ONLINE ACOUSTIC LOCALIZATION METHODS FOR AUTONOMOUS UNDERWATER VEHICLES

Guillem Vallicrosa Massaguer

Doctoral Thesis

Online Acoustic Localization Methods for Autonomous Underwater Vehicles

Guillem Vallicrosa Massaguer

2018

Doctoral Thesis

Online Acoustic Localization Methods for Autonomous Underwater Vehicles

Guillem Vallicrosa Massaguer

2018

Doctoral Program in Technology

Supervised by:

Pere Ridao

Thesis submitted to University of Girona
in fulfillment of the requirements for the degree of

Doctor of Philosophy

Certificate of Thesis Direction

Dr. Pere Ridao director of Computer Vision and Robotics Research Group (VICOROB) and member of Departament d'Arquitectura i Tecnologia de Computadors of Universitat de Girona,

Declares:

That the work entitled *Online Acoustic Localization Methods for Autonomous Underwater Vehicles* presented by Guillem Vallicrosa Massaguer to obtain the degree in Doctor of Philosophy has been developed under my supervision and fullfills the requirements to obtain the International Mention.

Therefore, in order to certify the aforesaid statement, I sign this document.

Girona, July 2018

Dr. Pere Ridao

Acknowledgements

The work presented in this thesis would have not been possible without the help of some amazing people that contributed directly or indirectly to this thesis.

M'agradaria començar agraint el suport de la meva família i especialment als meus pares. Sempre han estat al meu costat, m'han animat a seguir estudiant i m'han ajudat a arribar fins aquí.

Vull donar les gràcies també al meu supervisor, en Pere, per oferir-me la possibilitat de fer aquest doctorat (i participar en un munt de projectes), que m'ha permès aprendre tant i tant.

Un agraïment especial a tots els companys del CIRS, amb els quals hem passat molts bons moments i hem superat tots els entrebancs que ens hem trobat. Sense ells, els robots no funcionarien i no hi hauria ni experiments ni recerca en aquest laboratori. A part del doctorat també hem compartit activitats ben diverses, com passar setmanes a les Açores intentant que tot funcionés, tancar-nos una setmana per reescriure tot el codi dels robots, els esmorzars, els dinars, les festes, els partits de pàdel, que fan que aquest camí sigui més amè.

D'entre ells, vull destacar l'Albert i en Jep, companys des d'Enginyeria Industrial amb els que he compartit les misèries de *la cursa de cargols* que ha estat aquesta tesi, i que contra tot pronòstic he guanyat (o perdut segons com es miri). La Tali i en Narcís en les interminables discussions sobre com millorar l'arquitectura dels robots i sobre com fer cada experiment. En Juan David per la seva inestimable companyia, sobretot als experiments del MORPH. En Lluís, en David i en Carles per solucionar tots els problemes de hardware que se'ns presenten dia a dia. Als que ja han marxat del CIRS però hi han deixat la seva empremta: Arnau, Aggelos, Èric. Als que han començat el doctorat més tard que jo: Eduard, Klemen, Dina, Khadidja, ànims pel que us queda. A en Marc, sempre presionant perquè acabem la tesi d'una vegada. A la gent del P4, als de visió i els dels trens, que tot i que ens veiem menys, tots som una gran família. Tampoc oblidar-me de la Joseta, Mireia, Anna, Bego i Olga, que ens estalvien un munt de feina i ens solucionen un munt de problemes.

Thanks to Stefan Williams and all his team for the opportunity to visit their lab and go with them to a mapping campaign at the Great Barrier Reef.

A més, vull agrair a la Marta el seu suport i comprensió, sobretot durant la part final

d'aquesta tesi.

Last, I would like to thank the anonymous reviewers that contributed to improve this thesis with their invaluable comments. Also the many reviewers that contributed to the different publications derived from this work.

Finally I would like to thank the agencies that partially funded this work: the FPU12/05384 PhD grant from the Spanish Ministry of Science, the National projects 3DAUV, ARCHROV and COMAROB, as well as the European projects MORPH, EUROFLEETS2 and LOON-DOCK.

List of Publications

Publications in the compendium

The presented thesis is a compendium of the following research articles:

- **G. Vallicrosa**, P. Ridao, "H-SLAM: Rao-Blackwellized Particle Filter SLAM Using Hilbert Maps" in *Sensors*, vol. 18 no. 5 art. 1386 ISSN 1424-8220 DOI 10.3390/s18051386, May 2018. Quality index: [JCR2016 I&I IF 2.677, Q1 (10/58)].

- N. Palomeras, **G. Vallicrosa**, A. Mallios, J. Bosch, E. Vidal, N. Hurtós, M. Carreras, P. Ridao, "AUV homing and docking for remote operations" in *Ocean Engineering*, vol. 154 pp. 106-120 ISSN 0029-8018 DOI 10.1016/j.oceaneng.2018.01.114, February 2018. Quality index: [JCR2016 ME IF 1.894, Q1 (2/14)].

- **G. Vallicrosa**, P. Ridao, "Sum of Gaussian Single Beacon Range-Only Localization for AUV Homing" in *Annual Reviews in Control*, vol. 42 pp. 177-187 ISSN 1367-5788 DOI 10.1016/j.arcontrol.2016.09.007, September 2016. Quality index: [JCR2016 A&CS IF 2.627, Q2 (17/60)].

Publications derived from this thesis

The work developed in this thesis also led to the following publications:

- N. Hurtós, A. Mallios, N. Palomeras, J. Bosch, **G. Vallicrosa**, E. Vidal, D. Ribas, N. Gracias, M. Carreras, P. Ridao, "LOON-DOCK: AUV homing and docking for high-bandwidth data transmission" in *MTS/IEEE OCEANS 2017-Aberdeen*, pp. 1-7 ISSN 978-1-5090-5278-3 DOI 10.1109/OCEANSE.2017.8084806, Aberdeen, June 2017.

- **G. Vallicrosa**, J. Bosch, N. Palomeras, P. Ridao, M. Carreras, N. Gracias, "Autonomous homing and docking for AUVs using Range-Only Localization and Light Beacons" in *10th IFAC Conference on Control Applications in Marine Systems (CAMS 2016). Published in IFAC-PapersOnLine*, vol. 49 num. 23 pp. 54-60 ISSN 2405-8963 DOI 10.1016/j.ifacol.2016.10.321, Trondheim, September 2016.

- **G. Vallicrosa**, P. Ridao, D. Ribas, "AUV Single Beacon Range-Only SLAM with a SOG Filter" in *IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles (NGCUV'2015). Published in IFAC-PapersOnLine*, vol. 48 num. 2 pp. 26-31 ISSN 2405-8963 DOI 10.1016/j.ifacol.2015.06.005, Girona, April 2015.

- **G. Vallicrosa**, P. Ridao, D. Ribas, A. Palomer, "Active range-only beacon localization for AUV homing" in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014)*, pp. 2286-2291 ISSN 2153-0866 DOI 10.1109/IROS.2014.6942871, Chicago, September 2014.

- **G. Vallicrosa**, A. Palomer, D. Ribas, P. Ridao, "Realtime AUV Terrain Based Navigation with Octomap in a natural environment" in *First Iberian Robotics Conference (ROBOT 2013)*. Published in Advances in Intelligent Systems and Computing, vol. 252 pp. 41-53 ISSN 978-3-319-03412-6 DOI 10.1007/978-3-319-03413-3_4, Madrid, November 2013.

- **G. Vallicrosa**, A. Palomer, D. Ribas, P. Ridao, "Towards Realtime AUV SLAM with Occupancy Grids" in *MARTECH2013: 5th International Workshop on Marine Technology*. Published in Instrumentation Viewpoint, vol. 15 pp. 59-60 ISSN 1886-4864, Girona, October 2013.

- **G. Vallicrosa**, A. Palomer, D. Ribas, P. Ridao, "Realtime AUV Terrain Based Navigation with Octomap" in *XXXIV Jornadas de Automática*, Terrassa, September 2013.

Other publications

Parallel work at the time of this thesis led to the following publications:

- D. Centelles, E. Moscoso, **G. Vallicrosa**, N. Palomeras, J. Sales, J.V. Marti, R. Marin, P. Ridao, P.J. Sanz, "Wireless HROV control with compressed visual feedback over an acoustic link" in *MTS/IEEE OCEANS 2017-Aberdeen*, pp. 1-7 ISSN 978-1-5090-5278-3 DOI 10.1109/OCEANSE.2017.8084979, Aberdeen, June 2017.

- F. Mandić, N. Mišković, N. Palomeras, M. Carreras, **G. Vallicrosa**, "Mobile beacon control algorithm that ensures observability in single range navigation" in *10th IFAC Conference on Control Applications in Marine Systems (CAMS 2016). Published in IFAC-PapersOnLine*, vol. 49 num. 23 pp. 48-53 ISSN 2405-8963 DOI 10.1016/j.ifacol.2016.10.320, Trondheim, September 2016.

- J.D. Hernandez, E. Vidal, **G. Vallicrosa**, E. Galceran, M. Carreras, "Online Path Planning for Autonomous Underwater Vehicles in Unknown Environments" in *IEEE International Conference on Robotics and Automation (ICRA 2015)*, pp. 1152-1157 ISSN 1050-4729 DOI 10.1109/ICRA.2015.7139336, Seattle, May 2015.

- J.D. Hernandez, E. Vidal, **G. Vallicrosa**, E. Pairet, M. Carreras, "Simultaneous Mapping and Planning for Autonomous Underwater Vehicles in Unknown Environments" in *MTS/IEEE OCEANS 2015-Genova*, pp. 1-6 ISSN 978-1-4799-8736-8 DOI 10.1109/OCEANS-Genova.2015.7271469, Genova, May 2015.

- A. Palomer, P. Ridao, D. Ribas, **G. Vallicrosa**, "Multi-beam Terrain/Object Classification for Underwater Navigation Correction" in *MTS/IEEE OCEANS 2015-Genova*, pp. 1-5 ISSN 978-1-4799-8736-8 DOI 10.1109/OCEANS-Genova.2015.7271587, Genova, May 2015.

- J.D. Hernandez, **G. Vallicrosa**, E. Vidal, E. Pairet, M. Carreras, P. Ridao, "On-line 3D Path Planning for Close-proximity Surveying with AUVs" in *IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles (NGCUV'2015). Published in IFAC-PapersOnLine*, vol. 48 num. 2 pp. 50-55 ISSN 2405-8963 DOI 10.1016/j.ifacol.2015.06.009, Girona, April 2015.

- N. Palomeras, A. Peñalver, M. Massot-Campos, **G. Vallicrosa**, P. Negre, J. Fernandez, P. Ridao, P. Sanz, G. Oliver-Codina, A. Palomer, "I-AUV Docking and Intervention in a Subsea Panel" in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014)*, pp. 2279-2285 ISSN 2153-0866 DOI 10.1109/IROS.2014.6942870, Chicago, September 2014.

- A. Palomer, P. Ridao, D. Ribas, A. Mallios, **G. Vallicrosa**, "Octree-Based Subsampling Criteria for Bathymetric SLAM" in *XXXV Jornadas de Automatica*, pp. 1-6 ISSN 978-84-697-0589-6, Valencia, September 2014.

- N. Palomeras, P. Ridao, D. Ribas, **G. Vallicrosa**, "Autonomous I-AUV docking for fixed-base manipulation" in *19th World Congress of the International Federation of Automatic Control (IFAC 2014). Published in IFAC Proceedings Volumes*, vol. 47 num. 3 pp. 12160-12165 ISSN 1474-6670 DOI 10.3182/20140824-6-ZA-1003.01878, Cape Town, August 2014.

- A. Palomer, P. Ridao, D. Ribas, A. Mallios, **G. Vallicrosa**, "A Comparison of G2o Graph SLAM and EKF Pose Based SLAM With Bathymetry Grids" in *9th IFAC Conference in Control Applications in Marine Systems (CAMS 2013)*. Published in IFAC Proceedings Volumes, vol. 46 num. 33 pp. 286-291 ISSN 1474-6670 DOI 10.3182/20130918-4-JP-3022.00065, Osaka, September 2013.

- A. Palomer, P. Ridao, D. Ribas, A. Mallios, N. Gracias, **G. Vallicrosa**, "Bathymetry-based SLAM with Difference of Normals point-cloud subsampling and probabilistic ICP registration" in *2013 MTS/IEEE OCEANS-Bergen*, pp. 1-8 ISSN 978-1-4799-0000-8 DOI 10.1109/OCEANS-Bergen.2013.6608091, Bergen, June 2013.

Acronyms

2D	two-dimensional
3D	three-dimensional
AHRS	Attitude and Heading Reference System
AL	Active Localization
AUV	Autonomous Underwater Vehicle
CIRS	Centre d'Investigació en Robòtica Submarina
COLA2	Component Oriented Layer-based Architecture for Autonomy
DR	Dead Reckoning
DS	Docking Station
DVL	Doppler Velocity Log
EKF	Extended Kalman Filter
FLS	Forward-Looking Sonar
FOV	Field of View
GIB	GPS Intelligent Buoy
GP	Gaussian Process
GPS	Global Positioning System
H-SLAM	Hilbert Maps SLAM
HM	Hilbert Map
I-AUV	Intervention AUV
INS	Inertial Navigation System
LBL	Long Baseline
MSIS	Mechanically Scanning Imaging Sonar
OG	Occupancy Grid
PF	Particle Filter
RBPF	Rao-Blackwellized Particle Filter
ROS	Robot Operating System
ROV	Remote Operated Vehicle
SAUC-E	Student Autonomous Underwater Challenge - Europe
SBL	Short Baseline
SGD	Stochastic Gradient Descent

SLAM	Simultaneous Localization and Mapping
SOG	Sum of Gaussian
SVS	Sound Velocity Sensor
TBN	Terrain-Based Navigation
TDOA	Time Difference of Arrival
ToF	Time of Flight
USBL	Ultra-Short Baseline
UVL	Underwater Vision Lab

List of Figures

List of Tables

Contents

Abstract

Autonomous Underwater Vehicles (AUVs) true autonomy capabilities in complex potentially unknown environments, have not yet been fully achieved because of the lack of online algorithms that can solve fundamental problems such as localization, mapping and path-planning on-board the AUV and consequently react according to their outputs. These algorithms can empower them for new capabilities such as long-term deployments, autonomous exploration and autonomous intervention.

This thesis presents the development of two online localization algorithms for AUVs. The first algorithm is based on a Sum of Gaussian (SOG) filter for online range-only localization of an acoustic beacon, e.g. localization of a Docking Station (DS) for battery recharging and data uploading. Two different versions of the algorithm are developed, one based on Dead Reckoning (DR) navigation and one based on a full Simultaneous Localization and Mapping (SLAM) solution. Moreover, an Active Localization (AL) algorithm is also developed to autonomously select the best actions that minimize the range-only localization uncertainty. This algorithm is also tested as part of a wider project where it is combined with other algorithms to produce a complete homing and docking strategy. Consequently, this algorithm can help long-term deployed AUVs being able to always return to their base DS for battery recharging.

The second algorithm proposes a new online SLAM framework for continuous occupancy mapping named H-SLAM. It uses a Rao-Blackwellized Particle Filter (RBPF) where each particle carries a HM representation of the environment. HMs offer a low memory footprint and constant computational complexity $O(1)$ for insertion and query, suitable for online processing. This algorithm is tested on two real-world datasets offering a significantly better reconstruction of the environment than using DR navigation. Producing correct continuous occupancy maps and trajectories, opens plenty of possibilities for future combination with online path-planning algorithms.

Resum

El verdader potencial d'autonomia dels Vehicles Submarins Autònoms (AUVs) en entorns complexos i potencialment desconeguts, encara no ha estat completament assolit degut a la falta d'algoritmes que puguin resoldre problemes fonamentals com la localització, la construcció de mapes i la planificació de trajectòries en el mateix vehicle i conseqüentment reaccionar d'acord amb aquests. Aquests algorismes poden capacitar els AUVs amb noves funcions com desplegaments de llarga durada, exploració autònoma i intervenció autònoma.

En aquesta tesi es presenta el desenvolupament de dos algoritmes de localització online per AUVs. Per una banda, el primer algorisme està basat en un filtre de Suma de Gaussians (SOG) per localitzar online una balisa acústica mitjançant mesures de rang, per exemple per localitzar una Estació d'Acoblament (DS) per recarregar bateries i transmetre dades. S'han desenvolupat dues versions diferents de l'algorisme, una basada en Navegació per Estimació (DR) i una basada en una completa Localització i Construcció de Mapes Simultània (SLAM). A més, s'ha desenvolupat un algoritme de Localització Activa (AL) per decidir autònomament quines són les millors accions que minimitzen la incertesa de la localització per rangs. Aquest algorisme s'ha testejat dins un projecte molt més ampli on s'ha hagut de combinar amb altres algoritmes per tal de desenvolupar una estratègia completa per tal de localitzar la DS i poder-s'hi acoblar. Com a conseqüència , aquest algoritme pot ajudar els desplegaments de llarga durada permetent sempre al vehicle de tornar a la DS per carregar bateries.

Per altra banda, el segon algorisme proposa un nou mètode de SLAM per la construcció de mapes continus d'ocupació anomenat Hilbert Maps SLAM (H-SLAM). Utilitza un Filtre de Partícules Rao-Blackwellizat (RBPF) on cada partícula té un Mapa de Hilbert (HM) com a representació de l'entorn. Els HMs ofereixen un baix ús de memòria i una complexitat computacional constant $O(1)$ per inserció i consulta, adequat per ser processats online. Aquest algorisme s'ha testejat amb dos conjunts de dades reals produint una millora significativa en la reconstrucció de l'entorn respecte la reconstrucció obtinguda utilitzant la navegació DR. Obtenir mapes continus d'ocupació i trajectòries correctes, obre moltes possibilitats per futures combinacions amb algorismes online de planificació de trajectòries.

