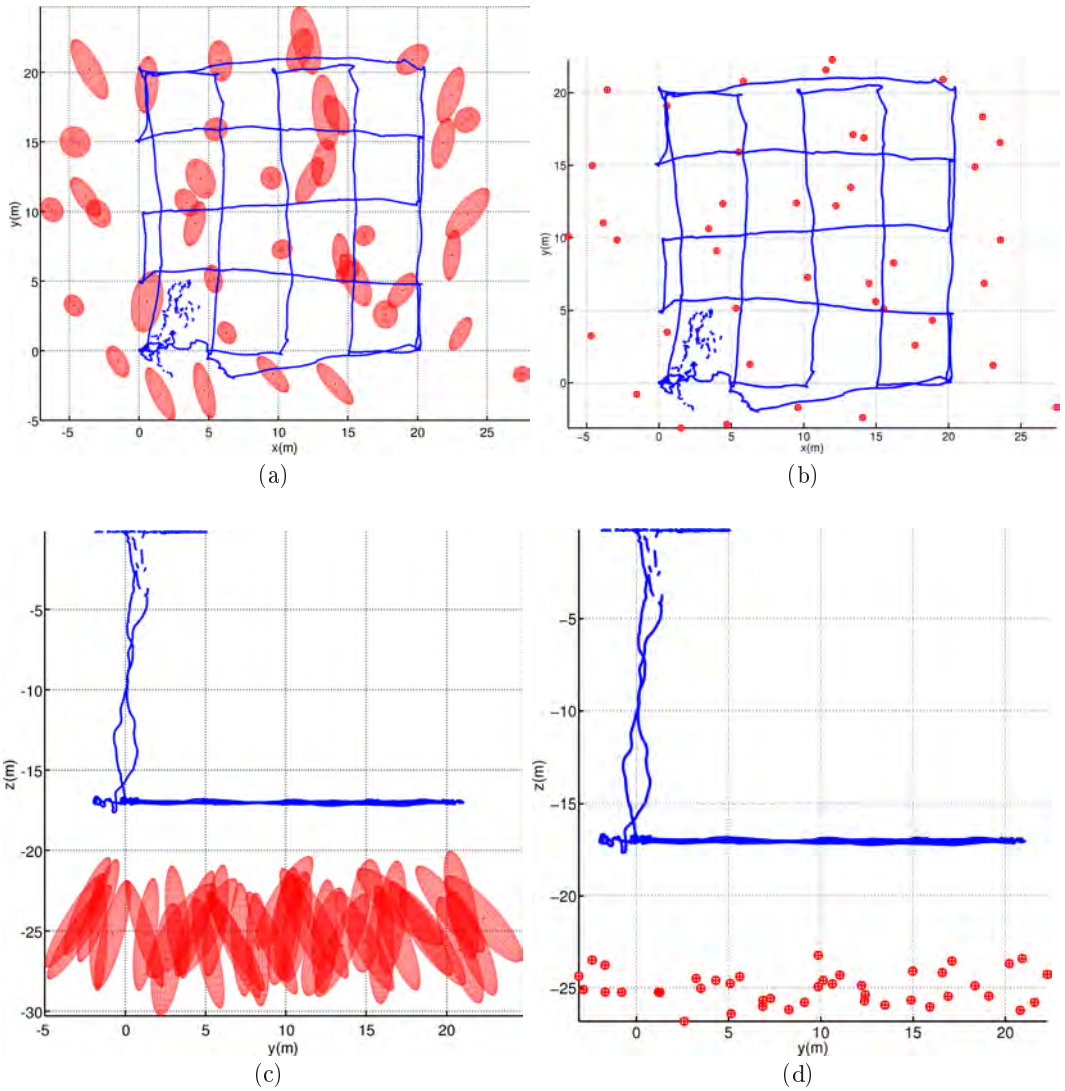


Figure 5.17: Different views of the results produced by SLAM. a) and b) show a top view, while c) and d) present a frontal view. On the left, landmark uncertainties are drawn, while on the right, only the landmark is shown.

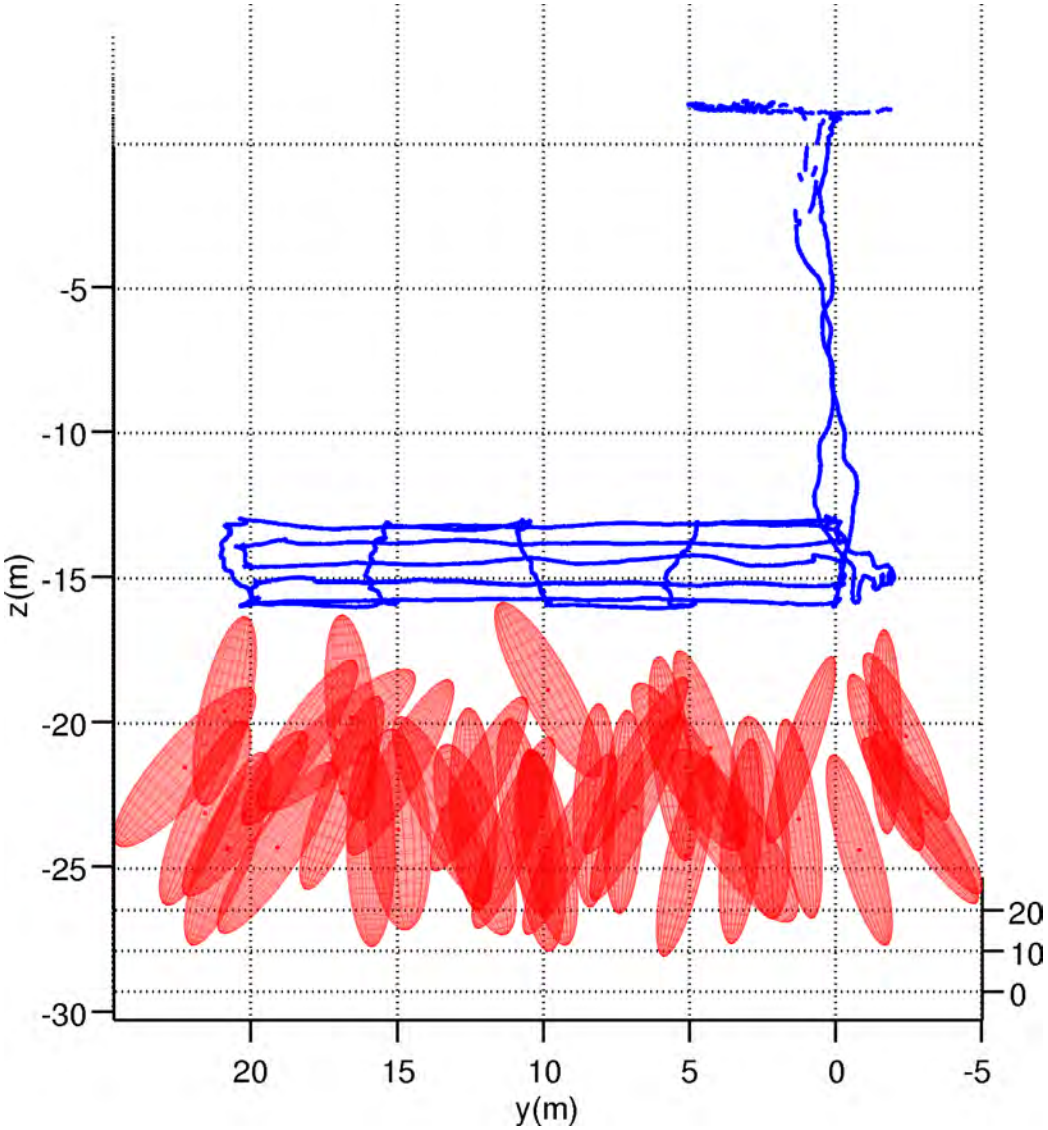


Figure 5.18: A 3D plot of the SLAM solution.