Resumen

El verdadero potencial de autonomía de los Vehículos Submarinos Autónomos (AUVs) en entornos complejos y potencialmente desconocidos, aún no ha sido completamente logrado debido a la falta de algoritmos que puedan resolver problemas fundamentales como la localización, la construcción de mapas y la planificación de trayectorias en el vehículo mismo y consecuentemente reaccionar de acuerdo con estos. Estos algoritmos pueden capacitar los AUVs con nuevas funciones como despliegues de larga duración, exploración autónoma e intervención autónoma.

En esta tesis se presenta el desarrollo de dos algoritmos de localización online para AUVs. Por un lado, el primer algoritmo está basado en un filtro de Suma de Gausianos (SOG) para localizar online una baliza acústica mediante medidas de rango, por ejemplo para localizar una Estación de Acople (DS) para recargar baterías y transmitir datos. Se han desarrollado dos versiones diferentes del algoritmo, una basada en Navegación por Estimación (DR) y una basada en una completa Localización y Construcción de Mapas Simultáneamente (SLAM). Además, se ha desarrollado un algoritmo de Localización Activa (AL) para decidir autónomamente cuales son las mejores acciones que minimizan la incertidumbre de la localización por rangos. Estos algoritmos han sido probados dentro de un proyecto mas amplio donde se ha combinado con otros algoritmos para desarrollar una estrategia completa para localizar la DS y poderse acoplar. Como consecuencia, este algoritmo puede ayudar a los despliegues de larga duración permitiendo al vehículo poder volver siempre a la DS para cargar baterías.

Por otro lado, el segundo algoritmo propone un nuevo método de SLAM para la construcción de mapas continuos de ocupación llamado Hilbert Maps SLAM (H-SLAM). Utiliza un Filtro de Partículas Rao-Blackwellizado (RBPF) donde cada partícula tiene un Mapa de Hilbert (HM) como representación del entorno. Los HMs ofrecen un bajo uso de memoria y una complejidad computacional constante $O(1)$ para inserción y consulta, adecuado para ser procesados online. Este algoritmo se ha probado con dos conjuntos de datos reales produciendo una mejora significativa en la reconstrucción del entorno respecto a la reconstrucción obtenida utilizando la navegación DR. Obtener mapas continuos de ocupación y trayectorias correctas, abre muchas posibilidades para futuras combinaciones con algoritmos online de planificación de trayectorias.

1

Introduction

In this chapter we present the main problems that have motivated this thesis: online underwater localization for true autonomous Autonomous Underwater Vehicles (AUVs). The motivations behind this problem are introduced in Section 1.1, relating them to the requirements of real-world applications and the limitations of current approaches. Next, we state the objectives of the thesis in Section 1.2 and we briefly describe, in Section 1.3, the context in which this work has been carried out. Finally, the organization of the thesis document is presented in Section 1.4.

1.1 Motivation

Interest in underwater robotics research has increased over the last couple of decades, particularly for deep water. This is not surprising since the 71% of Earth surface is covered by oceans, which represent 96% of Earth's water. Although life started in the oceans 3.5 billion years ago and more than 1500 new species were discovered over the last decade, only 5% of the underwater domain can be considered explored [NOAA, 2018]. Deep water is an extremely hostile environment for humans due to high pressures. Early underwater exploration started with manned submersibles, and although they are still in use, the development of Remote Operated Vehicles (ROVs) increased the safety and duration of underwater operations. ROVs are routinely used in offshore industries and in science applications among others (e.g. offshore oil and industry, biology, archeology). However the need for a large and expensive support vessel to be able to handle the crane, the tether and the involved ROV crew has stimulated the research and development of AUVs.

Free from the physical connection to the surface ship, AUVs provide a stand-alone platform [Desa et al., 2006] that can gather data close to the seafloor without human supervision and avoiding the risks related to the umbilical cable. Those new possibilities also open new challenges on the engineering side, like better batteries [Reader et al., 2002] and sensors for longer and safer deployments. Moreover, the limited communications (e.g. acoustic modems) or lack of human supervision, pose as a huge challenge to computer science engineering. True autonomy involves several key enabling technologies such as autonomous localization and map building, autonomous path planning and re-planning, and autonomous intervention. Having solutions that can be run online in the AUVs will make them increasingly more capable and therefore more useful in underwater operations.

Localization underwater has always been a difficult problem, mainly due to the lack of global localization systems like Global Positioning Systems (GPSs). Inertial Navigation Systems (INSs) with depth measurements from a pressure-cell, velocity measurements from Doppler Velocity Log (DVL), and orientation from Attitude and Heading Reference System (AHRS), drift over time and become unreliable after long missions unless externally aided with absolute position fixes. The most commonly available fixes can come from Long Baseline (LBL), Short Baseline (SBL), Ultra-Short Baseline (USBL) and GPS Intelligent Buoys (GIBs) (Figure 1.1). However these systems require time for deployment and calibration, and constrain the AUV to their coverage area.

Simultaneous Localization and Mapping (SLAM), arises here as a key enabling technology required to develop other capabilities. Once a correct map is built and the AUV position within it is known, path planning and autonomous intervention, for instance, can be tackled with better chances of success. This thesis focuses on the research and development of online acoustic based localization algorithms for AUVs. By online we refer to

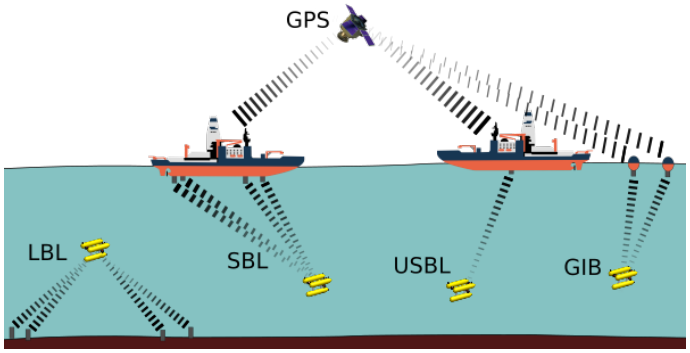


Figure 1.1: Types of acoustic absolute positioning fixes.

methods which have been designed to run in real-time, programmed to run on-board the AUV, accept sensor measurements as they arrive, and provide position and map estimates on time. This is in contrast to offline methods, which post-process data logged during a mission and not being required to meet time constrains.

When an online method is used to localize the AUV during a mission, we will refer to it as a real-time localization method. If the method is run online but over a dataset playback not during mission, it will be referred as online. Please note that the difference between online and real-time refers only to how the method has been validated. The main difference between online and real-time methods, is that the last ones have been tested enough to allow them to control the vehicle during an actual experiment.

This thesis addresses two methods. The first one works in a confined environment, the area of coverage of an acoustic beacon, and uses range-only information. The method was designed to support homing and docking operations using a Docking Station (DS) in the context of a remote launch/recovery scenario. The second one relieves the confinement constrains. It is an occupancy based SLAM which makes use of Hilbert Maps (HMs) to provide a continuous occupancy representation of the space at a very small memory footprint. The method is $O(1)$, so it is suitable for online implementation.

1.1.1 Range-only localization

Accurate target localization plays an important role in several application scenarios such as seismology, radar, mobile wireless communications, astronomy and sonar, to name a few. Underwater, methods based on visual detection such as optical cameras, are dependent on water visibility conditions and thus only work up to few meters. On the other hand, acoustic methods can be used at large distances regardless of water quality. There exist multiple methods of acoustic localization such as bearings-only localization and range-only localization. The first one uses an array of receivers to detect a Time Difference of Arrival (TDOA) that provides bearing information. That bearing information can be used on a moving vehicle to accurately locate the beacon source [Bishop et al., 2007, Hawkes and

Nehorai, 1998, Li and Nehorai, 2011, Moreno-Salinas et al., 2012]. However, such systems are not usually installed in AUVs and therefore cannot be used off-the-shelf. Alternatively, range-only measurements are easily measured on AUV's acoustic modems.

Range-only localization is a highly non-linear problem since given a measurement, the uncertainty in beacon localization has the shape of a ring in two-dimensional (2D) space and a spherical shell in three-dimensional (3D) space. This makes it impossible to solve directly by an Extended Kalman Filter (EKF) unless an initial guess is provided or delayed initializations are used. The EKF's single Gaussian covariance representation cannot represent this uncertainty shape.

For a first localization estimate, delayed initializations may include a least-squares approach [Newman and Leonard, 2003, Vaganay et al., 2000], a voting space similar to Hough transform [Olson et al., 2006], a pose-based EKF with delayed states [Webster et al., 2009], or a moving horizon estimation [Wang et al., 2013]. However, these methods are computed offline once the AUV is recovered and therefore cannot be used for autonomous homing and docking procedures.

[Blanco et al., 2008b] proposed a solution based on the Particle Filter (PF) to accurately represent the uncertainty of the beacon. This method was later improved in [Blanco et al., 2008a] with the introduction of a Sum of Gaussian (SOG) filter to cover more efficiently the space and being able to run it online to locate radio transceivers. Another solution based on the SOG filter with reduced sphere parametrization was proposed in [Fabresse et al., 2013].

To solve the range-only localization problem, it is necessary that measurements are gathered through a path that ensures observability. Observability is a key issue and several works have addressed it [Song, 1996, Song, 1999, Vaganay et al., 2000, Gadre and Stilwell, 2004, Gadre, 2007, Olson et al., 2006, Ross and Jouffroy, 2005]. Moreover, in [Salinas, 2013] there is an extensive research on best beacon positioning configurations to achieve minimum localization uncertainty. This is achieved through a maximization of the Fisher Information Matrix, also known as D-optimality or minimization of the uncertainty volume.

To have a truly autonomous localization method, an AUV must be able to decide which actions/movements help reduce the localization uncertainty. Algorithms related to Active Localization (AL) [Thrun et al., 2005, Stachniss, 2009, Vander Hook et al., 2014, Arsenio and Ribeiro, 1998] can help achieve this goal. Predicting how different actions affect the future uncertainty and evaluating it through some criteria [Carrillo et al., 2012], can provide autonomous decisions on the best actions to take.

1.1.2 Underwater SLAM

Multiple works have achieved successful SLAM implementations, either with optical imaging sensors or acoustic sonar sensors. Optical imagery has been used to construct 2D un-

derwater photo-mosaics that correct the inherent Dead Reckoning (DR) drift and enable an overview of extended areas of the seafloor [Eustice et al., 2002, Gracias et al., 2003, Ridao et al., 2010, Escartín et al., 2008, Singh et al., 2004, Bingham et al., 2010, Elibol et al., 2011, Eustice et al., 2005]. Additionally, in scenarios with a high 3D component, optical imagery has also been used for 3D reconstructions [Williams and Mahon, 2004, Pizarro et al., 2004, Nicosevici et al., 2009, Zhang and Negahdaripour, 2010, Johnson-Roberson et al., 2010].

On the other hand, among sonar sensors, Forward-Looking Sonar (FLS) provides a strong alternative to optical imagery mosaicking in low visibility conditions [Kim et al., 2005, Negahdaripour et al., 2005, Aykin and Negahdaripour, 2013, Hurtós et al., 2015]. Although FLS provides a longer measurement range, its Field of View (FOV) is limited and the change of orientation greatly affects the perceived appearance of measured objects.

Multibeam echosounders are commonly used to obtain 2.5D elevation maps of the seafloor thanks to their wide swath and long range of measurements. Typically used on surface ships to map the seafloor, they are also used in AUVs to obtain a better resolution closer to the bottom [Roman and Singh, 2005, Barkby et al., 2011, Barkby et al., 2012, Palomer et al., 2016].

Finally, mechanical scanning sonars and single beam echosounders have also been used for SLAM in man-made environments with line features [Ribas et al., 2008]. Even in fully 3D environments like caves, with occupancy grids [Fairfield et al., 2007], as well as with scan-matching algorithms [Mallios et al., 2014].

SLAM underwater is usually computed offline after the AUV is recovered from water and its data downloaded. After observing the obtained result, another mission can be scheduled to explore potential targets or cover the gaps of the first mission. This process can be inefficient and costly. However having the SLAM solution in real-time, could enable autonomous exploration [Hernández et al., 2017] or autonomous intervention [Palomeras et al., 2016] capabilities for the AUVs. Moreover, if the map solution is an occupancy map, it can also help the development of online path-planning algorithms.

To the best of the authors knowledge the only underwater SLAM algorithms that have been tested in real-time, are [Fairfield et al., 2007, Hurtós et al., 2015]. The first uses multiple single beam echosounders and provides an Occupancy Grid (OG) map using an efficient Deferred-Reference Octree representation to avoid huge copies in its PF. While the second one uses a FLS Fourier-based registration with a pose-graph representation with loop-closing detection.

1.2 Objectives

Once motivations have been described, we state the main goal of this thesis as:

"To research and develop advanced online acoustic based localization methods for AUVs. Two distinct methods are proposed. The first one localizes an AUV and a DS in the context of a remote launch/recovery application. The second one localizes an AUV using a continuous occupancy map."

This general goal can be broken down to the following more specific objectives:

Online localization method to localize a DS: develop an online localization algorithm to support the homing operation of an AUV towards its DS.

- **Range-only localization:** implement an online localization algorithm capable of localizing an unknown beacon position (DS) with range measurements and the on-board DR navigation.
- **Extension to Active Localization:** extend the algorithm to be capable of autonomously deciding which movements help achieve better and faster localization.
- **Extension to SLAM:** extend the algorithm to simultaneously estimate the position of the beacon and the AUV.

Online Occupancy based SLAM: research and develop an online acoustic based SLAM method, using an occupancy-based representation of the environment, easy to integrate in the future with online path-planning algorithms.

- **Efficient Occupancy Based Representation of the Environment:** adopt an occupancy mapping representation capable of running online in a PF-SLAM where each particle carries its own copy of the map. This representation needs to have a low memory usage and low computational complexity to allow fast queries and insertions.
- **Online occupancy SLAM:** develop a new online SLAM framework capable of localizing the AUV and building an occupancy map.

Experimental validation: Demonstrate the capabilities of the developed methods on real AUVs.

1.3 Context

The research of this thesis has been conducted at the Underwater Robotics Research Center, *Centre d'Investigació en Robòtica Submarina (CIRS)*, of the Computer Vision and

Robotics (ViCOROB) Institute of Universitat de Girona (UdG). Research in underwater robotics has been ongoing there since 1992, supported by several National and European research programs. The group has developed several AUV prototypes such as: GARBI [Amat et al., 1999] originally conceived as a ROV that was later redesigned as an AUV, URIS [Batlle et al., 2005] a very lightweight AUV, Ictineu [Ribas et al., 2007] that won the first Student Autonomous Underwater Challenge - Europe (SAUC-E) competition in 2006, Sparus that also won SAUC-E in 2010, Girona500 [Ribas et al., 2012] a reconfigurable AUV that has been used in the experimental part of this thesis, and Sparus II [Carreras et al., 2013, Carreras et al., 2018] a restyled version of the original Sparus presented as a commercial platform and winner of the multi-domain competition Eurathlon 2014, 2015 and 2017, that is also used in the experimental part of this thesis.

Research at CIRS revolves around AUV applications: control architectures [Palomeras et al., 2012], autonomous intervention [Carrera et al., 2015, Palomeras et al., 2014, Youakim et al., 2017, Cieslak et al., 2015, Palomer et al., 2018], SLAM [Ribas et al., 2008, Mallios et al., 2014, Palomer et al., 2016], and path planning [Hernández et al., 2011, Galceran Yebenes, 2014, Hernández et al., 2016].

There also exists a close collaboration between CIRS and the Underwater Vision Lab (UVL), also from VICOROB Institute. UVL research revolves around underwater computer vision, covering topics such as 3D reconstruction [Istenic et al., 2017], underwater panoramic image stitching [Bosch et al., 2015], image dehazing [Ancuti et al., 2017], multi-vehicle mapping [Campos et al., 2016] and fish detection [Prados et al., 2017].

Joint work between the two labs has lead to successful results in real-world applications such as dam inspection [Ridao et al., 2010], AUV mapping of archaeological sites [Gracias et al., 2013], and AUV mapping and intervention in a harbor scenario [Prats et al., 2012].

1.3.1 Participation in projects

The work presented in this thesis has contributed to the following projects in which CIRS has participated:

- MINECO Project 3DAUV: Automatic inspection of 3D underwater environments using an AUV (ref. DPI2015-73978-JIN), funded by the Spanish Ministry of Science and Innovation.

- EU FP7 Project LOON-DOCK/SUNRISE: Sensing, monitoring and actuating in the UNderwater world through a federated Research InfraStructure Extending the Future Internet (ref. FP7-ICT-2013-10-611449), funded by the European Commission.

- MINECO Project ARCHROV (part of MERBOTS): marine ARChaeology through

HROV/AUV cooperation (ref. DPI2014-57746-C3-3-R), funded by the Spanish Ministry of Science and Innovation.

- MINECO Project COMAROB (part of TRITON): Robótica cooperativa Marina para el mapeo acústico y la intervención (ref. DPI2011-27977-C03-02), funded by the Spanish Ministry of Science and Innovation.
- EU FP7 Project EUROFLEETS2: New Operational Steps towards an Alliance of European Research Fleets (ref. FP7-INFRASTRUCTURES-2012-1-312762), funded by the European Commission.
- EU FP7 Project MORPH: Marine robotic system of self-organizing, logically linked physical nodes (ref. FP7-ICT-2011-7-288704), funded by the European Commission.

1.3.2 Research collaborations

Throughout this thesis, various collaborations have taken place with other researchers from VICOROB and from abroad. The study of acoustic modems and their communications allowed for collaboration in [Centelles et al., 2017] where compressed visual feedback was transmitted through the acoustic modem. Moreover, acoustic range measurements were used to estimate the position of an AUV to accurately position a surface vehicle to improve the AUV localization in [Mandić et al., 2016]. In addition, early studies on online mapping methods helped to develop the work related to online path planning algorithms [Hernández et al., 2015a, Hernández et al., 2015b, Hernández et al., 2015].

Finally, this thesis has also benefited from a research stay of three months in the Marine Systems group at the Australian Centre for Field Robotics (ACFR) of University of Sydney (Sydney, Australia). ACFR is a science and engineering research center distinguished for their work in habitat mapping and SLAM [Barkby et al., 2011, Barkby et al., 2012, Johnson-Roberson et al., 2010, Williams and Mahon, 2004, Williams et al., 2010]. There, the author had the opportunity to join a 10-day mapping cruise with Sirius AUV.

1.4 Document Structure

The following chapters of this thesis are structured as follows (see Figure 1.2):

- **Chapter 2. Background.** Provides an explanation of the different sensors and AUVs that are used throughout the experimental part of this thesis. A detailed explanation of HM representation for continuous occupancy mapping is also provided.
- **Chapter 3. SOG Single Beacon Range-Only Localization for AUV Homing.** A new approach to range-only localization for homing based on the SOG filter is proposed. Besides providing correct localization of the target, this approach also

implements an AL method to autonomously minimize the uncertainty of localization. The proposed approach is tested in a real harbor scenario.

- **Chapter 4. AUV homing and docking for remote operations.** The range-only for homing approach of Chapter 3 is used as part of a more complex autonomous homing and docking strategy to demonstrate permanent deployment in the context of LOON-DOCK/SUNRISE European project.

- **Chapter 5. H-SLAM: Particle Filter SLAM Using Hilbert Maps.** A new SLAM framework, named Hilbert Maps SLAM (H-SLAM), for online continuous occupancy mapping with sonar sensors is proposed. The combination of a Rao-Blackwellized Particle Filter (RBPF) with a HM representation in each particle, provides a continuous occupancy map that correctly represents the environment. The proposed approach is tested in several 2D datasets.

- **Chapter 6. Results & Discussion.** A comprehensive discussion of the obtained results is provided in this chapter.

- **Chapter 7. Conclusions & Future Work.** Concludes this thesis by summarizing the main contributions and the future research directions.

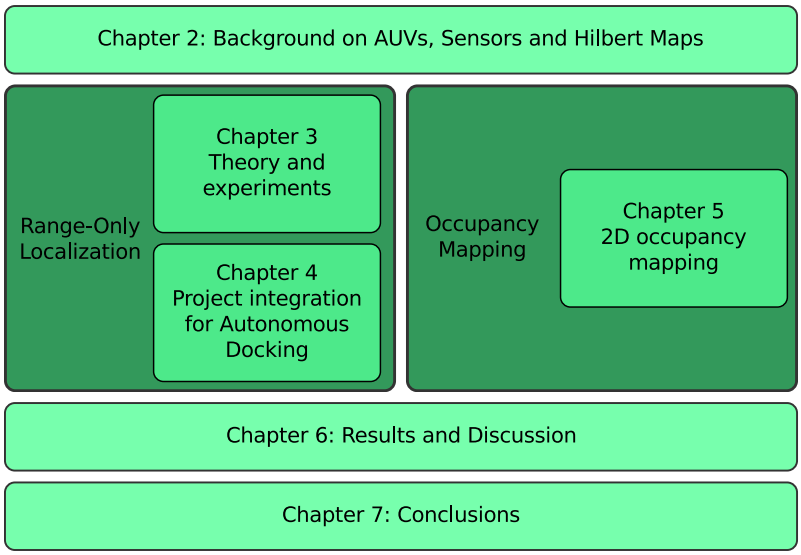


Figure 1.2: Document structure.

2

Background

Before moving to the compendium of journal publications that this thesis is composed of, it is essential to understand the hardware and the algorithms used in them. In this chapter we first describe the different sensors used by the AUVs in Section 2.1 and then in Section 2.2, the AUVs that were used in the experimental part of this thesis. Finally, we also describe in detail (Section 2.3) the HM representation that is used in Chapter 5 for map representation. Finally, a summary in Section 2.4 links the background of this chapter to the different journals in this compendium.

2.1 Underwater sensors

Nowadays AUVs can accommodate a wide range of different sensors from simple pressure cells to mass spectrometers [Short et al., 1999]. The most basic sensors that AUVs have, are the ones providing navigation. The most common of those navigation sensors are the following:

GPS provides accurate positioning, but it only works on surface. It is used mainly to correctly geolocalize the start of a mission and to evaluate the total drift of the AUV once it resurfaces. Combined with SBL, USBL or GIBs can provide absolute fixes underwater that bound the DR navigation drift.

Pressure sensor provides depth measurements. It can be directly measured by a pressure cell or, alternatively, integrated in a more complex one such as the Sound Velocity Sensor (SVS) (Figure 2.1). Additionally to depth measurements, SVS also measures sound velocity and temperature. Sound velocity measures are required by sound related measurements, e.g. acoustic ranges, sonars and DVLs.

Figure 2.1: Example of Sound Velocity Sensor (SVS) from Valeport®.

DVL provides linear velocities in all three axis (Figure 2.2). Measurements are obtained thanks to the Doppler shift effect from different acoustic beams. Velocities are provided relative to water and/or seabed. Moreover, their capacity to analyze the quality of the measurements, allows outliers to be discarded. It also provides measurements of altitude, that aid to maintain a distance to the seafloor.

Figure 2.2: Example of Doppler Velocity Log (DVL) from Teledyne®.

AHRS provides orientation and angular rate measurements. It is an essential sensor to know the orientation of the vehicle and therefore where it is moving when coupled with a DVL.

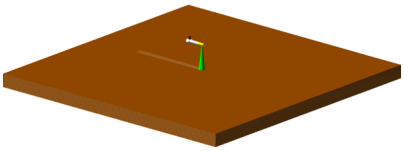
Apart from the basic navigation sensors, exteroceptive sensors such as acoustic modems and different types of sonars are also used in this thesis:

Acoustic modems (Figure 2.3) are mainly designed to transmit data at low rates. This allows operators to control the status of the AUV and to send simple commands. They can also behave as acoustic transponders and combined with LBL, SBL, USBL or GIB can provide accurate positioning that corrects the drift of the DR navigation. However, acoustic transponder networks are costly to deploy and calibrate, and/or require a surface ship.

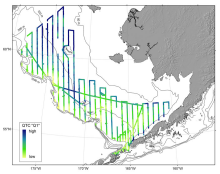
Figure 2.3: Example of acoustic modem from Evologics®.

In this thesis, acoustic modems are used as range sensors. Knowing the sound velocity and measuring the Time of Flight (ToF) of any data transmission between modems, this range can easily be computed. Moreover, the target can also send useful information such as its own depth and orientation to ease the localization process (Chapter 4).

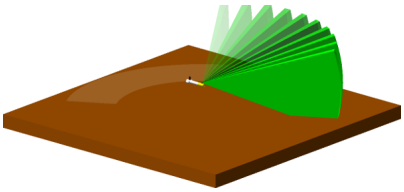
Sonar sensors (Figure 2.4) are used in many underwater operations that have long relied on sonar technology because acoustic waves are significantly less affected by attenuation than optical sensors such as cameras. Sonars that can deliver range measurements, such as single beam sonars (Figure 2.4a), profiler sonars (Figure 2.4e) or multibeam sonars (Figure 2.4g), have been used for obstacle avoidance, navigation, localization and mapping [Roman and Singh, 2005, Fairfield et al., 2007, Kinsey et al., 2006]. Multibeam sonar have been used extensively to create seafloor bathymetric charts. Imaging sonars, such as Mechanically Scanning Imaging Sonar (MSIS) (Figure 2.4c), sidescan sonars (Figure 2.4i) or FLS (Figure 2.4k) have also been widely used for obstacle-avoidance, localization and mapping applications [Tena et al., 2003, Mallios et al., 2014, Ribas et al., 2008, Aulinas et al., 2010, Baumgartner and Wales, 2006, Chen et al., 2011, Walter et al., 2008].



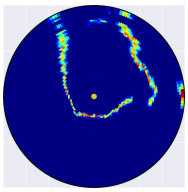
(a) Single beam sonar.



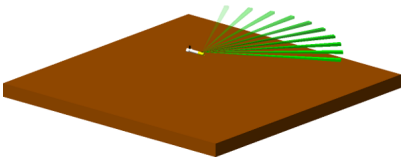
(b) Map from single beam sonar.



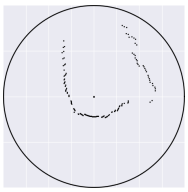
(c) MSIS.



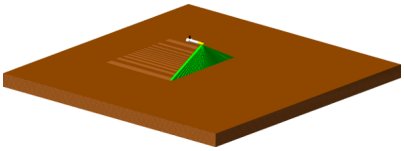
(d) 360° scan from MSIS.



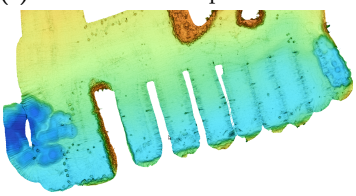
(e) Profiler sonar.



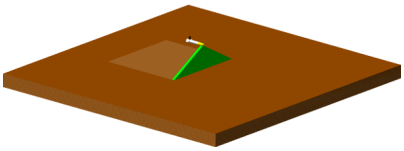
(f) 360° scan from profiler sonar.



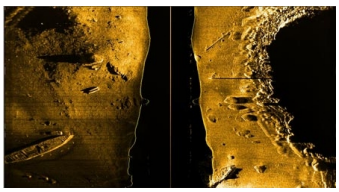
(g) Multibeam sonar.



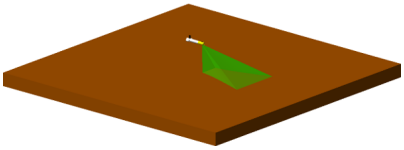
(h) Bathymetry map from multibeam data.



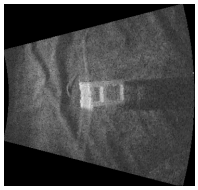
(i) Sidescan sonar.



(j) Waterfall view of sidescan data.



(k) FLS.



(l) 2D image from FLS.

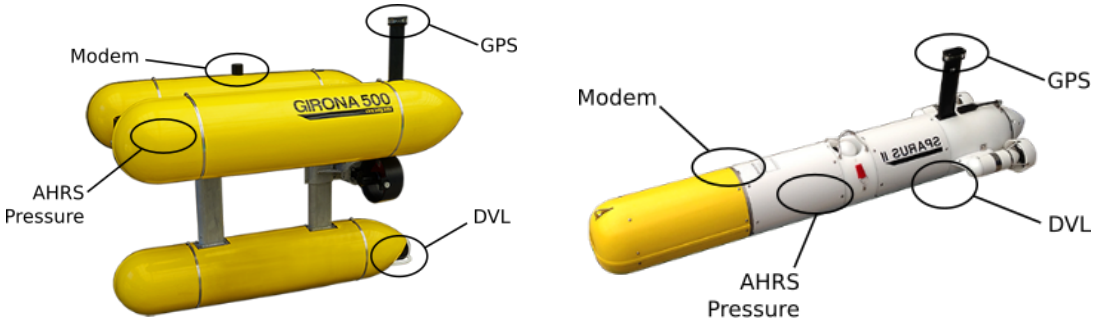
Figure 2.4: Sonar sensor types and example data.

2.2 Autonomous Underwater Vehicles (AUVs)

Two different AUVs were used in the experimental part of this thesis: Girona500 and Sparus II. Both of them were developed in-house at CIRS (Figure 2.5).

Girona500 AUV (Figure 2.5a) has dimensions 1 m × 1 m × 1.5 m, weights less than 200 kg and is rated for 500 m depth. It has a passive stability in pitch and roll due to the center of buoyancy being higher than the center of gravity. It is controlled by 5 thrusters that allow control on in heave, surge, sway and yaw. There is a big payload area in the lower hull, that can accept many kinds mission-specific sensors and actuators. It can even handle an arm, converting it to an Intervention AUV (I-AUV) [Ridao et al., 2015].

Sparus II AUV (Figure 2.5b) is a light torpedo-shaped hovering vehicle of length 1.6 m by 230 $mm\varnothing$, weights less than 60 Kg and is rated for 200 m depth. It is controlled by 3 thrusters that allow control in heave, surge and yaw. There is a payload area in the frontal part, that can accept many kinds mission-specific sensors and actuators.



(a) Girona500 AUV.

(b) Sparus II AUV.

Figure 2.5: Girona500 & Sparus II AUVs with navigation sensors and acoustic modem locations.

Both vehicles are equipped with a navigation sensor suite that includes a pressure sensor, a DVL, an AHRS and a GPS to receive position fixes while at surface. They also include an acoustic modem for underwater communication with other vehicles or surface stations (e.g. by using an USBL system), and a Wi-Fi antenna that can be used when the AUV is at surface. Moreover, the payload area makes these vehicles easily configurable according to the mission's requirements, and may include optical cameras, single-beam echosounders, MSIS, profiler sonar or multibeam sonar, to name a few.

Both AUVs pack an Intel i7 computer for internal processing. This computer runs the Component Oriented Layer-based Architecture for Autonomy (COLA2) software architecture [Palomeras et al., 2012] that is built on top of Robot Operating System (ROS) [Quigley et al., 2009]. Besides operating real robots, the software architecture is the same when operating in simulation. This is extremely useful to develop new capabilities since same code can be tested on simulation and on the real AUVs without changes. In this thesis newly developed algorithms request information to the software architecture such as the current DR navigation or the obtained sensor measurements. Small modifications to the acoustic

modem driver were made to be able to obtain range measurements when communicating with other modems.

2.3 Occupancy Representation

Occupancy representation has been historically done with Occupancy Grids (OGs) [Moravec, 1988] where space is discretized into square grid cells in 2D and cube voxels in 3D. The extent of the map has to be known beforehand, and all cells are kept in memory. Instead of saving the occupancy probabilities directly, log-odds representation (Section 2.3.1) is often used for better numerical stability [Thrun et al., 2005]. To simplify data insertion, cells are usually assumed to be independent from each other. Insertion and queries in OGs have a constant computational complexity $O(1)$ since they only require access to a table entry.

Most of real-world maps are not observed in all their extent and therefore a lot of memory is wasted storing unobserved regions of the map. To overcome this drawback, the same space can be represented by quadtrees in 2D or octrees in 3D. These tree structures only keep observed cells in memory and therefore there is a significant memory reduction. This reduction comes at the cost of incrementing the computational complexity for queries and insertion to logarithmic $O(\log(n))$. One well known example of such octree structures is the Octomap library [Hornung et al., 2013].

The main problem with cell independence assumption is that measurements do not affect neighboring cells. For example, when a single cell is unobserved but all surrounding cells are observed to be free multiple times, it could be assumed that the cell is also free up to some confidence degree. To overcome this problem, Gaussian Processs (GPs) offer a continuous occupancy representation that can be queried for occupancy but also for uncertainty [O'Callaghan et al., 2009]. The major drawback is that they significantly increase the computational complexity to $O(z^3)$ where z is the number of point observations used to learn the GP. Because point measurements are needed to learn a GP, range measurements must be randomly sampled to obtain free points. Improvements like accepting line measurements to reduce the number of observations [O'Callaghan and Ramos, 2012] and the use of local maps [Kim and Kim, 2013] managed to reduce computation time, but still remains too complex for online applications.

Hilbert Maps (HMs) [Ramos and Ott, 2016] were proposed to reduce this computational complexity to constant $O(1)$ and still maintain the continuous representation and reduction of memory usage with respect to OGs. HMs are an approximation, in a finite feature space, to the infinite space that GPs represent. However, in this approximation, uncertainty of occupancy cannot be computed. HMs are used throughout Chapter 5 for map representation, and a more detailed explanation is available in Section 2.3.2.

The main differences between the aforementioned representations can be observed in

Table 2.1 and Figure 2.6.

Table 2.1: Comparison of occupancy representations where r is the cell resolution, n is the number of cells in a full OG, z is the number of observations, and m is the number of features (where $m \ll n$).

	OG	Octrees	GP	HM
Independence	yes	yes	no	no
Uncertainty	no	no	yes	no
Resolution	r	r	∞	∞
Memory	n	$< n$	z^2	m
Complexity	$O(1)$	$O(\log(n))$	$O(z^3)$	$O(1)$

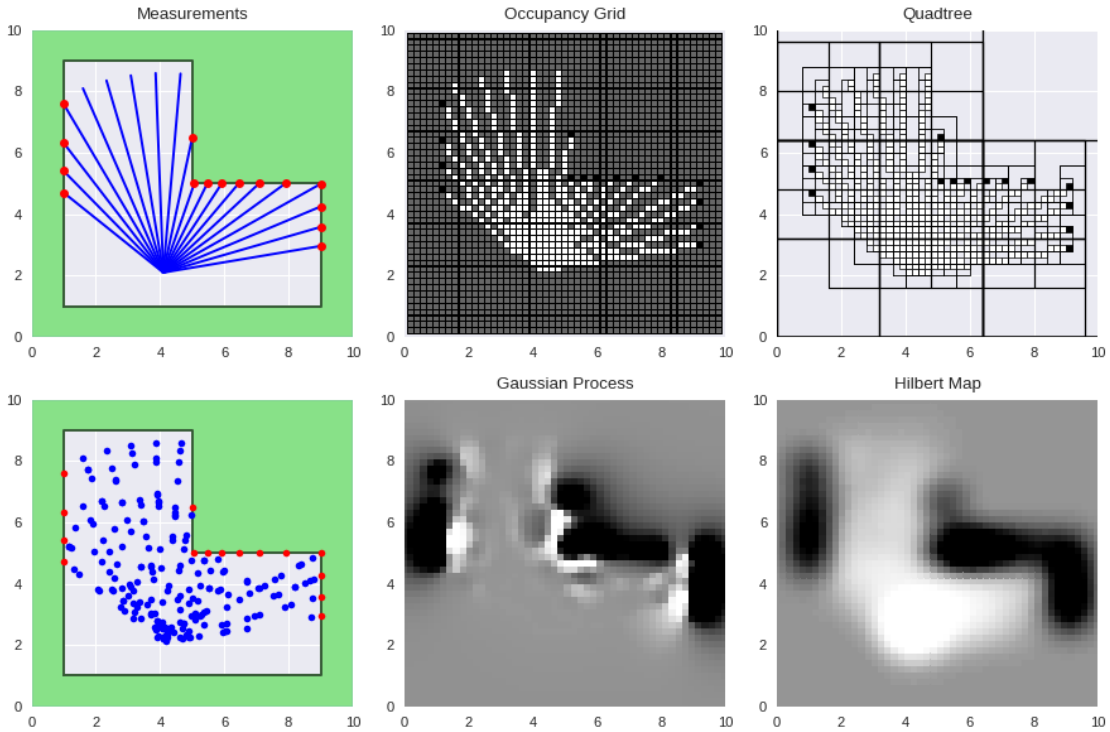


Figure 2.6: Comparison of different occupancy representations. *(top-left)* Environment with green space as occupied space, range measurements in blue and occupied points in red. *(bottom-left)* Environment with green space as occupied space, free randomly sampled points in blue and occupied points in red. *(top-center)* Range measurements saved in an OG (2500 cells at 0.2 m resolution). *(top-right)* Range measurements saved in a quadtree (690 cells at 0.2 m resolution). *(bottom-center)* Free and occupied points saved in a GP (112 point observations queried at 0.2 m resolution). *(bottom-right)* Free and occupied points saved in a HM (features of dimension 484 queried at 0.2 m resolution).