Chapter 6

Conclusions and further work

Contents

This chapter presents the conclusions and future possibilities opened by this work. First, the contents presented in each chapter are summarized. The scientific contributions of this PhD thesis are discussed. Further work and future perspectives are discussed. Finally, the list of publications related to this work is presented as well as scientific collaborations conducted during this thesis preparation.

6.1 Conclusions

This thesis focused on developing new techniques to tackle the issues related to the problem of SLAM. Over the last two decades this topic has been widely investigated but still remains as an unsolved problem. Several approaches exist, some of which produce consistent maps and precise localization but at a very high computational cost, i.e., optimization techniques, such as bundle adjustment. Others produce suboptimal solutions which converge on short missions but show several issues when dealing with large missions. These techniques were reviewed in Chapter 2. Most SLAM approaches work under gaussianity assumptions, i.e. noise is described using Gaussian distribution. This noise produce uncertainty, thus filtering techniques are necessary. In the first part of Chapter 2, a survey of common filtering techniques was presented, giving pros and cons, justifying the decision of using an EKF based SLAM. The EKF was chosen because it converges and can handle uncertainties. However, on large mission, the computational cost increases with the size of the map and linearization errors accumulate, which means that the solution could become inconsistent. For this reason, several researchers address this issue by using submaps.

Chapter 2 also surveyed existing submapping approaches and presented a comparison table, reaching the conclusion that the use of submaps is appropriate to reduce computational

costs and increase consistency. The survey focused on underwater SLAM solutions. The main conclusion for this section was that most of the approaches use an EKF based SLAM with point features from acoustic images. However, optical cameras are gaining interest due to the fact that they provide texture information, which might be necessary during intervention missions. Regardless of the sensor, SLAM algorithms require features in order to build the map. Therefore, feature extraction and matching algorithms are necessary, and a survey of underwater imaging approaches was presented in order to find which would be the proper feature extraction technique and which would be the proper feature matching approach. This section reviewed underwater imaging for two different sensors: a side-scan sonar and an optical camera. The reason the review was limited to these two sensing devices was the fact that these were the sensors on-board available vehicles. The main outcome of this chapter was the decision to focus efforts on the research of an EKF submapping based SLAM for autonomous robots.

Research conducted in this work produced the approach presented in Chapter 3. The approach proposed was called SSJS. SSJS uses an EKF to build submaps and a novel strategy to fuse them, which consists of deciding whether to fuse two independent maps, depending on the amount and the quality of the information they share. This chapter presented first the theoretical background behind an EKF. Afterwards, SSJS was described step by step. Finally, synthetic and real experiments were presented. Real experiments were conducted using the Victoria Park dataset. In addition, a comparison between state of the art submapping approaches and SSJS was presented in order to asses the performance of SSJS. These results demonstrated that SSJS produces convergent and consistent results, allowing us to go on and test the method on other vehicles. In this case, available vehicles were two AUVs: REMUS-100 and SPARUS. The former was carrying a side-scan sonar, while the later was carrying a down-looking optical camera. Both sensors acquired information from the sea floor, which needed to be processed in order to extract features for the SLAM.

Chapter 4 presented imaging algorithms used to extract and match features for three different cases: the Victoria Park dataset (taken from ACFR), the REMUS-100 dataset (gathered at OSL) and the SPARUS dataset (acquired at VICOROB). The first dataset contained laser range finder readings from a park populated with several trees. For the Victoria Park dataset, an approach capable of extracting point features from the laser range finder was presented. Each point feature corresponded to the centre of a tree. In addition, data association algorithms were summarized and compared, reaching the conclusion that the JCBB and the RJC are the most appropriate for this kind of scenario. For the second dataset, two approaches to detecting objects in side-scan sonar images were presented. The first approach used thresholds and common morphological operations to extract bright spots and shadows. The second approach used machine learning techniques, more precisely, the Viola and Jones approach, which was initially designd to detect faces. In this case, the

algorithm was trained to detect objects that appear in side-scan sonar images, i.e. manta type mines. For the SPARUS dataset, computer vision algorithms were presented. These algorithms can tackle the issues caused by underwater artifacts (distortion, back-scattering and non-uniform light conditions). Salient features were extracted by segmenting RoIs and SURF features. The results for each sensor were presented through experimental validation. All of them were capable of extracting features with high rate detection and a low rate of false positives.

Chapter 5 presented SSJS results for the REMUS-100 and the SPARUS using feature extraction and matching approaches presented in Chapter 4. Motion and observation models were described for the REMUS-100 AUV, later on the performance of SSJS was analysed. This performance was shown to be consistent and produced a convergent solution. Moreover, observation and motion models were described, but this time only for SPARUS. In this case, the observation model used was based on inverse depth parametrization. Despite a significant amount of noise in the sequence of images, SSJS performed consistently.

6.2 Contributions

The main contributions of this thesis are:

- A novel SLAM strategy based on dividing the whole scenario into several submaps. Although the method is similar to other state of the art techniques, it introduces the idea of deciding when to fuse two maps and when to leave them independent.

- A robust SLAM framework suitable for any autonomous platform. The core is an EKF based SLAM algorithm which takes advantage of sensor fusion to provide robust localization and map estimates.

- This thesis demonstrates the integration of several sensor systems and SLAM to provide reliable estimates of large scenes.

- The approach is validated with synthetic experiments. Large scale results are consistent and shown to be close to ground truth. Afterwards, the approach is also validated using the well known Victoria Park dataset [Guivant 2000a]. Our results were compared to those produced by state of the art approaches. This comparison showed that SSJS is consistent and its computational cost does not grow exponentially.

- Once the performance of the approach was tested and validated, it was adapted to underwater vehicles equipped with different sensors. These were a side-scan sonar used on a REMUS AUV and optical imagery on the SPARUS AUV. Both implementations produced consistent maps and accurate vehicle localization.

- Underwater imaging algorithms for underwater object detection and matching were presented, performing at a high detection rate and a low false positive rate.

6.3 Future work

Short term perspectives

- The approach presented in this work was compared using a 3-DOF example. Afterwards, the proposed approach was adapted to 6-DOF models in order to be tested on underwater vehicles. It would be interesting to compare different approaches with these 6-DOF models, so, this remains as immediate future work.
- It would be very useful to optimize the code by coding it into C++ or any other language that could suit the on-board computers of any given vehicle.

Long term perspectives

- It is worth mentioning that there is a very recent trend to use multiple target tracking with Probability Hypothesis Density (PHD) filters. This seems to be a good mathematical background that could solve data association issues. In fact, within VICOROB there is a student currently conducting a PhD on SLAM focusing on the use of PHD filters.
- Another interesting point arises from the increase of computational capabilities, namely, optimization techniques, such as bundle adjustment based approaches and SFM approaches. These are now gaining interest within SLAM community.
- In terms of sensing devices, the use of stereo optical cameras would be very interesting in order to produce better three-dimensional landmark estimates and a denser map reconstruction.
- It would be interesting to integrate SSJS into other AUVs, such as ICTINEU and Girona-500 from VICOROB.

6.4 Publications and Scientific Collaborations

The work developed in this thesis has produced three (one under review) journal publications and several contributions to international conferences. Contributions are presented in

the following list. Finally, scientific collaborations made during the thesis preparation are summarised.

Journal papers:

- Y. Petillot, F. Maurelli, N. Valeyrie, A. Mallios, P. Ridao, J. Aulinas, J. Salvi, "Acoustic-based techniques for AUV localization", *Journal of Engineering for Maritime Environment*, 224(4): 293–307, 2010.
- J. Aulinas, J. Salvi, X. Lladó, Y. Petillot, "Local map update for large scale SLAM", *Electronic Letters (ELT)*, 46(8), 564–566, 2010.
- J. Aulinas, J. Salvi, X. Lladó, Y.R. Petillot, J. Batlle, "Selective submap joining in Simultaneous Localization and Mapping", Robotics and Autonomous Systems, in preparation, 2011.