2.3.1 Log-odds Probability Representation

In occupancy mapping algorithms it is common to use the log-odds representation of the probability. The odds of a probability p is defined as the ratio between p and its complementary $(1-p)$ as

$$\text{odds}(p) = \frac{p}{1-p}. \tag{2.1}$$

The log-odds is given by the natural logarithm of the odds as

$$l = \text{log-odds}(p) = \ln\left(\frac{p}{1-p}\right) \tag{2.2}$$

where the log-odds(p) domain is $[0, 1]$ and its range is infinite $[-\infty, +\infty]$ (Figure 2.7).

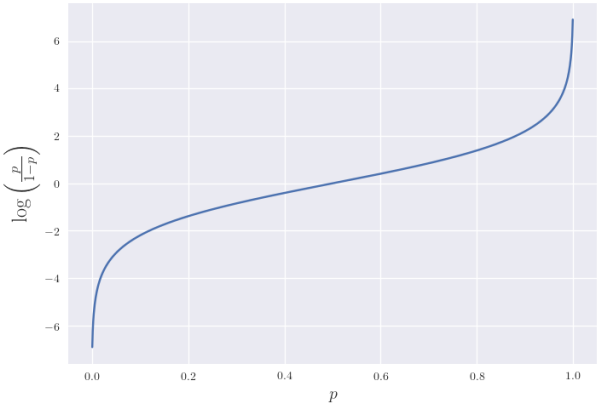


Figure 2.7: Log-odds of the probability.

Given a log-odds probability l, the corresponding probability can be recovered as

$$p = \frac{1}{1+e^{-l}} \tag{2.3}$$

Log-odds is a computationally elegant representation of probabilities since it avoids numerical problems at probabilities close to 0 or 1, and because it transforms equations based on the product of probabilities (e.g. the Bayes Rule) into equations based on additions of log-odds due to its logarithmic nature. Nowadays most OGs algorithms, use this representation.

2.3.2 Hilbert Maps

2.3.2.1 Notation

Let $\mathbf{x}_i$ be a point in the space $\mathbf{x}_i \in \mathbb{R}^D$, where D is the space dimension, and let y_i be a categorical variable representing the occupancy property of $\mathbf{x}_i$, $y_i \in \{-1, +1\}$ where -1 means *free* and $+1$ means *occupied*. Then $\mathcal{D}$, is a dataset $\mathcal{D} = \{\mathbf{x}_i, y_i\}_{i=1}^N$ of observed points and their categories.

The dataset $\mathcal{D}$, is obtained while the robot moves across the environment. Figure 2.6-*top-left* shows a possible example of a dataset captured from a single robot position where *free* space rays are represented in blue and *occupied* points are represented in red. Figure 2.6-*bottom-left* shows the points dataset $\mathbf{x}_i$ that is composed by the set of *occupied* red points and *free* randomly sampled points from the *free* rays.

2.3.2.2 Map model

The occupancy map is built by learning a discriminative model $p(y|\mathbf{x}, \mathbf{w})$, parametrized by a vector $\mathbf{w}$, to predict the occupancy property of new query points $\mathbf{x}_*$. The parameters $\mathbf{w}$ are learned through a logistic regression classifier using Stochastic Gradient Descent (SGD). Since a linear classifier would not be able to correctly classify the complex physical world into free and occupied categories, points $\mathbf{x}$ are first projected as features $\Phi(\mathbf{x})$ of a higher dimensional Hilbert Space where the classifier learns the linear model $p(y|\Phi(\mathbf{x}), \mathbf{w})$ on the projected features.

Given any training point $\mathbf{x}_i$, it can be mapped to a feature $\Phi(\mathbf{x}_i)$ through the feature mapping function

$$\Phi(\mathbf{x}_i) = [\Phi_1(\mathbf{x}_i)\ \Phi_2(\mathbf{x}_i)\ \ldots\ \Phi_M(\mathbf{x}_i)]^T, \tag{2.4}$$

where $\Phi_j(\mathbf{x}_i) : \mathcal{X} \to \mathbb{R}$.

The inner product of these features approximates popular kernels used in kernel machines for non-linear classification $k(\mathbf{x}, \mathbf{x}') \approx \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle$. This approximation allows the use of a linear classifier operating in the Hilbert Space instead of costly non-linear classifiers operating in the Euclidean Space.

2.3.2.3 Map Query

Once the model $p(y|\Phi(\mathbf{x}), \mathbf{w})$ is learned, the occupancy y_* of any query point $\mathbf{x}_* \in \mathcal{R}^D$ can be tested as *occupied* evaluating

$$p(y_* = +1|\Phi(\mathbf{x}_*), \mathbf{w}) = \frac{1}{1 + \exp\left(-\mathbf{w}^T \Phi(\mathbf{x}_*)\right)}. \tag{2.5}$$

which predicts the value in the Hilbert Space and then transforms it from log-odds to probability. In the same way, the probability of $free$ can be computed as

$$p(y_* = -1|\Phi(\mathbf{x}_*), \mathbf{w}) = 1 - p(y_* = +1|\Phi(\mathbf{x}_*), \mathbf{w}). \tag{2.6}$$

2.3.2.4 Learning

The learning of the parameters (or weights) vector $\mathbf{w}$, is done by minimizing the negative-log-likelihood (NLL) using SGD

$$\mathbf{w}_t = \mathbf{w}_{t-1} - \eta_t \frac{\partial}{\partial \mathbf{w}} NLL(\mathbf{w}), \tag{2.7}$$

where

$$NLL(\mathbf{w}) = \sum_{i=1}^{N} -\log(p(y_i|\Phi(\mathbf{x}_i), \mathbf{w})) + R(\mathbf{w}), \tag{2.8}$$

η_t is the learning rate, and $R(\mathbf{w})$ is the elastic net regularizer to prevent over-fitting and enforce sparseness.

2.3.2.5 Features

There exist many different types of feature mappings $\Phi(\cdot)$, from Fourier features [Ramos and Ott, 2016], to wall-like features [Guizilini and Ramos, 2017] and Gaussian features [Guizilini and Ramos, 2016]. In this thesis, a sparse triangle feature is used. Inducing points f_j * of the feature mapping are placed in a regular grid at f_{res} distance and affect a neighborhood defined by the radius r_{th}. The feature mapping is defined as a function of the inducing points as

$$\Phi_j(\mathbf{x}) = \begin{cases} \frac{r_{th} - ||\mathrm{f_j} - \mathbf{x}||_2}{r_{th}} & \text{if } ||\mathrm{f_j} - \mathbf{x}||_2 < r_{th} \\ 0 & \text{otherwise} \end{cases}. \tag{2.9}$$

A more visual explanation can be found in Figure 2.8 that depicts the feature $\Phi(\mathbf{x}_*)$.

2.3.2.6 Learning maps from range measurements

Section 2.3.2.4 explains how a HM can be learned from labeled points through SGD. However, in real world applications, measurements are usually obtained as ranges. Those ranges need to be discretized into points in order to be learned by the HM.

First, the HM is initialized with all weights equal to zero ($\mathbf{w} = \vec{0}$). This defines a completely unknown map, since any prediction as occupied (2.5) or free (2.6) will result in a 0.5 probability.

*Notice the change of notation from the one used in Chapter 5. Here subindex j is used to better differentiate between the point index i and the inducing point index j.

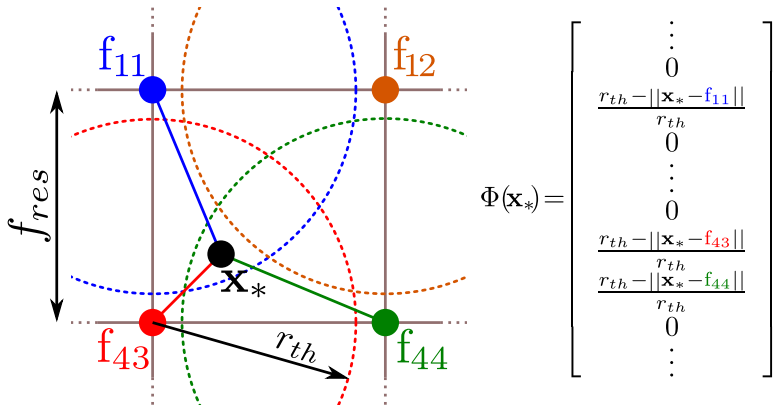


Figure 2.8: Schematic of HM triangle feature mapping. Each inducing point f_j is located in a regular grid at distance f_{res} and affects the neighborhood defined by the radius r_{th}. When predicting the occupancy of a point $\mathbf{x}_*$, it must be first projected to a feature $\Phi(\mathbf{x}_*)$ and pre-multiplied by the weights $\mathbf{w}$ according to (2.5). In the example shown, the query point is inside the neighborhoods of $(\mathrm{f}_{11}, \mathrm{f}_{43}, \mathrm{f}_{44})$ and all other feature values are zero.

For each range measurement, the endpoint of the range is labeled free if the range is maximum (no obstacle was found in its path) and occupied otherwise. Then, the rest of the ray (from sensor position to measured range) is sampled randomly and labeled free to properly cover the ray (Figure 2.9). Those points and labels are the ones learned into the HM. When the vehicle advances and obtains new measurements, those are also

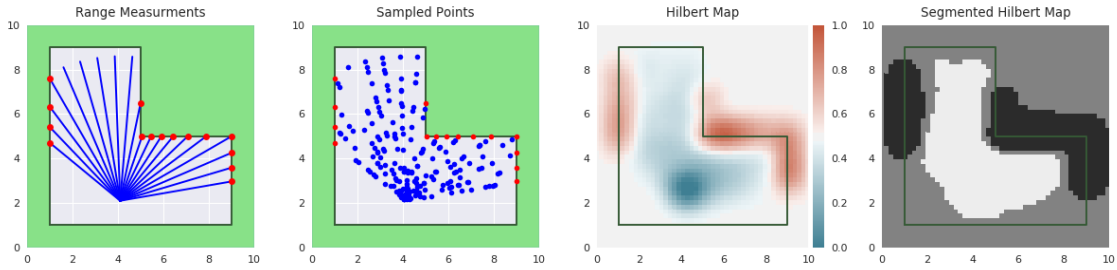


Figure 2.9: Example of HM learning from range measurements. (**1**) Range measurements obtained by a mobile vehicle in an scenario. Blue rays represent the measured free space and red dots represent hit-points or occupied points. (**2**) Ray measurements sampled randomly to obtain points. (**3**) Occupancy $p(y = 1)$ of the learned HM. (**4**) Segmented occupancy of the HM where $p(y = 1) < 0.5$ is *free* (white), $p(y = 1) > 0.5$ is *occupied* (black) and *unknown* (gray) otherwise.

incorporated into the HM, obtaining more confidence in the re-observed regions (see the second row in Figure 2.10).

2.3.2.7 Raycasting

When working with occupancy maps for localization it is necessary to be able to compare real measurements with expected measurements. The last ones are usually obtained by raycasting methods in grid-like representations [Amanatides et al., 1987]. For HM, we

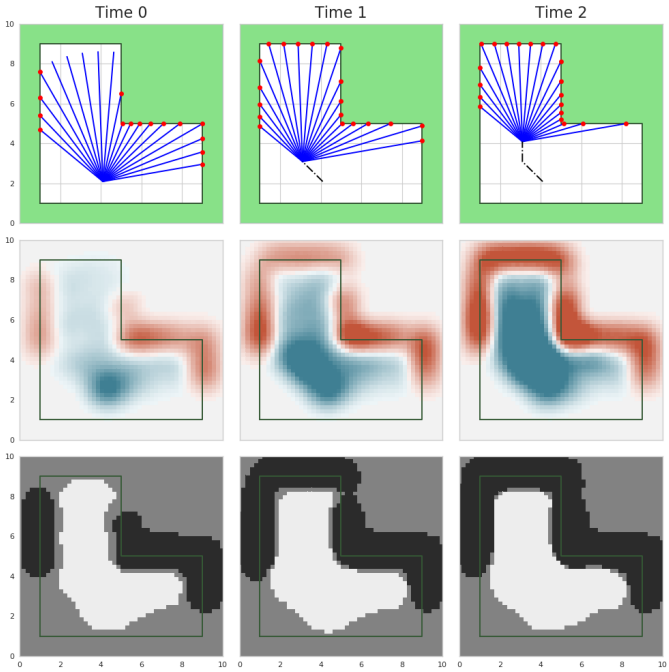


Figure 2.10: Example of HM learning while the vehicle advances in the scenario. First row shows the obtained range measurements where the second and third rows show the occupancy $p(y = 1)$ of the learned HM and its segmentation in {*free, occupied, unknown*} (refer to Figure 2.9 for occupancy values).

adopt a similar strategy to the one developed for GP occupancy maps [Hata et al., 2016]. Having a ray with origin and orientation that we want to know its range, we keep checking the probability of each point following the ray orientation at a certain resolution s. Once the predicted occupancy of that advancing point is bigger than a specified threshold th_p, the process is stopped and the point is considered a hit. To obtain the exact point where the threshold th_p was crossed, a linear interpolation between the hit point and the previous point is computed. If no hit point is found before reaching a maximum specified range r_{max} the raycast returns a free measurement. For a better explanation check Algorithm 2.3.1.

2.4 Summary

This chapter has explained the different sensors, AUVs and map models that are used throughout this thesis.

In Chapter 3, Girona500 AUV is used to localize an acoustic beacon through acoustic range measurements. There, the DR navigation solution provided by the robot's software architecture is compared against the SLAM navigation solution which uses raw sensor data from AHRS, DVL and depth sensors.

In Chapter 4, Sparus II AUV is used to also localize an acoustic beacon (a DS) through acoustic range measurements. The DR navigation is used in all the experiments. As a

Algorithm 2.3.1: HM raycasting.

Data: A ray origin $\mathbf{x}_o$, ray unit vector $\mathbf{u}$, occupancy probability threshold th_p, maximum range r_{max}, and query resolution s.
Result: A raycasted range r and its corresponding label y.

$\mathbf{x}_e \leftarrow \mathbf{x}_o$
$\mathbf{x}_{prev} \leftarrow \mathbf{x}_o$
$p \leftarrow$`Predict(`$\mathbf{x}_e$`)`
$p_{prev} \leftarrow p$
while $p < th_p$ **and** $||\mathbf{x}_e - \mathbf{x}_o|| < r_{max}$ **do**
  $\mathbf{x}_{prev} \leftarrow \mathbf{x}_e$
  $occ_{prev} \leftarrow p$
  $\mathbf{x}_e \leftarrow \mathbf{x}_e + s \cdot \mathbf{u}$
  $p \leftarrow$`Predict(`$\mathbf{x}_e$`)`
end
if $p < th_p$ **then**
  $r \leftarrow r_{max}$
  $y \leftarrow -1$
else
  $\mathbf{x}_e \leftarrow$`Interpolate(`$\mathbf{x}_{prev}, \mathbf{x}_e, p_{prev}, p, th_p$`)`
  $r \leftarrow ||\mathbf{x}_e - \mathbf{x}_o||$
  $y \leftarrow +1$
end

difference with respect to the previous chapter, in this case the range-only localization is done as a part of a wider project to demonstrate persistent autonomy for AUVs. When docking is required, the mission manager requests a range-only localization that serves as an accurate starting point for the final visual servoing docking.

In Chapter 5, Sparus II AUV is used to obtain a full occupancy map, represented as a HM, obtained using range measurements from a sonar profiler. Raw sensor data from AHRS, DVL and depth sensors is used in a RBPF where each particle carries its own version of the HM and range measurements weight the particles according to their self-consistency with the map constructed from all previous measurements. In this chapter, the reader has been introduced to the principles of work of the HM, before introducing H-SLAM in Chapter 5.

3

Sum of Gaussian Single Beacon Range-Only Localization for AUV Homing

In this chapter, we propose a new approach for range-only localization based on the SOG filter. This solution also provides an AL method to allow the vehicle to decide the best movements to minimize the localization uncertainty. The methods are tested in a harbor scenario and visual feedback is used to check localization accuracy.

The proposed method has been published in the following paper:

Paper published in **Annual Reviews in Control** journal
Volume: 42, Pages: 177–187, Published: September 2016
DOI: 10.1016/j.arcontrol.2016.09.007
JCR2016 A&CS IF 2.627, Q2 (17/60)

Annual Reviews in Control 42 (2016) 177–187

Contents lists available at ScienceDirect

Annual Reviews in Control

journal homepage: www.elsevier.com/locate/arcontrol

Review

Sum of gaussian single beacon range-only localization for AUV homing☆

Guillem Vallicrosa*, Pere Ridao

Underwater Robotics Research Center (CIRS), Computer Vision and Robotics Institute (VICOROB), Universitat de Girona, Girona 17004, Spain

ARTICLE INFO

Article history:
Received 4 June 2016
Revised 15 August 2016
Accepted 9 September 2016
Available online 16 September 2016

Keywords:
AUV
Robot navigation
Range data
Kalman filters
Particle filters

ABSTRACT

Range-only measurements are extensively used in many Autonomous Underwater Vehicle (AUV) applications. These measurements do not depend on water quality and can be taken from long distances. This paper proposes two methods based on the Sum of Gaussian (SoG) filter, to solve the range-only localization problem for homing. The use of the SoG allows us to combine the benefits of both a Particle Filter (PF) and an Extended Kalman Filter (EKF) approach in a single filter. An Active Localization (AL) method is applied to the SoG to autonomously choose the best waypoints for autonomous convergence. Both the SoG filter and the AL are tested in a real scenario with an Intervention Autonomous Underwater Vehicle (I-AUV) and compared with a vision-based method to confirm localization.

1. Introduction

Artificial benthic constructions such as submerged oil fields and permanent observatories need periodic inspection and maintenance. Currently these operations are being carried out by Remoted Operated Vehicles (ROVs), operated from very expensive surface ships endowed with Dynamic Positioning (DP) systems and bulky Tether Management Systems (TMSs). However, there is an increasing interest in demonstrating these kinds of operations using light Intervention Autonomous Underwater Vehicles (I-AUVs) equipped with one or several arms to perform manipulation.

I-AUVs can be either launched from a small inexpensive surface boat or have a permanent Docking Station (DS) (Alves, Potter, Guerrini, Zappa, & Lepage, 2014) in the vicinity of the structure to be inspected and maintained. Either way, the vehicle needs some means to properly locate the structure to inspect and/or its DS for battery charging and data uploading. Conventional Autonomous Underwater Vehicles (AUVs) navigation systems are based on Inertial Navigation System (INS) with updates from sensors like the Doppler Velocity Log (DVL) (velocity), pressure-cell (depth) and Attitude and Heading Reference System (AHRS) (orientation). Unless externally aided with USLB/LBL/SBL, the navigation drifts over time and can become unreliable after a long inspection mission, making it impossible for the AUV to home back to the DS or to locate a desired structure to inspect (Ribas, Palomeras, Ridao, Carreras, & Mallios, 2012).

Methods based on visual detection of the structure only work in good visibility conditions, and at a few meters of distance. On the other side, acoustic methods can be used at larger distances regardless of the water quality. One such method for localization, is range-only localization. Range-only measurements are extensively used in many underwater homing applications (Blanco, Fernandez-Madrigal, & González, 2008a; Newman & Leonard, 2003; Olson, Leonard, & Teller, 2006; Vaganay, Baccou, & Jouvencel, 2000; Wang, Chen, Hu, & Gu, 2013). However, these measurements present a high non-linearity and standard Extended Kalman Filters (EKFs) cannot cope with it, unless delayed initializations are used (Webster, Eustice, Singh, & Whitcomb, 2009).

This paper presents a solution to the *range-only localization for homing* problem using a Sum of Gaussian (SoG) filter. Two variations of the SoG filter are proposed. In the first, each Gaussian represents a candidate position of the beacon (SoG-DR), while in the second, each Gaussian jointly represents a candidate pair of the beacon and AUV positions (SoG-SLAM). Both filters are tested in real experiments, where an AUV performs an autonomous localization and homing maneuver to a DS following the goals proposed by the Active Localization (AL) method. To assess their performance, ground truth measurements of the DS are provided using an accurate visual detection algorithm. Visual detection is aided by light beacons located on the DS structure

☆ This work was supported by the Spanish national funded project ARCHROV (Marine ARChaeology through HROV/AUV cooperation) DPI2014-57746-C3-3-R, and the LOON-DOCK/SUNRISE (Sensing, monitoring and actuating on the UNderwater world through a federated Research InfraStructure Extending the Future Internet) FP7-ICT-2013-10-611449 by the European commission, and the Spanish Government FPU12/05384 PhD grant (to G. Vallicrosa).

* Corresponding author.
E-mail address: gvallicrosa@eia.udg.edu (G. Vallicrosa).

http://dx.doi.org/10.1016/j.arcontrol.2016.09.007
1367-5788/

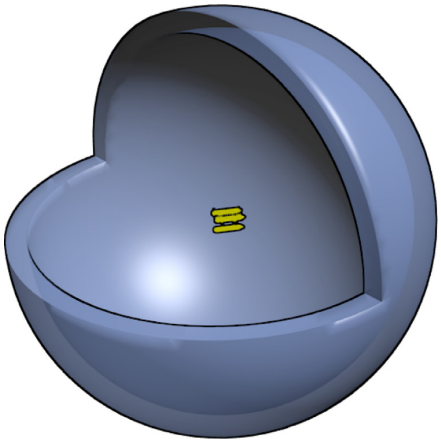


Fig. 1. Spherical shell describing all possible positions of an acoustic beacon given a single range measurement.

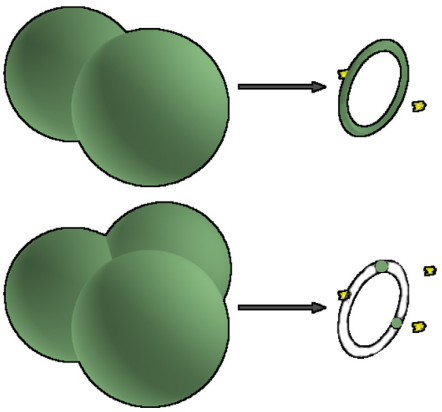


Fig. 2. Symmetries when intersecting two and three spheres.

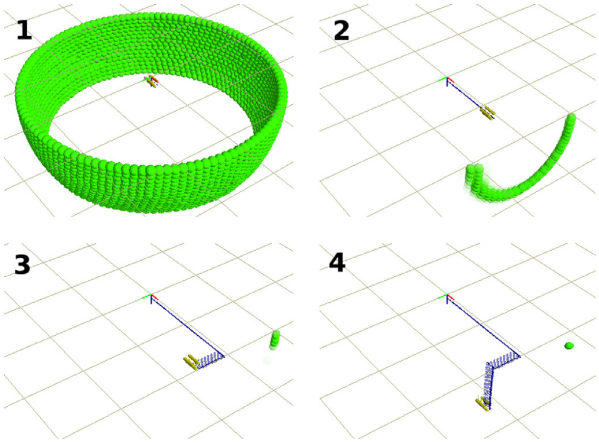


Fig. 3. Example of movement in three axis leading to a single location.

(Bosch, Gracias, Ridao, Istenic, & Ribas, 2016). This paper builds on previous works Vallicrosa, Ridao, and Ribas (2015); Vallicrosa, Ridao, Ribas, and Palomer (2014) extending them with new experiments in which the different algorithms are compared.

This paper is organized as follows. Section 2 describes the range-only localization problem along with the proposed solution, the SoG filter. It completes the description with its initialization, methods of performing AL re-sampling, and computation of the final localization of the acoustic beacon. Section 3 details the filter implementation. Section 4 describes the AUV and its sensors, along with the DS used during the experimental validation. Section 5 discusses the results both with and without using AL. Finally Sections 6 and 7 present the conclusions and plans for future work.

2. The range-only localization

Range-only localization is a highly non-linear problem. Given a range measure at a certain vehicle position, the acoustic beacon location can lie in any position on the spherical shell centered at the robot position. (Fig. 1). The shell radius is the measured range, and its thickness the range measurement uncertainty.

Particularly interesting are the symmetries produced when moving along a non-observable path (Vaganay et al., 2000) (i.e., a straight line). In 2D, the intersection of two circumferences are two points, and hence, to resolve this ambiguity a third circumference is necessary. The centre of this third circumference cannot be collinear with the centres of the other two circumferences, or the symmetry problem will remain.

In the 3D case, at least four sphere surfaces with non-coplanar centers are needed to solve for the beacon position while avoiding multiple solutions (Fig. 2). Given two spheres, their intersection is a circumference. Adding a third non-collinear sphere provides two candidate solutions. Finally, adding a fourth non-coplanar sphere narrows down localizations to a single point. This process of localization with range-only measurements is also known as Trilateration.

This means that to discover the 3D position of an acoustic beacon, an AUV must follow a 3D (Fig. 3) trajectory. However, if extra restrictions apply, such as when some of the solutions can be discarded because they lay above the water surface or below the sea bottom, the trajectory can be simplified.

Multiple methods to solve this localization problem using variations of the trilateration procedure have been proposed in the literature (Blanco et al., 2008a; Newman & Leonard, 2003; Olson et al., 2006; Vaganay et al., 2000; Wang et al., 2013). EKFs filters can be used when an *a priori* estimate of the beacon position is available or after a Gaussian initial guess has been computed by some means. For instance, a delayed EKF initialization (Webster et al., 2009) can be used to calculate this initial position before incorporating it into the EKF as a point feature. The main problem being that the EKF works with Gaussian uncertainties and a spherical shell uncertainty is impossible to properly represent by a single Gaussian.

The reported methods using delayed initializations include a least-squares approach (Newman & Leonard, 2003; Vaganay et al., 2000), a Hough Transform (Olson et al., 2006), an EKF with delayed states (Webster et al., 2009) and a Moving Horizon Estimation (Wang et al., 2013). The least-squares approach checks intersections between circumferences defined by vehicle position and range. The Hough Transform uses a voting space up to a certain resolution and takes the most voted cell as the final localization. The pose-based EKF with delayed states, contains the current state and the previous ones up to a specified time in the state vector to take into account the delay in the range measurement Time of Flight (TOF). The Moving Horizon Estimation uses a fixed window for the range update.

In the field of mobile robotics, a solution to this problem was proposed in Blanco et al. (2008a) with the use of a SoG filter. The system was demonstrated on a mobile land robot gath-

ering range measurements through a Ultra-WideBand (UWB) radio system. The SoG filter was compared to a Particle Filter (PF) previously proposed in Blanco, González, and Fernandez-Madrigal (2008b) achieving better accuracy with less computation time. Like PFs, the SoG filter can represent arbitrary multi-modal density functions providing, for some problems, a more efficient space representation. A SoG filter requires a significantly smaller number of Gaussians than the number of particles that would be needed by a PF to represent the same density function. The SoG filter also allows easy computation of the entropy. This entropy can be seen as the volume of uncertainty of the filter, the less entropy the more accurate the localization is. This entropy can be used to autonomously select a goal waypoint to drive the AUV in the direction which minimizes the predicted entropy, effectively reducing the localization uncertainty and autonomously localizing the DS.

In the next subsections, the implementation of the SoG filter is explained along with its initialization. In addition an AL method is proposed to ensure autonomous convergence of localization by reducing the entropy of the filter.

2.1. Sum of gaussian filter

The most common method of properly representing the space of possible beacon locations (spherical shell) would be the use of particles. However, even though PF are appropriate for non-linear filtering, a large number of particles would be necessary to cover the spherical shell at enough resolution, making the filter computationally demanding. To overcome this problem, a SoG approximation (Alspach & Sorenson, 1972; Kotecha & Djuric, 2003) is used to represent the belief of the map $\mathcal{M}$ (the beacon position) according to the odometry x_k and the measurements z_k:

$$p(\mathcal{M}|x_k, z_k) \approx \sum_{i=1}^{N} \upsilon_k^i \mathcal{N}\left(z_k; \mu_k^i, \Sigma_k^i\right) \tag{1}$$

where υ_k^i is the weight associated with each Gaussian, and μ_k^i and Σ_k^i are its mean and covariance matrix.

The representation of the possible beacon locations with Gaussians, allows us to cover all space with the use of less Gaussians than particles in a PF. Furthermore, the use of Gaussians allows use of an EKF to estimate their position and uncertainty according to the range measurements. A SoG-EKF implementation consists of the following steps:

1. **Prediction** Use EKF prediction equations on each Gaussian.
2. **Weighting**
   (a) Use a standard EKF update to update each Gaussian mean μ_k^i and uncertainty Σ_k^i.
   (b) Use the innovation y_k^i of the EKF update to update each Gaussian weight υ_k^i.
   (c) Normalize weights.
3. **Resampling** Optional step. PF-like re-samplings can be applied (discussed in Section 2.4).

For the sake of completeness EKF prediction and update equations are reproduced here. For simplicity, the following notation is going to be adopted, being $x_k = \mu_k^i$ and $P_k = \Sigma_k^i$. The prediction equations are:

$$x_{k|k-1} == f(x_{k-1}, u_k) \tag{2}$$

$$P_{k|k-1} == F_{k-1} P_{k-1} F_{k-1}^T + Q_k \tag{3}$$

where $x_{k|k-1}$ and $P_{k|k-1}$ are the predicted state vector and covariance matrix respectively, $f(x_{k-1}, u_k)$ is the process model equation and F_{k-1} its Jacobian, Q_k is the process noise covariance, u_k is the control vector.

In the update equations, the innovation y_k depends on the real measurement z_k and expected measurement $h(x_{k|k-1})$. H_k is the expected measurement Jacobian, R_k the observation noise, S_k the covariance of the innovation and K_k the Kalman gain.

$$y_k = z_k - h(x_{k|k-1}) \tag{4}$$

$$S_k = H_k P_{k|k-1} H_k^T + R_k \tag{5}$$

$$K_k = P_{k|k-1} H_k^T S_k^{-1} \tag{6}$$

$$x_k = x_{k|k-1} + K_k y_k \tag{7}$$

$$P_k = (I - K_k H_k) P_{k|k-1} \tag{8}$$

The weight update for each Gaussian in step 2(b) is computed following the likelihood $p(z|x)$ of the measurement. Innovation y_k^i reflects the difference between the real and the expected measurement and therefore is inversely proportional to the likelihood. Weight is computed with a standard exponential decay as follows:

$$\upsilon_k^i = \upsilon_{k-1}^i \exp\left(-\gamma [y_k^i]^2\right) \tag{9}$$

being i the index of the Gaussian. This equation maximizes the weight when the expected measurements and the real ones agree. The γ variable controls the degree by which the weight decreases when increasing the innovation. For this work, it is set as $\gamma = 0.5/\sigma_z^2$, where σ_z is the uncertainty of the range measurement.

2.2. Initialization of the SOG

The initialization of the SoG filter is produced in the first range measurement. Two different initializations are proposed, both covering the whole space of possible beacon locations. The first, the naive initialization, spreads Gaussians of a specified size over all possible latitudes and longitudes of the corresponding sphere. The second, the Geodesic Grid initialization, starts with an *a priori* fixed number of Gaussians to be used to represent the density function, and searches for their best location in order to cover the complete sphere. A fixed number of Gaussians bounds the computation power required for each range update, which is particularly important for real-time implementations.

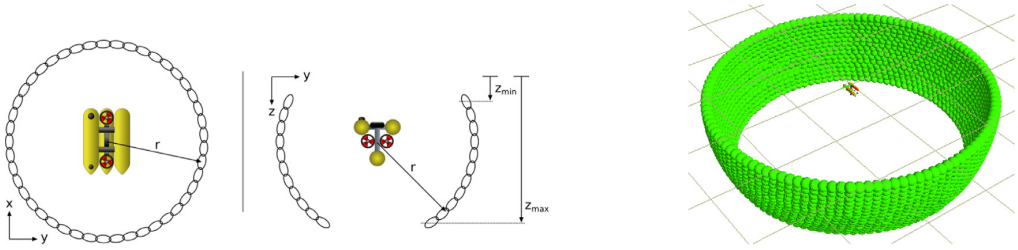
The vehicle depth and altitude are known with high precision and can be used to discard some of the candidate beacon locations. Assuming a flat seafloor terrain, the bottom altitude can be used to rule out candidate beacon locations lying below the seafloor. Likewise, depth information can be used to discard out-of-water positions. Furthermore, the depth location of the beacon can be roughly known in advance, in which case minimum and maximum values for depth can be specified.

There are two parameters that define each initialized Gaussian, the radial uncertainty σ_r (the thickness of the shell) which is equal to the range measurement uncertainty $\sigma_r = \sigma_z$, and the tangential uncertainty σ_t. The tangential uncertainty defines how much space each Gaussian covers on the spherical surface. All Gaussian weights are initialized to $1/N$ being N the total number of Gaussians.

2.2.1. Naive initialization

The naive initialization (Fig. 4) consists in using N Gaussians ($N = N_{lat} \times N_{lon}$) each one defined as

$$\mathcal{N}(\mu_k^i, \Sigma_k^i) = \mathcal{N}\left([z_k, \theta, \varphi]^T,\ diag(\sigma_r, \sigma_t, \sigma_t)\right) \tag{10}$$



(a) Concept. (b) Real initialization example.

Fig. 4. Naive initialization.

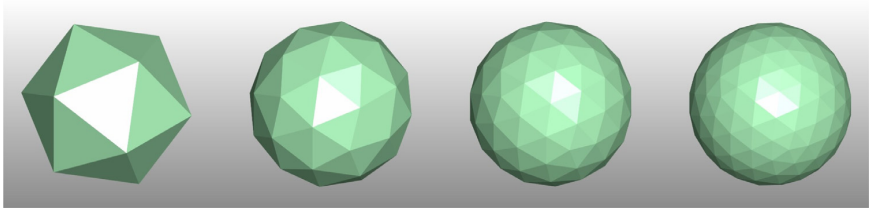


Fig. 5. Geodesic Grid with 12, 42, 162 and 642 vertices, corresponding to the subdivision of the original 12 vertices icosahedron (levels 0, 1, 2, 3) [Wikipedia].

in spherical coordinates (radius z_k, latitude θ and longitude φ), where σ_t is a user parameter defining the space resolution of the Gaussian. Given a range z_k, a certain latitude θ and a certain σ_t, it is possible to compute the number of Gaussians N_{lon} covering the complete circumference (all possible longitudes) (12). In the same way, given a range of latitudes $[\theta_{\min}, \theta_{\max}]$, a range measurement z_k and σ_t, it is possible to compute the number of Gaussians N_{lat} to cover all latitudes (11).

$$N_{lat} = \left\lceil \frac{z_k(\theta_{max} - \theta_{min})}{2\sigma_t} \right\rceil \tag{11}$$

$$N_{lon} = \left\lceil \frac{2\pi z_k \cos\theta}{2\sigma_t} \right\rceil \tag{12}$$

2.2.2. Geodesic grid initialization

The Geodesic Grid initialization is based on the concept of Geodesic Grids (Heikes & Randall, 1995), often used when modelling weather and ocean circulation. Geodesic Grid is a technique used to model a surface of a sphere (such as the Earth) with a subdivided polyhedron, usually an icosahedron.