Conference papers:

- J. Aulinas, J. Salvi, X. Lladó, Y. Petillot, J.J. Serrano, "Feature extraction for visual SLAM based 3D reconstruction", *IEEE/OES Oceans Conference,* ***OCEANS'11***, Santander (Spain), June 6-9, 2011.
- J. Aulinas, Y. Petillot, X. Lladó, J. Salvi, "Vision based underwater SLAM for the SPARUS AUV", *10th Conferenece on Computer Applications and Information Technology in the Marine Industries, pp. 171–181,* ***COMPIT***, Berlin (Germany), May 2-4, 2011.
- J. Aulinas, A.Fazlollahi, J. Salvi and X. Lladó, "Robust automatic landmark detection for underwater SLAM using side-scan sonar imaging ", *11th International Conference on Mobile Robots and Competitions,* ***ROBOTICA***, pp. 22–27, Lisboa (Portugal), April 6-10, 2011.
- J. Aulinas, C.S. Lee, J. Salvi, Y. Petillot, "Submapping SLAM based on acoustic data from a 6-DOF AUV", *8th IFAC Conference on Control Applications in Marine Systems,* ***IFAC/CAMS***, Rostock-Warnemünde (Germany), September 15-17, 2010.
- J. Aulinas, x. Lladó, J. Salvi, Y. Petillot, "Selective Submap Joining for underwater Large Scale 6-DOF SLAM", *IEEE/RSJ International conference on Intelligent Robots and Systems,* ***IROS***, pp. 2552-2557, Taipei(Taiwan), October 18-22, 2010.
- J. Aulinas, X. Lladó, J. Salvi, Y. Petillot, "Feature based SLAM using Side-Scan salient objects", *MTS/IEEE Oceans conference,* ***OCEANS'10***, Seattle (USA), September 20-23, 2010.

- J. Aulinas, X. Lladó, J. Salvi, Y. Petillot, "SLAM base Selective Submap Joining for the Victoria Park Dataset", *7th IFAC Symposium on Intelligent Autonomous Vehicles,* ***IFAC/IAV***, Lecce (Italy) September 6-8, 2010.

- J. Aulinas, X. Lladó, Y. Petillot, J. Salvi, "Independent Local Mapping for Large-Scale SLAM", *Proceedings of the 4th European Conference on Mobile Robotics,* ***ECMR'09***, Dubrovnik (Croatia) September 23-25, pp. 67-76, 2009.

- J. Aulinas, "Submapping and Hierarchical SLAM: Selective Submap Joining SLAM", *Workshop on Stochastic Filtering, Smoothing and Estimation for Multi-Sensor Target Tracking and Data Fusion,* ***ICMS***, Edinburgh (Scotland) November 9, 2009.

- J. Aulinas, Y.R. Petillot, J. Salvi, X. Lladó, "The SLAM problem: a survey", *11th International Conference of the Catalan Association for Artificial Intelligence,* ***CCIA'08***, Sant MartÃ d'EmpÃºries (Spain) October 22-24, 2008. Frontiers in Artificial Intelligence and Applications, vol. 184, pp. 363-371, 2008.

- J. Salvi, Y. Petillot, S.J. Thomas, J. Aulinas, "Visual SLAM for Underwater Vehicles using Video Velocity Log and Natural Landmarks", *MTS/IEEE Oceans Conference,* ***OCEANS'08***, Quebec City (Canada) September 15-18, 2008

Scientific collaborations

This thesis was developed during the period between 2008-2011. A large part was made at VICOROB at the University of Girona. However, a total of 6 months were spent at Heriot Watt University (Edinburgh, Scotland) for a research stay in the OSL group. Several meetings occured during the whole period, according to the following information:

- A 2 week stay with the OSL group at Heriot Watt University. Period: 01/03/2008 – 15/03/2008. Supervisor: Prof. Yvan R. Petillot.

- A 6 month stay with the OSL group at Heriot Watt University. Period: 01/09/2009 – 28/02/2010. Supervisor: Prof. Yvan R. Petillot.

- A 2 week stay with the OSL group at Heriot Watt University. Period: 28/03/2010 – 10/04/2010. Supervisor: Prof. Yvan R. Petillot.

Appendix A

Auxiliar Transformations

We use a 3-vector representation to define some fundamental coordinate frame operations. These operations are particularly useful in map-to-map relationships, when referring two uncorrelated maps to a common base reference.

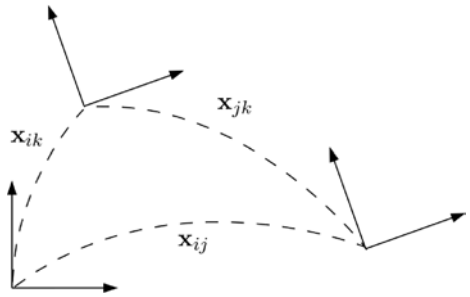


Figure A.1: Three arbitrary coordinate frames i, j and k, chosen to illustrate the operations used in stochastic mapping.

A.1 3-Degree Of Freedom (DOF) Pose Definition

The location of a reference j relative to a reference i (or transformation from i to j) can be expressed using a vector with a 3-Degree Of Freedom (DOF) vector $\mathbf{x}_{ij} = (x_{ij}\ y_{ij}\ \psi_{ij})^{\mathrm{T}}$ (see Figure A.1). Two basic operations used in stochastic mapping are composition of transformations and inverse transformation, which were suggested by [Smith 1988] using operators $\oplus$ and $\ominus$, respectively.

A.2 Transformations and their Jacobians in 2D

Given pose vectors $\mathbf{x}_{ij} = (x_{ij}\ y_{ij}\ \psi_{ij})^{\mathrm{T}}$ and $\mathbf{x}_{jk} = (x_{jk}\ y_{jk}\ \psi_{jk})^{\mathrm{T}}$, the *composition* operation yields frame k with respect to frame i (i.e., $\mathbf{x}_{ik} = (x_{ik}\ y_{ik}\ \psi_{ik})^{\mathrm{T}}$) as illustrated in Eq. (A.1). Its associated Jacobians are a useful quantity that allow us to compute a first-order covariance estimate of $\mathbf{x}_{ik}$.

$$\mathbf{x}_{ik} = \mathbf{x}_{ij} \oplus \mathbf{x}_{jk} = \begin{bmatrix} x_{ij} + x_{jk}\cos\psi_{ij} - y_{jk}\sin\psi_{ij} \\ y_{ij} + x_{jk}\sin\psi_{ij} + y_{jk}\cos\psi_{ij} \\ \psi_{ij} + \psi_{jk} \end{bmatrix} \tag{A.1}$$

$$J_{1\oplus}\{\mathbf{x}_{ij}, \mathbf{x}_{jk}\} = \frac{\partial(\mathbf{x}_{ij} \oplus \mathbf{x}_{jk})}{\partial \mathbf{x}_{ij}} = \begin{bmatrix} 1 & 0 & -x_{jk}\sin\psi_{ij} - y_{jk}\cos\psi_{ij} \\ 0 & 1 & x_{jk}\cos\psi_{ij} - y_{jk}\sin\psi_{ij} \\ 0 & 0 & 1 \end{bmatrix}$$

$$J_{2\oplus}\{\mathbf{x}_{ij}, \mathbf{x}_{jk}\} = \frac{\partial(\mathbf{x}_{ij} \oplus \mathbf{x}_{jk})}{\partial \mathbf{x}_{jk}} = \begin{bmatrix} \cos\psi_{ij} & -\sin\psi_{ij} & 0 \\ \sin\psi_{ij} & \cos\psi_{ij} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The inverse transformation is another fundamental operation. Given pose vector x_{ij}, the inverse operation yields frame i with respect to frame j, (i.e., x_{ji}), as shown in Eq. (A.2).