The construction of the grid is generated by the subdivision of an icosahedron into cells by iteratively bisecting the edges of the polyhedron and re-projecting the new cells onto the sphere (Fig. 5).

This approach provides a fixed number of equally spaced Gaussians on the sphere surface. This is particularly important when dealing with real-time applications, where only a limited number of Gaussians can be processed in the available time.

This initialization also utilizes the minimum and maximum depth information to discard Spherical Grid cells at initialization (Fig. 6). All the Gaussian centers are pre-computed at different Geodesic Grid levels (Table 1) for a unitary sphere. On the filter initialization, these centers are scaled by the measured range z_K. In this case, σ_t is not user-provided but computed according to the unitary sphere edge length.

2.3. Active localization with a SOG

As mentioned earlier, the main goal is to autonomously locate the beacon. However, if the AUV chooses a trajectory such that the range-only measurements are not useful enough to localize the beacon, the whole localization solution may lead to unsatisfactory results. Therefore we would like to determine the optimal motion of the AUV such that the next range measurement, reveals important information and helps reduce the filter uncertainty, which is deeply related to entropy. The entropy for a Gaussian Random Vector μ with covariance matrix Σ can be computed in closed form as:

Table 1
Geodesic grid levels with number of vertices and unitary edge length.

Level	Vertices	Edge/Radius
0	12	1.0514622
1	42	0.5465330
2	162	0.3669588
3	642	0.2759044

$$H_{\backslash}(\mu) = \frac{1}{2}\ln\left((2\pi e)^n \det(\Sigma)\right) \tag{13}$$

where n is the dimension of the vector. Entropy can be interpreted as the volume of the uncertainty bounding hyper-ellipsoid (Sim & Roy, 2005). Minimizing this entropy is equivalent to the D-optimality criterion (Carrillo, Reid, & Castellanos, 2012). Therefore, the value of the entropy for a single Gaussian can be computed as the product of the eigenvalues λ of its uncertainty matrix Σ_k^i. For better numerical stability the entropy $\mathcal{E}$ can be computed in the logarithmic space as:

$$\log \mathcal{E}_j = \sum \log \lambda_{\Sigma_k^i} \tag{14}$$

This equation can also be used to compute the entropy of the complete SoG filter using the global uncertainty estimate of the beacon $\hat{\Sigma}_k$ which is computed proportionally to the weights and uncertainty matrices of each Gaussian as:

$$\hat{\Sigma}_k = \sum_{i=1}^{N} \upsilon_k^i \Sigma_k^i \tag{15}$$

In order to actively decide (on-line) the following waypoint for the AUV, a set of candidate directions are proposed. For example,

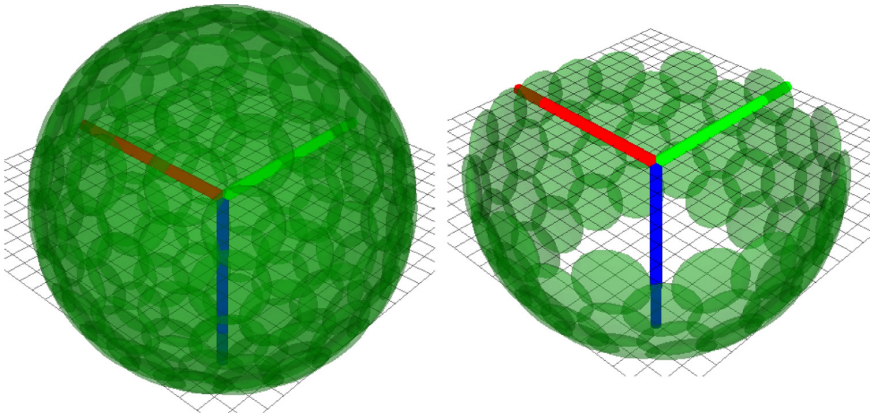


Fig. 6. Example of the geodesical grid initialization without and with depth limitations.

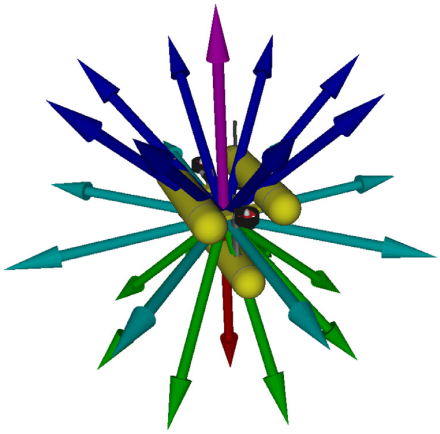


Fig. 7. 26 proposed actions for active localization ($8 \times 3 + 2$).

the proposed actions could be of a fixed displacement of $3m$ and spread at 45 ° intervals over the horizontal plane at three different levels (−45, 0 and 45 °), plus actions for moving up and downwards (Fig. 7).

The AL method implemented in this paper, takes advantage of the fact that in an EKF framework, the uncertainty obtained after the update step does not depend on the measurement itself, but only on its uncertainty (8). For each Gaussian in the SoG, each candidate motion is evaluated providing a predicted beacon uncertainty Σ^i_{k+1} whose entropy is calculated with (14).

A voting scheme is setup, where each Gaussian votes with its weight whichever action among those proposed minimizes the predicted entropy. The action that receives more votes is set as the next motion goal. This goal is the optimal among the proposed candidate motions, but might not be the global optimal one.

2.4. Resampling with a SOG

In a PF, noise is introduced in the prediction step to account for the motion model uncertainty. This noise is essential for the particles, because otherwise the motion wouldn't be probabilistic but deterministic. During the resampling process of a PF, some high-weight particles get duplicated while other low-weight particles disappear. If for instance two particles are resampled to the same position and there is no noise in the prediction step, both will always behave exactly in the same way and therefore one of them

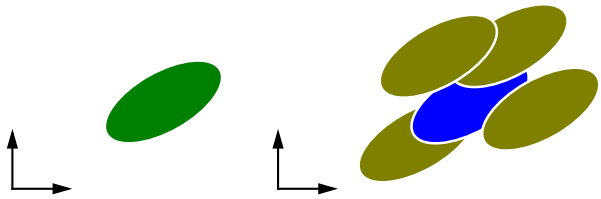


Fig. 8. *Left:* Gaussian of the beacon position. *Right:* First resampled Gaussian in blue and following resamplings over same Gaussian using beacon position sampling.

becomes redundant. The same problem would appear in the proposed SoG filter implementation. Since the prediction step is based on the EKF prediction step, no difference is observable between two co-located Gaussians. Hence, if the optional resampling step needs to be added to the SoG filter, this must introduce some noise to avoid having duplicated Gaussians.

The proposed resampling for the SoG filter is the following. After using a common Sequential Importance Resampling (SIR) (Del Moral, Doucet, & Jasra, 2012) used in PFs and before the actual copy of the Gaussians, an extra rule is added. If the copy is the first one, an exact copy is obtained. If the copy is not the first one, a random copy is obtained based on the current uncertainty of the acoustic beacon position (Fig. 8). This method ensures copies of the Gaussian that are close enough and introduces the necessary noise. The resampling is only applied if the effective number of particles N_{eff} (Del Moral et al., 2012) is smaller than $N/2$, where N is the number of particles and $N_{eff} = 1/\sum_i w^{i2}$ reflects the variance of the weights.

2.5. Choosing a solution

Having a multimodal function as the SoG filter does not facilitate the selection of a solution as final localization for homing and docking purposes. A solution should be accepted when the SoG contains a single blob of weighted Gaussians and they are concentrated in a small area/volume. When this occurs, the SoG can be terminated and a reliable acoustic beacon location obtained. This location can be later used for a homing procedure or added as a feature to the EKF navigation to bound the Dead Reckoning (DR) navigation drift.

We adopt an approach that searches for an equivalent Gaussian to represent the whole SoG filter using the Unscented Transform (Wan & Van Der Merwe, 2000). If multiple random points are drawn from each of the Gaussians, a mean and covariance can

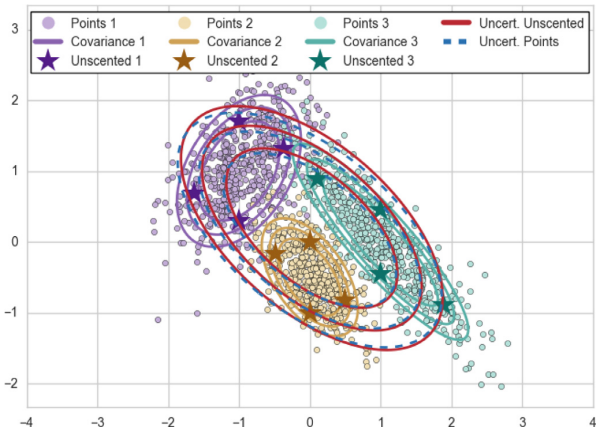


Fig. 9. Comparison of equivalent covariance calculated from all points (blue) or from the Unscented Transform (red). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

be calculated from all of them. To make this process less computationally expensive, the Unscented Transform is used to obtain $2n$ points per Gaussian, being n the number of dimensions. The points can be computed as:

$$\mathcal{X}_j^i = \mu^i + \left(\sqrt{n\Sigma^i}\right)_j \quad \text{for } j = 1, \ldots, n \tag{16}$$

$$\mathcal{X}_j^i = \mu^i - \left(\sqrt{n\Sigma^i}\right)_{n-j} \quad \text{for } j = n+1, \ldots, 2n \tag{17}$$

being i the index of the Gaussian. For example, for a 3D acoustic beacon localization, six points are obtained for each Gaussian and their weights are equal to the weight of the Gaussian. Then the equivalent Gaussian $\mathcal{N}(\hat{\mu}, \hat{\Sigma})$ can be computed as:

$$\hat{\mu} = \frac{1}{\sum_i \upsilon^i} \sum_i \upsilon^i \mu^i \tag{18}$$

$$\hat{\Sigma} = \frac{1}{2n \sum_i \upsilon^i} \sum_i \left(\upsilon^i \sum_j (\mathcal{X}_j^i - \hat{\mu})(\mathcal{X}_j^i - \hat{\mu})^T \right) \tag{19}$$

In order to check the equivalence of the use of the Unscented Transform, 1000 random points were sampled from different Gaussians. Those random points were then used to compute an equivalent Gaussian from all of them. This result was compared (Fig. 9) with the one obtained with the points from the Unscented transform as follows:

This method is used in order to determine a final location of the acoustic beacon from the SoG filter. A user-defined distance threshold is used to determine if the 2σ-bounds of the equivalent Gaussian fall below its value. When this condition is met, the SoG filter is considered finished and the acoustic beacon localized.

3. Filters for SoG range-only localization

Two different filters are presented in this paper: 1) a SoG filter which estimates the position of the beacon from DR navigation and 2) a SoG filter which jointly estimates the beacon position as well as the AUV's position and velocity.

Although both initializations can be used by both filters, each of the proposed algorithms only implements one, providing already enough results on real data.

3.1. Localizing the DS with DR navigation

According to this filter formulation, first presented in Vallicrosa et al. (2014), the filter state vector only contains the position of the beacon. The SoG filter is initialized at the first range using the naive initialization (Section 2.2.1) and the DR navigation is taken directly from the one already available in the AUV software architecture (Palomeras, El-Fakdi, Carreras, & Ridao, 2012). All Gaussians are preserved throughout the process and no resampling is performed (Algorithm 1).

Algorithm 1 Algorithm for DR3D and DR2D.

```
// Init filter
z_k = getRangeFromAcousticModem()
x_k = getDRodometry()
sog, N = initFilterNaive(x_k, z_k) // return SoG and number of Gaussians N (Section 2.2.1)
// Continue until localization threshold passed (equivalent Gaussian using Eq. (19) and Eq. (20))
while sog.getEquivalentGaussian() > LocalizationThreshold do
  // Get measurements
  z_k = getRangeFromAcousticModem()
  x_k = getDRodometry()
  // Update filter
  for i = 0 to i = N − 1 do
    y_k^i = sog.Gaussian[i].EKFupdate(x_k, z_k)
    sog.Gaussian[i].weight(y_k^i) // using Eq. (10)
  end for
  sog.normalizeWeights()
  // Active localization (see Section 2.3)
  if UseActiveLocalization then
    votes = zeros(num_ProposedActions)
    for i = 0 to i = N − 1 do
      action_id = sog.Gaussian[i].getSmallestEntropyAction(ProposedActions)
      votes[action_id] = votes[action_id] + sog.Gaussian[i].weight
    end for
    goal = getMostVotedAction(votes)
    sendGoalToVehicle(goal)
  end if
end while
```

Two different versions of the algorithm are presented. The first one assumes that the beacon depth is unknown (named as DR3D, Dead-Reckoning full 3D estimation). Hence, the state vector of each Gaussian is 3D. The second version assumes a known depth (named as DR2D) using a 2D state vector for each Gaussian. In this case, the range measurements z_k have to be projected into the XY plane before introducing the measure to the filter:

$$\hat{z}_k = \sqrt{z_k^2 - (z_b - z_r)^2} \tag{20}$$

where z_r is the current vehicle depth and z_b is the known beacon depth.

Known depth is a straightforward assumption, since depth measurements are precise and easy to take either in the AUV or when the DS is placed underwater. However, tides must be taken into account for correct localization. The reduction of one dimension also allows location of the acoustic beacon, using only movements in the xy-plane.

Because the depth of the beacon is known, the AUV usually operates around the z_b depth. So z_b is almost equal to z_r and hence $\hat{z}_k$ is almost equal to z_k allowing for a good Gaussian approximation, where its uncertainty is obtained through the projection of the corresponding Jacobian of (20).

G. Vallicrosa, P. Ridao / Annual Reviews in Control 42 (2016) 177–187 183

The state vector μ_k^i, the process model $f(\mu_k^i)$ and the measurement model $h(\mu_k^i)$ are the following:

$$\mu_k^i = [x_b, y_b, z_b]^T \tag{21}$$

$$f(\mu_k^i) = \mu_k^i \tag{22}$$

$$h(\mu_k^i) = ||(x_b, y_b, z_b) - (x_r, y_r, z_r)|| \tag{23}$$

where $[x_r, y_r, z_r]$ is the vehicle position obtained from DR navigation. In the DR2D the equations are modified by avoiding the use of z coordinates.

3.2. Localizing the DS with SLAM

First presented in Vallicrosa et al. (2015), it contains a single-feature Simultaneous Localization and Mapping (SLAM) in each Gaussian., where the state vector μ_k^i, the process model $f(\mu_k^i)$ and the measurement model $h(\mu_k^i)$ are the following:

$$\mu_k^i = [x_r \quad y_r \quad z_r \quad \dot{x}_r \quad \dot{y}_r \quad \dot{z}_r \quad x_b \quad y_b \quad z_b]^T \tag{24}$$

$$f(\mu_k^i) = \begin{bmatrix} \bar{x}_{r,k} \\ \bar{y}_{r,k} \\ \bar{z}_{r,k} \end{bmatrix} = \begin{bmatrix} x_{r,k-1} \\ y_{r,k-1} \\ z_{r,k-1} \end{bmatrix} + Rot(\phi_k, \theta_k, \psi_k) \cdot \begin{bmatrix} \dot{x}_{r,k-1} \\ \dot{y}_{r,k-1} \\ \dot{z}_{r,k-1} \end{bmatrix} \cdot \Delta t \tag{25}$$

$$h(\mu_k^i) = ||(x_b, y_b, z_b) - (x_r, y_r, z_r)|| \tag{26}$$

being $[x_r, y_r, z_r]$ the AUV position in the world frame, $[\dot{x}_r, \dot{y}_r, \dot{z}_r]$ the AUV velocities in the vehicle frame, and $[x_b, y_b, z_b]$ the acoustic beacon location in the world frame, Δt the time increment between measurements at current time k and previous measurements at time $k-1$, and $Rot(\phi_k, \theta_k, \psi_k)$ the rotation matrix with respect to the vehicle orientation.

The SoG filter is initialized at the first range using the Geodesic grid initialization (Section 2.2.2) and the position and velocities of the AUV estimated through DR. Each candidate beacon position is initialized at a different position on the spherical shell, and all AUV positions are defined as the current DR position estimate.

At each sensor measurement, prediction and update steps of the EKF are performed in the same way as they are performed in the AUV's own navigation filter. Those sensors include an AHRS, a pressure sensor and a DVL. All Gaussians are maintained throughout the process and resampled as explained in Section 2.4 (Algorithm 2).

Algorithm 2 Algorithm for SLAM.

```
// Init filter
z_k = getRangeFromAcousticModem()
x_k = getDRodometry()
sog, N = initFilterGeodesicGrid(x_k, z_k) // return SoG and number of Gaussians N (Section 2.2.2)
// Continue until localization threshold passed (equivalent Gaussian using Eq. (19) and Eq. (20))
while sog.getEquivalentGaussian() > LocalizationThreshold do
  // Navigation sensors and DR navigation
  ahrs, depth, dvl = getNavigationSensors()
  sog.EKFpredict&update(ahrs, depth, dvl)
  // Update filter
  z_k = getRangeFromAcousticModem()
  for i = 0 to i = N − 1 do
    y_k^i = sog.Gaussian[i].EKFupdate(z_k)
    sog.Gaussian[i].weight(y_k^i) // using Eq. (10)
  end for
  sog.normalizeWeights()
  // Resampling (Section 2.4)
  if sog.computeNeff() < N/2 then
    sog.resamplingWithNoise()
  end if
  // Active localization (see Section 2.3)
  if UseActiveLocalization then
    votes = zeros(num_ProposedActions)
    for i = 0 to i = N − 1 do
      action_id = sog.Gaussian[i].getSmallestEntropyAction(ProposedActions)
      votes[action_id] = votes[action_id] + sog.Gaussian[i].weight
    end for
    goal = getMostVotedAction(votes)
    sendGoalToVehicle(goal)
  end if
end while
```

4. Experimental setup

The three algorithms proposed in the previous section (DR2D, DR3D and SLAM) were experimentally validated in real-world conditions in Sant Feliu de Guixols' harbor. A DS with an acoustic modem mounted on it is placed on a flat terrain without any obstacles in the close vicinity. A Girona500 AUV (Ribas, Ridao, Magi, Palomeras, & Carreras, 2011) measures ranges to the acoustic modem and uses these measurements to locate the DS. Once the DS is located, the vehicle homes to the detected position for visual confirmation.

4.1. Girona 500 I-AUV

The Girona 500 is a compact-size AUV designed and developed in the University of Girona for a maximum operating depth of 500 m. The vehicle is built around an aluminum frame which supports three torpedo-shaped hulls as well as other elements such as the thrusters (Fig. 10). The overall dimensions of the vehicle are 1.0 m in height, 1.0 m in width, 1.5 m in length, and a weight (in its basic configuration) of about 140 kg. The two upper hulls, which contain the flotation foam and the electronics housing, are positively buoyant, while the lower one contains the heavier elements such as the batteries and the payload. This particular arrangement of the components provides the vehicle with passive

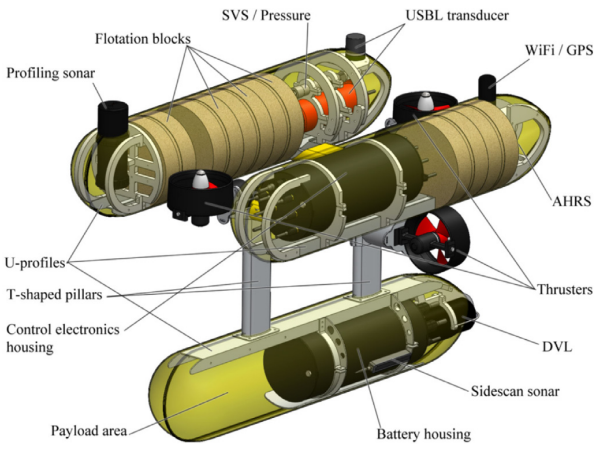


Fig. 10. Girona 500 I-AUV with detailed positioning of its components. An electric 4 degrees of freedom (DoF) arm can be mounted on the payload area for intervention purposes.

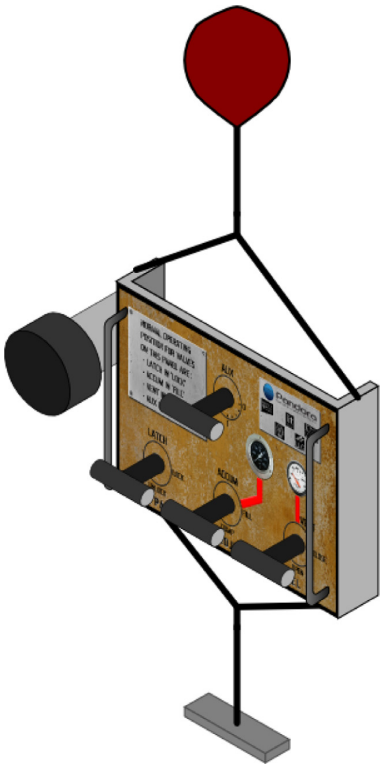


Fig. 11. Mock-up docking station panel used in PANDORA project for free-floating manipulation.

stability in pitch and roll, making it suitable for tasks requiring a stable platform such as video surveying or intervention. The most remarkable characteristic of the Girona 500 is its capacity to be reconfigured for different tasks. On its basic configuration, the vehicle is equipped with typical navigation sensors, DVL, AHRS, pressure gauge and Ultra-Short Baseline (USBL). In addition to these sensors, almost half of the volume of the lower hull is reserved for mission-specific payload, which makes it possible to modify its sensing and actuation capabilities as required. The addition of an electric-actuated 4DoF arm provides the intervention capabilities necessary in an I-AUV.

Navigation sensor measurements are fused in the Girona 500 in the EKF navigation of its software architecture Component Oriented Layer-based Architecture for Autonomy (COLA2). The navigation frame is defined as the North East Depth (NED) frame, and can be aided with Global Positioning System (GPS) information when the vehicle is on surface or USBL when underwater. An acoustic modem is used to obtain a range measurement against the modem on the DS. Finally, a forward-looking camera was incorporated to visually confirm correct homing.

4.2. Docking station

In order to emulate a DS panel, a custom mock-up panel from the FP7 PANDORA project (Lane et al., 2012) was used. It consists of four valves that can be turned and a custom printed background to look like a real one (Fig. 11).

The panel was also equipped with an acoustic modem to be able to measure ranges relative to it. In order to obtain a ground truth location of the DS, a set of four light beacons (one at each of the corners) were mounted. These LED light beacons were detected with a vehicle-mounted camera, and a vision-based algorithm provided a global localization of the light beacons. Given the high accuracy of the DS pose estimation using visual markers, those measurements were used as ground-truth.

Table 2
Parameters and algorithms used for localization at the two batches of experiments performed at sea.

Batch	Type	σ_z	σ_t	min_z	max_z	AL
1 (Exp 1–4)	DR3D	1.0	1.0	3.0	10.0	✗
2 (Exp 5–12)	DR3D	2.0	1.0	3.0	10.0	✓

Table 3
Localization errors (in the 2D plane and in 3D) against Ground truth (vision) for each of the experiments.

Experiment	MAN/AL	$Error_{xyz}$	Experiment	MAN/AL	$Error_{xyz}$
1	MAN	2.95	7	AL	1.46
2	MAN	0.98	8	AL	1.36
3	MAN	2.18	9	AL	1.54
4	MAN	1.61	10	AL	3.29
5	AL	3.72	11	AL	0.70
6	AL	2.05	12	AL	0.75

5. Results

In order to test the performance of the algorithms, two batches of experiments were performed at sea on different days (Table 2). The first batch was manually tele-operated (experiments 1–4) while the second one (experiments 5–12) was guided by the AL algorithm.

All experiments started at different positions (Fig. 12). The AUV was first commanded either by the AL, or manually, until the acoustic beacon was located with enough precision. Next, the AUV was guided straight to the detected beacon location to obtain visual confirmation. This visual confirmation was used as ground truth measurement to compare with the localization provided by the SoG filter.

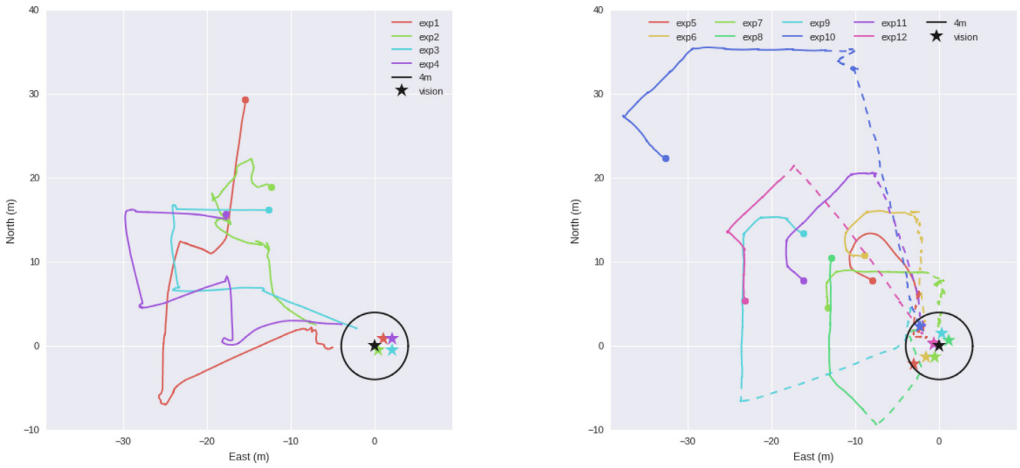
To achieve the visual confirmation, the AUV uses the imagery provided by its forward-looking camera (Fig. 13). The algorithm described in Bosch et al. (2016) is used to detect the light beacons in the images when visible. The algorithm detects the lights, identifies them based on their coded blinking frequency, and estimates the AUV position with respect to the DS based on the *a priori* knowledge of their position within the DS panel. The relative DS-to-AUV position obtained by the light beacons is then compounded with the AUV navigation to obtain the world frame DS-position for evaluation purposes.

The localization errors obtained by both batches at sea are given in Table 3. The results allowed us to conclude that in all experiments the AUV was able to home onto the DS with an error smaller than 4 m, this being enough to establish visual confirmation and detection of the DS. If we take a detailed look at some experiments, examining all the ranges obtained during localization (Fig. 14), it can be appreciated how challenging the process of estimating the correct location of the beacon is, especially during the first movements.

5.1. SoG filter with active localization

Active Localization (AL) was tested on Batch 2 using the DR3D filter. The vehicle demonstrated that even with non-perfect range measurements it was capable of locating the DS autonomously and home to it. The localization errors obtained, which are below 4.0 m, are acceptable values for docking purposes as demonstrated by the successful visual detection before docking.

G. Vallicrosa, P. Ridao / Annual Reviews in Control 42 (2016) 177–187 185



(a) Batch 1 teleoperated. (b) Batch 2 wirh AL.

Fig. 12. Trajectories followed by the vehicle in each experiment. *(circle)* starting position of the AUV. *(unbroken line)* trajectory during the SoG algorithm execution. *(dashed line)* approach trajectory to the detected beacon position. (star) detected beacon position by SoG filter. *(black)* star representing the ground truth position detected by visual confirmation with a 4 m radius circle where detection is enough for docking purposes.

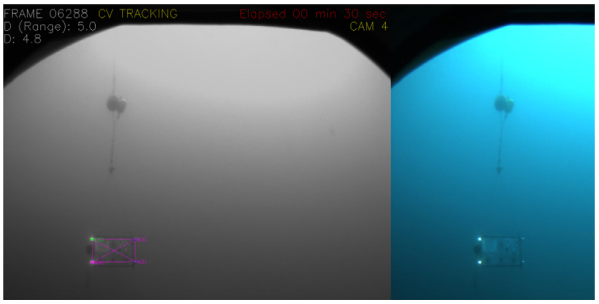


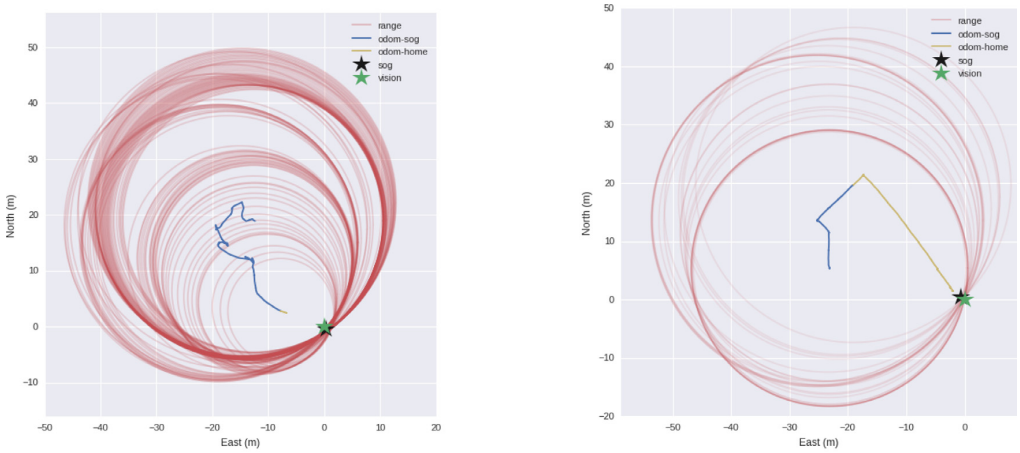
Fig. 13. Visual detection of the light beacons on the panel.

It is worth noting that while searching for the next goal with the Active Localization algorithm, the entropy is minimized while correctly locating the acoustic beacon (Fig. 15). All experiments show a clearly decreasing trend. Note also that the trajectories performed by the vehicle (Fig. 12b) tend to make sharp changes in the direction of movement in order to decrease the entropy of the filter.

Finally, it can be appreciated in Fig. 16 that for all the experiments the localization error is decreasing over time in concert with the entropy. A huge entropy decrease is observed when localization symmetries are solved.

5.2. SLAM filter

To assess the performance of the SLAM filter, the two batches of datasets were run again using the playback mechanism provided by the Robot Operating System (ROS). By this means, the actual sensor measurements messages logged during the experiment are reproduced in real-time and the navigation filter is run on-line, as if it were running in real-time on the AUV. This allows the running of new navigation algorithms on already available data-sets in real-



(a) Experiment 2 (batch 1). (b) Experiment 12 (batch 2).

Fig. 14. Examples of the AUV's trajectories with the corresponding range measurements.

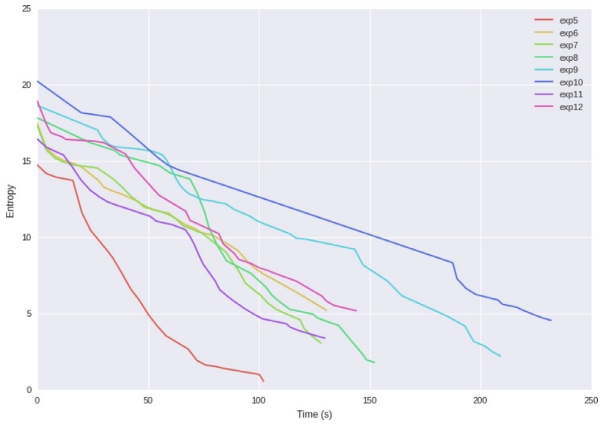


Fig. 15. Batch 2 entropy over time for each of the experiments (computed using (14) and (15)).

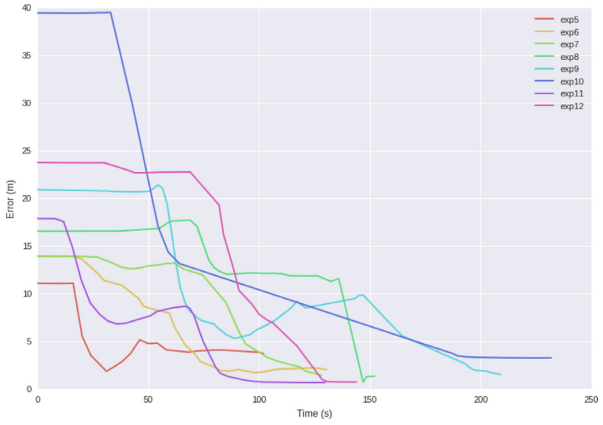


Fig. 16. Batch 2 errors over time for each of the experiments.

time conditions. However, the AL algorithm is not used since it is recorded data, so it is not possible to change the trajectory that was followed by the AUV.

The Table 4 and the Fig. 17 report the estimation errors obtained for each algorithm. Note that Fig. 17 also contains the results obtained in the *LIVE* trials at the harbour which are already shown in Table 3. In this case we have differentiated between the

Table 4
Localization errors against visual ground truth.

Experiment	DR2D	DR3D	SLAM	Experiment	DR2D	DR3D	SLAM
1	3.34	3.17	3.31	7	0.91	1.18	1.68
2	0.82	1.16	2.14	8	2.13	1.25	1.32
3	2.27	2.16	2.35	9	0.66	1.33	0.24
4	2.01	1.73	2.35	10	3.22	3.27	2.76
5	2.17	3.59	3.81	11	0.64	0.60	0.56
6	0.87	1.70	1.94	12	0.87	0.71	0.91

SoG filter running in 2D (assuming that the DS depth is *a priori* known) and in 3D.

From the analysis of the results it is not possible to conclude that one of the methods performed better than the others. All of them seem able to solve the problem with an error smaller than a value around 4 m, which is enough to establish visual contact to guide a vision-based docking process after homing. However, having a known depth offers clear advantages in most of the experiments, providing a much better localization.

6. Conclusions

This paper presents two range-only localization filters for AUV homing to a DS, based on the SoG filter. The first, depends on the DR navigation and is tested in both 2D (with DS depth known) and 3D. The second includes a full SLAM in each Gaussian and it is implemented for the 3D case. Different initialization and resampling methods are also tested with both filters.

Both filters exhibit accurate localization in a real scenario of an unknown acoustic beacon. Results are corroborated by a vision-based algorithm which detects the DS. The datasets obtained from real sea experiments were used to compare all algorithms, achieving a similar performance over a total of 8 different trajectories. Final localization errors are below the 4 m bound necessary for accurate visual detection, needed for the inspection/docking task. The analysis of the results does not provide insights into which algorithm performs better.

The addition of an AL method based on the predicted entropy of the filter after a given action is taken, enables the vehicle to autonomously select waypoints. This procedure allows for a complete autonomous homing without any human intervention, which is the goal of the LOON-DOCK project.

7. Future work

Since all algorithms performed similarly in the experimental test cases, the simplest one, DR2D, will be used for future work.

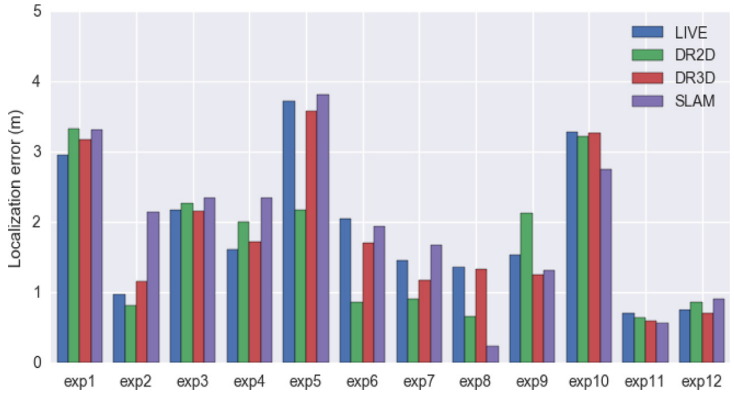


Fig. 17. Comparison of total localization errors $||(e_x, e_y, e_z)||$ per experiment and algorithm. Includes results of the *live* experiments Table 3 and the replayed data Table 4.

This range-only acoustic localization will be used for docking in the LOON-DOCK project (Alves et al., 2014). First the DS will be localized with the DR2D filter. Then, it will be added to the EKF-SLAM navigation filter as a feature for navigational drift correction. Then the vehicle will home to the estimated position until light beacons are detected. From there, the vehicle will autonomously dock with visual feedback. Once docked, it will transmit the data gathered during the mission to the base station.