$$\mathbf{x}_{ij} = \ominus\mathbf{x}_{ji} = \begin{bmatrix} -x_{ij}\cos\psi_{ij} - y_{ij}\sin\psi_{ij} \\ x_{ij}\sin\psi_{ij} - y_{ij}\cos\psi_{ij} \\ -\psi_{ij} \end{bmatrix} \tag{A.2}$$

$$J_{\ominus}\{\mathbf{x}_{ij}\} = \frac{\partial(\ominus\mathbf{x}_{ij})}{\partial \mathbf{x}_{ij}} = \begin{bmatrix} -\cos\psi_{ij} & -\sin\psi_{ij} & -x_{ij}\sin\psi_{ij} - y_{ij}\cos\psi_{ij} \\ \sin\psi_{ij} & -\cos\psi_{ij} & x_{ij}\cos\psi_{ij} + y_{ij}\sin\psi_{ij} \\ 0 & 0 & -1 \end{bmatrix}$$

A.3 Transformations in 3D

Two basic 3D transformations are necessary in stochastic mapping: direct transformation (see eq. (A.3)) and inverse transformation (see eq. (A.4)). Given a 6DOF pose of a frame j with respect to frame i, i.e. $({}^{i}x_j,\ {}^{i}y_j,\ {}^{i}z_j,\ {}^{i}\phi_j,\ {}^{i}\theta_j,\ {}^{i}\psi_j)$, one can obtain the corresponding transformations ${}^{i}\mathcal{T}_j$ and ${}^{j}\mathcal{T}_i$ from eq. (A.5) and eq. (A.6).

$${}^{i}\mathcal{T}_j = \begin{bmatrix} {}^{i}\mathbf{R}_j & {}^{i}\mathbf{t}_j \\ \mathbf{0} & 1 \end{bmatrix} \tag{A.3}$$

$${}^{j}\mathcal{T}_i = \begin{bmatrix} {}^{i}\mathbf{R}_j^t & -{}^{i}\mathbf{R}_j^t \Delta {}^{i}\mathbf{t}_j \\ \mathbf{0} & 1 \end{bmatrix} \tag{A.4}$$

$$
\begin{aligned}
{}^{i}\mathbf{R}_{j} &= rotz(\psi)^{t} roty(\theta)^{t} rotz(\phi)^{t} \\
&= \begin{bmatrix} cos(\psi) & sin(\psi) & 0 \\ -sin(\psi) & cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}^{t} \\
&\quad \begin{bmatrix} cos(\theta) & 0 & -sin(\theta) \\ 0 & 1 & 0 \\ sin(\theta) & 0 & cos(\theta) \end{bmatrix}^{t} \\
&\quad \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\phi) & cos(\phi) \\ 0 & -sin(\phi) & cos(\phi) \end{bmatrix}^{t}
\end{aligned} \tag{A.5}
$$

$$
{}^{i}\mathbf{t}_{j} = [{}^{i}x_{j}\ {}^{i}y_{j}\ {}^{i}z_{j}]^{t} \tag{A.6}
$$

Bibliography

[Artieda 2009] J. Artieda, J.M. Sebastian, P. Campoy, J.F. Correa, I.F. Mondragón, C. Martínez and M. Olivares. *Visual 3-D SLAM from UAVs.* Journal of Intelligent and Robotic Systems, vol. 55, pages 299–321, August 2009. (Cited on page 14.)

[Atallah 2005] L. Atallah, C. Shang and R. Bates. *Object Detection at Different Resolution in Archaeological Side-scan Sonar Images.* In IEEE Oceans (OCEANS'05), Brest (France), June 2005. (Cited on page 70.)

[Aulinas 2010a] J. Aulinas, X. LLadó, J. Salvi and Y. Petillot. *Feature based SLAM using Side-Scan salient objects.* In MTS/IEEE Oceans (OCEANS'10), Seattle (USA), September 2010. (Cited on page 66.)

[Aulinas 2010b] J. Aulinas, X. LLadó, J. Salvi and Y. Petillot. *Selective Submap Joining for underwater Large Scale 6-DOF SLAM.* In IEEE/RSJ International conference on Intelligent Robots and Systems (IROS), pages 2552–2557, Taipei (Taiwan), October 2010. (Cited on page 88.)

[Aulinas 2010c] J. Aulinas, X. LLadó, J. Salvi and Y. Petillot. *SLAM base Selective Submap Joining for the Victoria Park Dataset.* In 7th IFAC Symposium on Intelligent Autonomous Vehicles (IFAC/IAV), pages 67–76, Lecce(Italy), September 2010. Morgan Kaufmann Publishers Inc. (Cited on page 60.)

[Aulinas 2011] J. Aulinas, A. Fazlollahi, J. Salvi, X. LLadó and R. Garcia. *Robust automatic landmark detection for underwater SLAM using side-scan sonar imaging.* In 11th International Conference on Mobile Robots and Competitions, Lisboa (Portugal), April 6-10 2011. (Cited on page 70.)

[Bailey 2006] T. Bailey and H. Durrant-Whyte. *Simultaneous localization and mapping (SLAM): Part II Part II State of the Art.* IEEE Robotics and Automation Magazine, vol. 13, no. 3, pages 108–117, 2006. (Cited on page 12.)

[Balasuriya 2002] A. Balasuriya and T. Ura. *Vision-based underwater cable detection and following using AUVs.* In MTS/IEEE Oceans, volume 3, pages 1582 – 1587, 2002. (Cited on pages 31, 33 and 34.)

[Barkby 2009] S. Barkby, S.B. Williams, O. Pizarro and M. Jakuba. *Incorporating prior bathymetric maps with distributed particle bathymetric SLAM for improved AUV navigation and mapping.* In Proceedings MTS/IEEE Oceans Conference and Exhibition, volume 2, pages 1–7, Biloxi (USA), 2009. (Cited on pages 26 and 29.)

[Bay 2006] H. Bay, T. Tuytelaars and L. J. Van Gool. *SURF: Speeded up robust features.* In In Proceedings of the ECCV'06: European Conference on Computer Vision, volume 3952, pages 404–417, 2006. (Cited on page 33.)

[Bell 2002] J. M. Bell, E. Dura, S. Reed, Y. R. Petillot and D. M. Lane. *Extraction and classification of objects from sidescan sonar.* In IEEE Workshop on Nonlinear and Non-Gaussian Signal Processing, pages 404–417, 2002. (Cited on pages 30 and 32.)

[Bosse 2004] M. Bosse, P.M. Newman, J.J. Leonard and S. Teller. *SLAM in large scale cyclic environments using the atlas framework.* International Journal Robotics Research, vol. 23, no. 12, pages 1113–1139, 2004. (Cited on pages 20 and 25.)

[Bouguet 2009] Jean-Yves Bouguet. *Camera Calibration Toolbox for Matlab*, 2009. Retrieved 02 December 2009. (Cited on page 75.)

[Burgard 1999] W. Burgard, D. Fox, H. Jans, C. Matenar and S. Thrun. *Sonar-based mapping with mobile robots using EM.* In Proceedings 16th International Conference on Machine Learning, pages 67–76. Morgan Kaufmann, 1999. (Cited on pages 18 and 19.)

[Caiti 2005] A. Caiti, A. Garulli, F. Livide and D. Prattichizzo. *Localization of autonomous underwater vehicles by floating acoustic buoys: a set membership approach.* IEEE Journal of Oceanic Engineering, vol. 30, no. 1, pages 140–152, 2005. (Cited on page 24.)

[Canny 1986] J. Canny. *A Computational Approach To Edge Detection.* IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 8, no. 6, pages 679–698, 1986. (Cited on page 32.)

[Carreras 2003] M. Carreras, P. Ridao, R. Garcia and T. Nicosevici. *Vision-based localization of an underwater robot in a structured environment.* In Proceedings IEEE International Conference Robotics and Automation, volume 1, pages 971–976, 2003. (Cited on page 26.)

[Castellanos 2007] J. A. Castellanos, R. Martínez-Cantin, J. D. Tardós and J. Neira. *Robocentric map joining: Improving the consistency of EKF-SLAM.* Robotics and Autonomous Systems, vol. 55, no. 1, pages 21–29, 2007. (Cited on pages 15, 20 and 81.)

[Chandran 2002] V. Chandran, S. Elgar and A. Nguyen. *Detection of Mines in Acoustic Images Using Higher Order Spectral Features.* IEEE Journals of Oceanic Engineering, vol. 27, no. 3, pages 610–618, 2002. (Cited on page 30.)

[Chen 2007] Z. Chen, J. Samarabandu and R. Rodrigo. *Recent advances in simultaneous localization and map-building using computer vision.* Advanced Robotics, vol. 21(3-4), pages 233–265, 2007. (Cited on page 18.)

[Ciany 2000] C. M. Ciany and J. Huang. *Computer aided detection/computer aided classification and data fusion algorithms for automated detection and classification of underwater mine.* In Proc. MTS/IEEE Oceans Conference and Exhibition, volume 1, pages 277–284, 2000. (Cited on pages 30 and 32.)

[Civera 2008] J. Civera, A.J. Davison and J.M.M. Montiel. *Inverse Depth Parametrization for Monocular SLAM.* IEEE Transactions on Robotics, vol. 24, no. 5, pages 932–945, 2008. (Cited on pages 93 and 95.)

[Civera 2010] J. Civera, O.G. Grasa, A.J. Davison and J.M.M. Montiel. *1-Point RANSAC for EKF Filtering: Application to Real-Time Structure from Motion and Visual Odometry.* Journal of Field Robotics, vol. 27, no. 5, pages 609–631, 2010. (Cited on page 14.)

[Coiras 2007] E. Coiras, Y.R. Petillot and D. M. Lane. *Multiresolution 3-D Reconstruction From Side-Scan Sonar Images.* IEEE Transactions on Image Processing, vol. 16, no. 2, pages 382–390, 2007. (Cited on pages 30, 31 and 32.)

[Cufi 2002] X. Cufi, R. Garcia and P. Ridao. *An approach to vision-based station keeping for an unmanned underwater vehicle.* In IEEE/RSJ International Conference on Intelligent Robots and System, volume 1, pages 799 – 804, 2002. (Cited on pages 31 and 34.)

[Davison 2002] A.J. Davison. *SLAM with a Single Camera.* SLAM/CML Workshop ICRA, 2002. (Cited on page 19.)

[Davison 2007] A.J. Davison, I. Reid, N. Molton, and O. Stasse. *MonoSLAM: Real-Time Single Camera SLAM.* IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), vol. 29(6), pages 1052 – 1067, 2007. (Cited on pages 14 and 16.)

[Durrant-Whyte 1988] H. F. Durrant-Whyte. *Uncertain geometry in robotics.* IEEE Journal of Robotics and Automation, vol. 4, no. 1, pages 23–31, 1988. (Cited on page 14.)

[Durrant-Whyte 2006] H. Durrant-Whyte and T. Bailey. *Simultaneous localization and mapping (SLAM): Part I The Essential Algorithms.* IEEE Robotics and Automation Magazine, vol. 13, no. 2, pages 99–108, 2006. (Cited on pages 3 and 12.)

[Erol 2007] M. Erol, L. Vieira and M. Gerla. *AUV-aided localization for underwater sensor networks.* In International Conference on Wireless Algorithms, Systems and Applications (WASA 2007), pages 44–54, August 2007. (Cited on page 24.)

[Estrada 2005] C. Estrada, J. Neira and J.D. Tardós. *Hierarchical SLAM: real-time accurate mapping of large environments.* IEEE Transactions on Robotics, vol. 21, no. 4, pages 588–596, August 2005. (Cited on pages 14, 21, 23 and 25.)

[Eustice 2005a] R. Eustice, O. Pizarro and H. Singh. *Large-Area Visually Augmented Navigation for Autonomous Underwater Vehicles.* PhD thesis, Massachusetts Institute of Technology and the Woods Hole Oceanographic Institution, 2005. (Cited on page 27.)

[Eustice 2005b] R. Eustice, H. Singh, J. Leonard, M. Walter and R. Ballard. *Visually Navigating the RMS Titanic with SLAM Information Filters.* In Proceedings of Robotics Science and Systems, pages 57–64, June 2005. (Cited on pages 27 and 31.)

[Eustice 2006a] R.M. Eustice, H. Singh, J. Leonard and M. Walter. *Visually Mapping the RMS Titanic: Conservative Covariance Estimates for SLAM Information Filters.* International Journal of Robotics Research, vol. 25(12), pages 1223–1242, 2006. (Cited on page 27.)

[Eustice 2006b] R.M. Eustice, H. Singh and J.J. Leonard. *Optimization of the Simultaneous Localization and Map Building Algorithm for Real Time Implementation.* IEEE Transactions on Robotics and Automation, vol. 22, no. 6, pages 1100–1114, 2006. (Cited on page 20.)

[Eustice 2008] R.M. Eustice, H. Singh, J. Leonard and M. Walter. *Visually Augmented Navigation for Autonomous Underwater Vehicles.* IEEE Journal of Oceanic Engineering, vol. 33, no. 2, pages 103–122, 2008. (Cited on pages 14, 27, 29, 34 and 81.)

[Fairfield 2007] N. Fairfield, G. Kantor and D. Wettergreen. *Real-time slam with octree evidence grids for exploration in underwater tunnels.* Journal of Field Robotics, vol. 24, pages 3–21, 2007. (Cited on pages 14, 26 and 29.)

[Fairfield 2008] N. Fairfield and D. Wettergreen. *Active localization on the ocean floor with multibeam sonar.* In In Proceedings of MTS/IEEE OCEANS, pages 1–10, Quebec City (CANADA), 2008. (Cited on pages 26 and 29.)

[Folkesson 2003] J. Folkesson and H. Christensen. *Outdoor Exploration and SLAM using a Compressed Filter.* In Proceedings of the IEEE Internstional Conference on Robotics and Automation, volume 1, pages 419–426, 2003. (Cited on page 20.)

[Freund 1995] Y. Freund and R. E. Schapire. *A decision-theoretic generalization of on-line learning and an application to boosting.* Computational Learning Theory: Eurocolt 95, pages 23–37, 1995. (Cited on page 71.)

[Garcia 2001] R. Garcia, J. Batlle, X. Cufi and J. Amat. *Positioning an underwater vehicle through image mosaicking.* In IEEE International Conference on Robotics and Automation, volume 3, pages 2779 – 2784, 2001. (Cited on page 31.)

[Garcia 2002] R. Garcia, T. Nicosevici and X. Cufi. *On the Way to Solve Lighting Problems in Underwater Imaging.* In IEEE OCEANS Conference (OCEANS), pages 1018–1024, 2002. (Cited on page 32.)

[Garcia 2006] R. Garcia, X. Cufi, P. Ridao and M. Carreras. Constructing photo-mosaics to assist uuv navigation and station-keeping, chapitre 9, pages 195–234. Robotics and Automation in the Maritime Industries, 2006. (Cited on page 31.)

[Gracias 2003] N. Gracias, S. van der Zwaan, A. Bernardino and J. Santos-Victor. *Mosaic-based navigation for autonomous underwater vehicles.* IEEE Journal of Oceanic Engineering, vol. 28, no. 4, pages 609–624, 2003. (Cited on pages 31, 33, 34 and 82.)

[Gracias 2005] N. Gracias and S. Negahdaripour. *Underwater Mosaic Creation using Video sequences from Different Altitudes.* In Proceedings of MTS/IEEE OCEANS, volume 2, pages 1295–1300, 2005. (Cited on pages 31, 33 and 34.)

[Gracias 2008] N. Gracias, S. Negahdaripour, L. Neumann, R. Prados and R. Garcia. *A motion compensated filtering approach to remove sunlight flicker in shallow water images.* In Proceedings of the MTS/IEEE Oceans 2008 Conference (OCEANS), pages 1018 – 1024, 2008. (Cited on pages 32 and 74.)

[Grisetti 2005] G. Grisetti. *Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling.* In In Proc. of the IEEE Internatinal Conference on Robotics and Automation, (ICRA), pages 2443–2448, 2005. (Cited on page 12.)

[Guivant 2000a] J. Guivant. *Victoria park data set*, 2000. Retrieved 12 November 2008. (Cited on pages 52 and 103.)

[Guivant 2000b] J. Guivant, E. Nebot and S. S. Baiker. *Autonomous navigation and map building using laser range sensors in outdoor applications.* Journal of Robotic Systems, vol. 17, pages 3817–3822, 2000. (Cited on page 12.)

[Guivant 2001] J. Guivant and E. Nebot. *Optimization of the Simultaneous Localization and Map Building Algorithm for Real Time Implementation.* IEEE Transactions on Robotics and Automation, vol. 17, no. 3, pages 242–257, 2001. (Cited on pages 17, 19, 20, 25 and 48.)

[Guivant 2002a] J. Guivant, F. Masson and E. Nebot. *Simultaneous localization and map building using natural features and absolute information.* International Journal of Robotics and Autnonomous Systems, vol. 40, pages 79–90, 2002. (Cited on page 61.)

[Guivant 2002b] J. Guivant and E. Nebot. *Improving Computational and Memory Requirements of Simultaneous Localization and Map Building Algorithms.* In IEEE Transaction of Robotics and Automation, volume 3, pages 2731 – 2736, 2002. (Cited on pages 20 and 56.)

[Harper 1977] H.A. Harper, V.W. Rodwell and P.A. Mayes. Review of physiological chemistry. 16th ed., Lange Medical Publications, 1977. (Cited on page 1.)

[Harris 1988] C. Harris and M. Stephens. *A combined corner and edge detector.* In In Proceedings of the 4th Alvey Vision Conference, pages 147–151, Manchester (UK), 1988. (Cited on page 33.)

[Hartley 2000] R. Hartley and A. Zisserman. Multiple view geometry in computer vision. Cambridge University Press, 2000. (Cited on pages 31, 92 and 93.)

[Hunt 1974] M.M. Hunt, W.M. Marquet, D.A. Moller, K.R. Peal, W.K. Smith and R.C. Spindel. *An acoustic navigation system, Technical Report WHOI-74-6.* Rapport technique, Woods Hole Oceanographic Institution, December 1974. (Cited on page 24.)

[Jensfelt 2006] P. Jensfelt, D. Kragic, J. Folkesson and M. Björkman. *A Framework for Vision Based Bearing Only 3D SLAM.* In Proceedings - IEEE International Conference on Robotics and Automation, ICRA, numéro 1641990, pages 1944–1950, 2006. (Cited on pages 14, 16 and 19.)

[Johnson-Roberson 2010] M. Johnson-Roberson, O. Pizarro, S.B. Williams and I.J. Mahon. *Generation and Visualization of Large-Scale Three-Dimensional Reconstructions from Underwater Robotic Surveys.* In Journal of Field Robotics, vol. 27, no. 1, pages 21–51, 2010. (Cited on pages 14 and 31.)

[Johnson 1994] S. G. Johnson and M. A. Deaett. *The application of automated recognition techniques to side-scan sonar imagery.* IEEE Journals of Oceanic Engineering, vol. 19, pages 138–144, 1994. (Cited on page 30.)

[Kahanov 2001] Y. Kahanov and J. Royal. *Analysis of hull remains of the Dor D vessel.* International Journal of Nautical Archaeology, vol. 30, no. 2, pages 257–265, 2001. (Cited on page 31.)

[Kalman 1960] R.E. Kalman. *A new approach to linear filtering and prediction problems.* Journal of Basic Engineering, vol. 82, no. 1, pages 35–45, 1960. (Cited on page 39.)

[Kim 2007] J. Kim and S. Sukkarieh. *Real-time implementation of airborne inertial-SLAM.* Robotics and Autonomous Systems, vol. 55, pages 62–71, 2007. (Cited on page 14.)

[Knight 2001] J. Knight, A. Davison and I. Reid. *Towards constant time SLAM using postponement.* In Proceedings IEEE/RJS International Conference on Intelligent Robots and Systems, pages 406–412, 2001. (Cited on page 20.)

[Leonard 1991] J.J. Leonard and H.F. Durrant-whyte. *Simultaneous map building and localization for an autonomous mobile robot.* In IEEE/RSJ International Workshop on: Intelligence for Mechanical Systems, Intelligent Robots and Systems (IROS), pages 1442–1447, 1991. (Cited on page 14.)

[Leonard 2001] J.J Leonard and H.J.S. Feder. *Decoupled stochastic mapping [for mobile robot and AUV navigation].* IEEE Journal of Oceanic Engineering, vol. 26, no. 4, pages 561–571, 2001. (Cited on pages 14, 20, 26 and 29.)

[Leonard 2003] J.J. Leonard and P.M. Newman. *Consistent, convergent, and constant-time SLAM.* In Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI'03), pages 1143–1150, San Francisco, CA, USA, 2003. Morgan Kaufmann Publishers Inc. (Cited on pages 14, 16 and 19.)

[Lianantonakis 2007] M. Lianantonakis and Y.R. Petillot. *Side-scan Sonar Segmentation Using Texture Descriptors and Active Contours.* IEEE Journal of Oceanic Engineering, vol. 23, pages 744–752, 2007. (Cited on pages 30 and 32.)

[Lowe 2004] D. Lowe. *Distinctive Image Features from Scale-Invariant Keypoints.* International Journal of Computer Vision, vol. 60, no. 2, pages 91–110, 2004. (Cited on page 33.)

[Mahon 2004] I. Mahon and S. Williams. *SLAM using natural features in an underwater environment.* In IEEE Control, Automation, Robotics and Vision Conference, volume 3, pages 2076–2081, December 2004. (Cited on page 13.)

[Mallios 2009] A. Mallios, P. Ridao, E. Hernàndez and D. Ribas. *Pose-Based SLAM with Probabilistic Scan Matching Algorithm Using a Mechanical Scanned Imaging Sonar.* In Proceedings of the 3rd Congrés Internacional sobre Tecnologia Marina. Martech'09, numéro 8 de Instrumentation viewpoint, November 2009. (Cited on page 29.)

[Maurelli 2008] F. Maurelli, S. Krupinski, Y. Petillot and J. Salvi. *A Particle Filter Approach for AUV Localization.* In MTS/IEEE OCEANS, OCEANS'08, pages 1–7, September 2008. (Cited on page 25.)

[Mei 2009] C. Mei, G. Sibley, M. Cummins, P. Newman and I. Reid. *A Constant Time Efficient Stereo SLAM System.* In Proceedings of the British Machine Vision Conference (BMVC), London, September 2009. (Cited on page 14.)

[Mindru 1999] F. Mindru, T. Moons and L. Van Gool. *Recognizing color patterns irrespective of viewpoint and illumination.* In In Proceedings of Conference on Computer Vision and Pattern Recognition, pages 368–373, 1999. (Cited on page 33.)

[Momchilova 2007] Yana Momchilova. *Invariance with Optic Flow*, 2007. (Cited on page 75.)

[Montemerlo 2002] M. Montemerlo, S. Thrun, D. Koller and B. Wegbreit. *FastSLAM: A factored solution to the simultaneous localization and mapping problem.* In Proceedings of the AAAI National Conference on Artificial Intelligence, pages 593–598, 2002. (Cited on pages 17 and 19.)

[Montemerlo 2003] M. Montemerlo, S. Thrun, D. Roller and B. Wegbreit. *FastSLAM 2.0: an improved particle filtering algorithm for simultaneous localization and mapping that provably converges.* In Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI'03), pages 1151–1156, San Francisco, CA, USA, 2003. Morgan Kaufmann Publishers Inc. (Cited on pages 17 and 19.)

[Montemerlo 2007] M. Montemerlo and S. Thrun. FastSLAM: A scalabale method for the simultaneous localizaton and mapping problem in robotics, volume 27. Springer Tracts in Advanced Robotics, 2007. ISBN 978-3-540-46399-3. (Cited on page 17.)

[Negahdaripour 2006] S. Negahdaripour and P. Firoozfam. *An ROV stereovision system for ship-hull inspection.* In IEEE Journal on Oceanic Engineering, volume 31, pages 551–564, 2006. (Cited on page 31.)

[Neira 2001] J. Neira and J.D. Tardós. *Data Association in Stochastic Mapping Using the Joint Compatibility Test.* IEEE Transactions on Robotics and Automation, vol. 17, no. 6, pages 890 – 897, December 2001. (Cited on page 63.)

[Neira 2003] J. Neira, J. D. Tardós and J. A. Castellanos. *Linear time vehicle relocation in SLAM.* In IEEE International Conference on Robotics and Automation, volume 1, pages 427–433, Taipei (Taiwan), September 2003. (Cited on page 64.)

[Newman 2003] P. Newman, J. J. Leonard and R. R. Rikoski. *Towards Constant-Time SLAM on an Autonomous Underwater Vehicle Using Synthetic Aperture Sonar.* In Proceedings of the International Symposium of Robotics Research, pages 409–420, Sienna (Italy), 2003. (Cited on pages 20, 25, 26 and 29.)

[Newman 2006] P. Newman, D. Cole and K. Leong Ho. *Outdoor SLAM using Visual Appearance and Laser Ranging.* In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Orlando, Florida (USA), May 2006. (Cited on page 14.)

[Nicosevici 2007] T. Nicosevici, R. Garcia, S. Negahdaripour, M. Kudzinava and J. Ferrer. *Identification of Suitable Interest Points Using Geometric and Photometric Cues in Motion Video for Efficient 3-D Environmental Modeling.* In IEEE International Conference on Robotics and Automation, pages 4969–4974, Roma (Italy), April 2007. (Cited on pages 33 and 34.)

[Nicosevici 2008] T. Nicosevici and R. Garcia. *On-line robust 3D Mapping using structure from motion cues.* In MTS/IEEE Techno-Ocean Conference (Oceans'08), Kobe (Japan), April 2008. (Cited on pages 14 and 31.)

[NOAA 2008] NOAA, Office of Ocean Exploration and Research. *Types of Offshore Oil and Gas Structures.* NOAA Ocean Explorer: Expedition to the Deep Slope. National Oceanic and Atmospheric Administration, December 2008. Retrieved 10 February 2011. (Cited on page 2.)

[Olson 2006] E. Olson, J. Leonard and S. Teller. *Robust Range-Only Beacon Localization.* Journal of Oceanic Engineering, vol. 31, no. 4, pages 949–958, 2006. (Cited on page 24.)

[Ortiz 2002] A. Ortiz, M. Simo and G. Oliver. *A vision system for an underwater cable tracker.* International Journal of Machine Vision and Applications, vol. 13, no. 3, pages 129–140, 2002. (Cited on pages 33 and 34.)

[Ortiz 2008] A. Ortiz, J. Antich and G. Oliver. *A particle filter-based approach for tracking undersea narrow telecommunication cables.* Machine Vision and Applications, vol. 22, no. 2, pages 283–302, 2008. (Cited on pages 31 and 34.)

[Parry 2007] M.L. Parry, O.F. Canziani, J.P. Palutikof, P.J. van der Linden and C.E. Hanson. Contribution of working group ii to the fourth assessment report of the intergovernmental panel on climate change. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, 2007. (Cited on page 1.)

[Paz 2007a] L.M Paz, J. Guivant, J.D. Tardós and J. Neira. *Data Association in O(n) for Divide and Conquer SLAM.* In Robotics: Science and Systems, volume 3, pages 281–288, Atlanta, USA, June 2007. (Cited on page 64.)

[Paz 2007b] L.M. Paz, P. Jensfelt, J. D. Tardós and J. Neira. *EKF SLAM updates in O(n) with Divide and Conquer SLAM.* In IEEE International Conference on Robotics and Automation, volume 4209325, pages 1657–1663, 2007. (Cited on page 21.)

[Paz 2008] L.M. Paz, J.D. Tardós and J. Neira. *Divide and Conquer: EKF SLAM in O(n).* IEEE Transactions on Robotics, vol. 24, no. 5, pages 1107–1120, October 2008. (Cited on pages 21, 25 and 51.)

[Perry 2004] S. W. Perry and L. Guan. *Pulse-Length-Tolerant Features and Detectors for Sector-Scan Sonar Imagery.* IEEE Journal of Oceanic Engineering, vol. 29, no. 1, pages 138–156, January 2004. (Cited on pages 30 and 32.)

[Piniés 2008] P. Piniés and J.D. Tardós. *Large Scale SLAM Building Conditionally Independent Local Maps: Application to Monocular Vision.* IEEE Transactions on Robotics, vol. 24, no. 5, pages 1094–1106, October 2008. (Cited on pages 22 and 24.)

[Piniés 2009] P. Piniés, L.M. Paz and J.D. Tardós. *CI-Graph: An efficient approach for Large Scale SLAM.* In IEEE International Conference on Robotics and Automation, pages 3913–3920, Kobe, Japan, May 2009. (Cited on pages 16, 22, 23 and 25.)

[Pizarro 2003] O. Pizarro and H. Singh. *Toward Large-Area Underwater Mosaicing for Scientific Applications.* IEEE Journal of Oceanic Engineering, vol. 28, no. 4, pages 651–672, 2003. (Cited on pages 33 and 34.)

[Pizarro 2004] O. Pizarro, R. Eustice and H. Singh. *Large Area 3D Reconstructions from Underwater Surveys.* In Ocean '04 - MTS/IEEE Techno-Ocean: Bridges across the Oceans - Conference Proceedings, volume 2, pages 678–687, 2004. (Cited on pages 20 and 83.)

[Reed 2003] S. Reed, Y.R. Petillot and J. Bell. *An Automatic Approach to the Detection and Extraction of Mine Features in Side-scan Sonar.* IEEE Journal of Oceanic Engineering, vol. 28, no. 1, pages 90–105, 2003. (Cited on pages 30 and 32.)

[Reed 2006] S. Reed, I. Tena Ruiz, C. Capus and Y. Petillot. *The Fusion of Large Scale Classified Side-Scan Sonar Image Mosaics.* IEEE Transaction on Image Processing, vol. 15, no. 7, pages 2049–2060, 2006. (Cited on pages 30 and 32.)

[Ribas 2008a] D. Ribas. *Underwater SLAM for Structured Environment Using an Imaging Sonar.* PhD thesis, University of Girona, 2008. (Cited on page 28.)

[Ribas 2008b] D. Ribas, P. Ridao, J.D. Tardós and J. Neira. Underwater SLAM in Man Made Structured Environments, vol. 25, no. 11–12, pages 898–921, 2008. (Cited on pages 13, 14, 16, 19, 27, 29, 60, 61 and 81.)

[Roman 2007] C. N. Roman and H. Singh. *A Self-Consistent Bathymetric Mapping Algorithm.* Journal of Field Robotics, vol. 24, no. 1–2, pages 23–50, 2007. (Cited on pages 26 and 29.)

[Ruiz 2001] I. Tena Ruiz, Y. Petillot, D. M. Lane, and C. Salson. *Feature extraction and data association for AUV concurent mapping and localization.* In Proceedings of the IEEE International Conference on Robotics and Automation, pages 2785–2790, Seoul (Korea), 2001. (Cited on pages 30 and 32.)

[Sáez 2006] J.M. Sáez, A. Hogue, F. Escolano and M. Jenkin. *Underwater 3D SLAM through Entropy Minimization.* In Proceedings of IEEE International Conference on Robotics and Automation, numéro 1642246, pages 3562–3567, 2006. (Cited on pages 14, 31 and 34.)

[Saisan 2008] P. Saisan and S. Kadambe. *Shape Normalized Subspace Analysis for Underwater Mine Detection.* In IEEE International Conference on Image Processing, volume 1, pages 1892–1895, 2008. (Cited on pages 30 and 32.)

[Salvi 2008] J. Salvi, Y. Petillot, S.J. Thomas and J. Aulinas. *Visual SLAM for Underwater Vehicles using Video Velocity Log and Natural Landmarks.* In MTS/IEEE OCEANS, pages 2082–2089, Quebec City (Canada), September 2008. (Cited on pages 14, 33 and 34.)

[Se 2002] S. Se, D. Lowe and J. Little. *Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks.* The international Journal of robotics Research, vol. 21(8), pages 735–758, 2002. (Cited on page 14.)

[Smith 1986] R.C. Smith and P. Cheeseman. *On the Representation and Estimation of Spatial Uncertainty.* International Journal of Robotics Research, vol. 5, no. 4, pages 56–68, 1986. (Cited on pages 14 and 25.)

[Smith 1988] R. Smith, M. Self and P. Cheeseman. *A stochastic map for uncertain spatial relationships.* In Proceedings of the 4th International Symposium on Robotics Research, pages 467–474, Cambridge, MA, USA, 1988. MIT Press. (Cited on pages 14 and 107.)

[Smith 1990] R. Smith, M. Self and P. Cheeseman. *Estimating uncertain spatial relationships in robotics.* Autonomous robot vehicles, pages 167–193, 1990. (Cited on page 14.)

[Tardós 2002] J.D. Tardós, J. Neira, P.M. Newman and J.J. Leonard. *Robust Mapping and Localization in Indoor Environments using Sonar Data.* International Journal of Robotics Research, vol. 21, no. 4, pages 311–330, 2002. (Cited on pages 16, 20, 25 and 45.)

[Tena-Ruiz 2004] I. Tena-Ruiz, S. Raucourt, Y. Petillot and D. Lane. *Concurrent mapping and localization using side-scan sonar.* IEEE Journal of Oceanic Engineering, vol. 29, no. 2, pages 442–456, 2004. (Cited on pages 28 and 29.)

[Thrun 2001] S. Thrun. *A Probabilistic Online Mapping Algorithm for Teams of Mobile Robots.* International Journal of Robotics Research, vol. 20, no. 5, pages 335–363, 2001. (Cited on page 18.)

[Thrun 2002] S. Thrun. *Robotic Mapping: A Survey.* In G. Lakemeyer and B. Nebel, editeurs, Exploring Artificial Intelligence in the New Millenium. Morgan Kaufmann, 2002. (Cited on pages 12, 13, 15 and 18.)

[Thrun 2003] S. Thrun and Y. Liu. *Multi-Robot SLAM With Sparse Extended Information Filers.* In Proceedings of the 11th International Symposium of Robotics Research (ISRR'03), volume 15, pages 254 – 266, 2003. (Cited on pages 16 and 19.)

[Thrun 2004a] S. Thrun, Y. Liu, D. Koller, A.Y. Ng, Z. Ghahramani and H. Durrant-Whyte. *Simultaneous localization and mapping with sparse extended information filters.* International Journal of Robotics Research, vol. 23, no. 7-8, pages 693–716, 2004. (Cited on pages 16 and 20.)

[Thrun 2004b] S. Thrun, C. Martin, Y. Liu, D. Hähnel, R. Emery Montemerlo, C. Deepayan and W. Burgard. *A Real-Time Expectation Maximization Algorithm for Acquiring Multi-Planar Maps of Indoor Environments with Mobile Robots.* IEEE Transactions on Robotics and Automation, vol. 20, no. 3, pages 433–443, 2004. (Cited on pages 16 and 19.)

[Thrun 2005] S. Thrun, D. Burgard and W. Fox. Probabilistic robotics. MIT Press, 2005. ISBN 0-262-20162-3. (Cited on page 16.)

[Viola 2001] P. A. Viola and M. J. Jones. *Rapid Object Detection using a Boosted Cascade of Simple Features.* In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), volume 1, pages 511 – 518, 2001. (Cited on pages 65, 66 and 68.)

[Viola 2004] P. Viola and M. Jones. *Robust Real-Time Face Detection.* International Journal of Computer Vision, vol. 57, no. 2, pages 137–154, 2004. (Cited on pages 66 and 71.)

[Walter 2005] M. Walter, Eustice, R. and J. Leonard. *A Provably Consistent Method for Imposing Exact Sparsity in Feature-based SLAM Information Filters.* In Proceedings of the 12th International Symposium of Robotics Research (ISRR), pages 214 – 234, 2005. (Cited on page 16.)

[Walter 2007] M.R. Walter, R.M. Eustice and J.J. Leonard. *Exactly Sparse Extended Information Filters for Feature-based SLAM.* International Journal of Robotics Research, vol. 26, no. 4, pages 335–359, 2007. (Cited on pages 16 and 20.)

[Walter 2008] M. Walter, F. Hover and J. Leonard. *SLAM for ship hull inspection using exactly sparse extended information filters.* In International Conference on Robotics and Automation, pages 1463–1470, Pasadena, CA, 2008. (Cited on pages 27, 29 and 30.)

[Wan 2001] E. Wan and R. Van Der Merwe. *Chapter 7 The Unscented Kalman Filter.* In Kalman Filtering and Neural Networks, pages 221–280. Wiley, 2001. (Cited on pages 16, 19 and 20.)

[Whitcomb 1999] L.L. Whitcomb, D.R. Yoerger and H. Singh. *Combined Doppler/LBL based navigation of underwater vehicles.* In International Symposium on Unmanned Untethered Submersible Technology, May 1999. (Cited on page 24.)

[Williams 2001] S.B. Williams. *Efficient Solutions to Auntonomous Mapping and Navigation Problems.* PhD thesis, Australian Center for Field Robotics, University of Sydney, 2001. (Cited on pages 25 and 29.)

[Williams 2002] S.B. Williams, G. Dissanayake and H. Durrant-whyte. *An Efficient Approach to the Simultaneous Localisation and Mapping Problem.* In Proceedings IEEE International Conference Robotics and Automation, volume 1, pages 406–411, 2002. (Cited on pages 20 and 25.)

[Williams 2004] S. Williams and I. Mahon. *Simultaneous localisation and mapping on the Great Barrier Reef.* In In Proceedings of the IEEE International Conference on Robotics and Automation, volume 2, pages 1771–1776, New Orleans, LA, 2004. (Cited on pages 14, 26 and 29.)

[Willis 1995] A.J. Willis, N.P. Dunnett, R. Hunt and J.P. Grime. *Does Gulf Stream Position Affect Vegetation Dynamics in Western Europe?* Oikos, vol. 73, no. 3, pages 408–410, 1995. (Cited on page 2.)

[Wirth 2008] S. Wirth, A. Ortiz, D. Paulus and G. Oliver. *Using Particle Filters for Autonomous Underwater Cable Tracking.* In IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles, volume 2, pages 221–280, Killaloe (Ireland), 2008. (Cited on pages 31 and 34.)

[Xu 2006] B. Xu, S.R. Pandian, M. Inoue anb N. Sakagami and S. Kawamura. *Model-based Sliding Mode Control of Underwater Robot Manipulators.* International Journal of Offshore and Polar Engineering, vol. 16, no. 3, pages 210–217, 2006. (Cited on page 3.)