Since, unlike Girona 500, most AUVs are non-holonomic, the study of the kinematic constraints of a torpedo-like vehicle will be very valuable, proposing smooth candidate actions in the AL algorithm.

References

Alspach, D., & Sorenson, H. (1972). Nonlinear bayesian estimation using gaussian sum approximations. *IEEE Transactions on Automatic Control, 17*(4), 439–448. doi:10.1109/TAC.1972.1100034.

Alves, J., Potter, J., Guerrini, P., Zappa, G., & Lepage, K. (2014). The LOON in 2014: test bed description. *Underwater communications and networking (UComms)* (pp. 1–4). IEEE.

Blanco, J.-L., Fernandez-Madrigal, J.-A., & González, J. (2008a). Efficient probabilistic range-only SLAM. In *IEEE/RSJ international conference on intelligent robots and systems (iros 2008)* (pp. 1017–1022). IEEE.

Blanco, J.-L., González, J., & Fernandez-Madrigal, J.-A. (2008b). A pure probabilistic approach to range-only SLAM. In *IEEE international conference on robotics and automation (ICRA 2008)* (pp. 1436–1441). IEEE.

Bosch, J., Gracias, N., Ridao, P., Istenic, K., & Ribas, D. (2016). Close-range tracking of underwater vehicles using light beacons. *Sensors, 16*(4), 429.

Carrillo, H., Reid, I., & Castellanos, J. A. (2012). On the comparison of uncertainty criteria for active SLAM. In *IEEE international conference on robotics and automation (ICRA 2012)* (pp. 2080–2087). IEEE.

Del Moral, P., Doucet, A., & Jasra, A. (2012). On adaptive resampling strategies for sequential monte carlo methods. *Bernoulli, 18*(1), 252–278. doi:10.3150/10-BEJ335.

Heikes, R., & Randall, D. (1995). Numerical integration of the shallow-Water equations on a twisted icosahedral grid. part i: Basic design and results of tests. *Monthly Weather Review, 123*(6), 1862–1880.

Kotecha, J. H., & Djuric, P. M. (2003). Gaussian sum particle filtering. *IEEE Transactions on Signal Processing, 51*(10), 2602–2612. doi:10.1109/TSP.2003.816754.

Lane, D. M., Maurelli, F., Kormushev, P., Carreras, M., Fox, M., & Kyriakopoulos, K. (2012). Persistent autonomy: The challenges of the PANDORA project. *Proceedings of IFAC MCMC*.

Newman, P., & Leonard, J. (2003). Pure range-only sub-sea SLAM. In *Ieee international conference on robotics and automation (icra 2003): vol. 2* (pp. 1921–1926). IEEE.

Olson, E., Leonard, J. J., & Teller, S. (2006). Robust range-only beacon localization. *Oceanic Engineering, IEEE Journal of, 31*(4), 949–958.

Palomeras, N., El-Fakdi, A., Carreras, M., & Ridao, P. (2012). COLA2: A control architecture for AUVs. *IEEE Journal of Oceanic Engineering, 37*(4), 695–716.

Ribas, D., Palomeras, N., Ridao, P., Carreras, M., & Mallios, A. (2012). Girona 500 AUV: From survey to intervention. *IEEE/ASME Transactions on Mechatronics, 17*(1), 46–53.

Ribas, D., Ridao, P., Magi, L., Palomeras, N., & Carreras, M. (2011). The Girona 500, a multipurpose autonomous underwater vehicle. In *Oceans 2011 IEEE - Spain* (pp. 1–5). IEEE. doi:10.1109/Oceans-Spain.2011.6003395.

Sim, R., & Roy, N. (2005). Global A-optimal robot exploration in SLAM. In *Proceedings of the 2005 IEEE international conference on robotics and automation* (pp. 661–666). IEEE.

Vaganay, J., Baccou, P., & Jouvencel, B. (2000). Homing by acoustic ranging to a single beacon. In *Oceans 2000 MTS/IEEE conference and exhibition: vol. 2* (pp. 1457–1462). IEEE.

Vallicrosa, G., Ridao, P., & Ribas, D. (2015). AUV single beacon range-only SLAM with a SOG filter. In *4th IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles (NGCUV 2015): vol. 48* (pp. 26–31). IEEE.

Vallicrosa, G., Ridao, P., Ribas, D., & Palomer, A. (2014). Active range-only beacon localization for AUV homing. In *2014 IEEE/RSJinternational conference on intelligent robots and systems (IROS 2014)* (pp. 2286–2291). IEEE.

Wan, E. A., & Van Der Merwe, R. (2000). The unscented Kalman filter for nonlinear estimation. In *The IEEE adaptive systems for signal processing, communications, and control symposium (as-spcc 2000)* (pp. 153–158). IEEE.

Wang, S., Chen, L., Hu, H., & Gu, D. (2013). Single beacon based localization of AUVs using moving horizon estimation. In *IEEE/RSJ international conference on intelligent robots and systems (IROS 2013)* (pp. 885–890). IEEE.

Webster, S. E., Eustice, R. M., Singh, H., & Whitcomb, L. L. (2009). Preliminary deep water results in single-beacon one-way-travel-time acoustic navigation for underwater vehicles. In *IEEE/RSJ international conference on intelligent robots and systems (IROS 2009)* (pp. 2053–2060). IEEE.

4

AUV HOMING AND DOCKING FOR REMOTE OPERATIONS

In this chapter, the range-only localization method explained in Chapter 3 is integrated in a cost-effective autonomous homing and docking strategy to provide remote AUV deployment/recovery in the framework of the EU funded project LOON-DOCK/SUNRISE. The algorithm is simplified to the 2D case where the depth of the DS is known. Range-only localization offers a first estimate of the DS position, that is then confirmed by visual localization that aids in the last meters of the docking procedure. The homing and docking strategy was successfully tested on the LOON testbed in La Spezia (Italy).
The thesis author contribution is focused on the range-only localization algorithm as well as the systems integration required to interact with the other modules such as mission control and visual localization. The author also helped to develop the simulated environment [Vallicrosa et al., 2016]. He also demonstrated the system in simulation and contributed to the experimental demonstration.

The proposed method has been published in the following paper:

Paper published in **Ocean Engineering** journal
Volume: 154, Pages: 106–120, Published: February 2018
DOI: 10.1016/j.oceaneng.2018.01.114
JCR2016 ME IF 1.894, Q1 (2/14)

Ocean Engineering 154 (2018) 106–120

Contents lists available at ScienceDirect

Ocean Engineering

journal homepage: www.elsevier.com/locate/oceaneng

AUV homing and docking for remote operations

N. Palomeras *, G. Vallicrosa, A. Mallios, J. Bosch, E. Vidal, N. Hurtos, M. Carreras, P. Ridao

Underwater Robotics Research Center (CIRS), Computer Vision and Robotics Institute (VICOROB), Universitat de Girona, 17004 Girona, Spain

ARTICLE INFO

Keywords:
AUV
Docking station
Acoustic localization
Visual tracking
Remote operation

ABSTRACT

One of the major goals of the SUNRISE FP7 project is to make the Underwater Internet of Things a reality. In this context, the LOON-DOCK project presented here extends the existing Litoral Ocean Observatory Network testbed with a Docking Station tailored to the Sparus II AUV. The docking system allows a remote user to program survey-like missions through a web-based interface as well as to retrieve the data gathered by the AUV once a mission finalizes. To enable the autonomous docking of the AUV, two complementary and cost-effective localization systems have been developed. The first one implements a range-only localization algorithm to approach the docking station while the second, based on active light beacons, provides high accuracy at short ranges to complete the docking maneuver. The system has been extensively tested, in different trials from a controlled water tank environment to more realistic sea operation conditions proving its viability despite very poor water visibility conditions.

1. Introduction

The increased number of deployed subsea systems (infrastructures, sensors, robots, gliders and others) during the last years, is raising the need for interconnection among themselves and to the exterior world, for a better management and exploitation. This is one of the major goals of the SUNRISE FP7 project (Petrioli et al., 2013), which is devoted to make the Underwater Internet of Things (UIoT) a reality. Connecting the underwater systems to the network and endowing them with the capability of making their data widely accessible while minimizing the need of human interaction, has the potential of providing ocean data at an unprecedented scale.

Persistent deployment of buoyancy-driven vehicles (gliders) in open waters has already been achieved for periods of time spanning months (Manley and Willcox, 2010; Meyer, 2016). However, persistent deployment of survey-type Autonomous Underwater Vehicles (AUVs), capable of more complex mapping missions, is a challenging problem that arises the requirement of docking. The objective behind the docking concept is to extend the deployment time by installing a Docking Station (DS) that allows to extract the data of finished missions, program new missions and recharge the batteries of the vehicle without recovering it at the surface. The first systems were designed for oceanographic sampling purposes (Curtin et al., 1993; Singh et al., 2001). More recently, the interest has grown towards using AUVs in commercial scenarios, for the periodic inspection and maintenance of subsea installations (Brignone et al., 2007; Krupinski et al., 2008; Jacobson et al., 2013). In addition, a significant part of docking-related works have emerged from operational environments where launch and recovery is a difficult task such as under ice operations (King et al., 2009). However, despite many demonstrations have taken place since the 90's, the combined need of infrastructure (physical mounting, power, communications) and the demanding vehicle reliability required to operate continuously without human servicing, still makes docking a state-of-the-art problem (Bellingham, 2016).

In the context of the SUNRISE project, the LOON-DOCK project presented here aims to extend the existing Litoral Ocean Observatory Network (LOON) testbed (Alves et al., 2014) with a DS to demonstrate data transmission from a survey-AUV to the Internet. The idea is to be able to remotely operate the AUV from the SUNRISE GATE, (Petrioli et al., 2014), a web interface created to remotely schedule experiments using assets persistently deployed in different testbed facilities around the world. For the AUVs to really become part of the UIoT, it is essential to provide docking solutions that are cost-effective, while maintaining the performance and reliability of the system. In this sense, our proposal strives to keep the implemented docking approach low-cost in all the addressed aspects, namely the mechanical design, the communications and the autonomous docking procedure. The power transference to the AUV is not covered in the scope of this work. However, it is worth noting that solutions for wireless inductive power transference already exist in the market and could be integrated at a certain cost.

From the hardware point of view, the developed DS is based on a

* Corresponding author.
E-mail address: npalomer@eia.udg.edu (N. Palomeras).

https://doi.org/10.1016/j.oceaneng.2018.01.114
Received 8 May 2017; Received in revised form 21 November 2017; Accepted 31 January 2018
Available online 12 February 2018

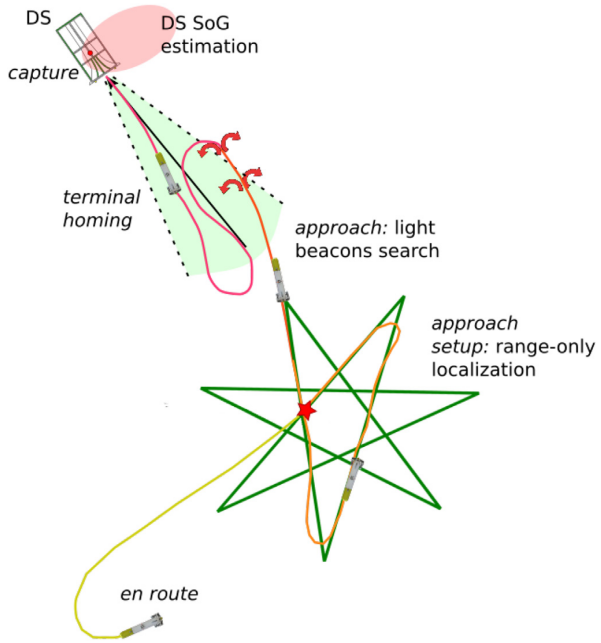


Fig. 1. Docking phases scheme.

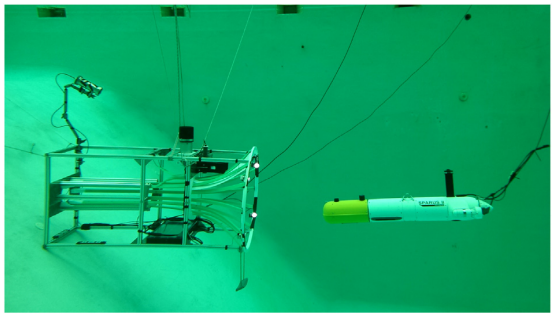


Fig. 2. Testing the docking station and the approach procedures in a water tank. The AUV cable visible in the image is only for monitoring purposes.

funnel-shaped receptacle, which is a traditional solution (Cowen et al., 1997; Allen et al., 2006; McEwen et al., 2008) to physically guide the vehicle into the dock. Another common design found in the literature are the vertical dock poles (Singh et al., 2001; Stone et al., 2010) that allow the vehicle to approach from any direction, making them more robust to changing water disturbances. To alleviate the directionality constrain, the funnel receptacle has been designed so that it can pivot over the static base and be oriented according to the water currents of the environment. Compared to a pole dock, the funnel-shaped design eases the installation of the necessary equipment to support the communications and the homing maneuvers, facilitates the establishment of links with the vehicle and protects the AUV against environmental hazards while docked. With regards to communications, the developed docking system is directly connected to the Internet through the LOON infrastructure, and features two modes of communication with the AUV. The first mode uses an acoustic modem and the SUNSET protocol (Petrioli et al., 2015), a lightweight networking framework well-suited for underwater acoustic communications. The second mode, intended for when the AUV is docked, is based on a contactless WiFi module installed in the DS, which is crucial to transfer at high speed the vast amounts of data that a survey-AUV can gather in a mission. The use of a contactless radio frequency module, based on a commercial off-the-shelf (COTS) WiFi modem, provides a reliable and low-cost data transmission system (McEwen et al., 2008) capable of achieving transfer speeds in the order of tens of Mbps. It does not require the mating of connectors underwater, thus preventing the exposure of electrical contacts, and provides a less sensitive alignment with respect to establishing direct electrical connections (Stokey et al., 2001) or inductive coupling (Feezor et al., 2001).

To assist the autonomous docking procedure, the DS employs the acoustic modem as a range-only transponder for mid-range homing, and a set of light beacons installed in the funnel entrance for the terminal phase of the docking. The autonomous docking is tackled from the general scenario in which the vehicle does not know the exact position of the DS, usually because it has lost the communication with the DS during a mission execution and only has an *a priori* coarse estimate of its position. Then, following the nomenclature established in Bellingham (2016), our proposed approach for autonomous docking works as follows (see Fig. 1): first, the vehicle starts from an *en route* phase, that gets the vehicle close enough so it can sense the DS with the on-board acoustic transponder. This navigation is performed according to the on-board navigation filter that merges information from different sensors to navigate relative to the Earth. Next, in the *approach-setup* phase, a range-only localization filter is used to estimate the DS location while the AUV is guided along an observable trajectory. Once an estimation of the DS position becomes known, the vehicle approaches the DS to bring it within visual reach (i.e., *approach* phase). The light beacon navigation system is used to estimate the DS pose with respect to the AUV on-board camera. Visual information is used to update a single landmark simultaneous localization and mapping filter that provides the relative position between the AUV and the DS with the accuracy required for the *terminal homing* phase. Preliminary versions of these two localization algorithms have been previously reported in a simulation environment (Vallicrosa et al., 2016). In order to back up the visual localization during the very last few meters of the approach, when the light beacons are no longer inside the camera field of view, a complementary system has been integrated using augmented reality (AR) markers (Garrido-Jurado et al., 2014). After this phase, the AUV ends up inside the docking funnel and is guided up to the latch mechanism by controlling the generated forward thrust (i.e., *capture* phase).

Existing solutions to perform autonomous docking usually rely in more complex acoustic sensors including USBL (McEwen et al., 2008; Allen et al., 2006) or inverted SBL setups (Smith and Kronen, 1997). Optical sensors have also been explored for the terminal phase, with solutions comprising either active light sources -with single (Cowen et al., 1997; Murarka et al., 2009; Li et al., 2015, 2016) or multi-light systems (Hong et al., 2003; Park et al., 2009)- or passive patterns and markers (Kushnerik et al., 2009; Maire et al., 2009) that can be detected by on-board cameras. Altogether, our approach provides an hierarchical homing procedure that ensures a reliable approach and terminal homing maneuvers enabled with low-cost equipment and minimal requirements on both the vehicle and the DS sides. Notice that, indeed, the acoustic ranging and optical image acquisition are capabilities that either already exist in most AUVs or can be easily added at a reasonable cost. The same reasoning applies to the docking station, since the light beacon system and the AR markers are relatively inexpensive to manufacture.

The reminder of this paper is organized as follows: next section describes in more detail the employed AUV and the design of the proposed docking station. Section 3 introduces the navigation system of the AUV together with the developed algorithms for autonomous docking, including both the range-only localization and the visual localization of the DS. Section 4 covers the insights about operation and control, involving the remote operation through the SUNRISE web-based

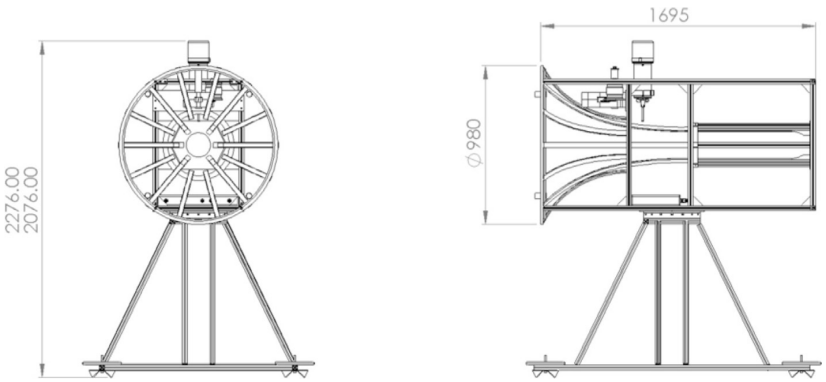


Fig. 3. Docking station CAD model (dimensions in mm).

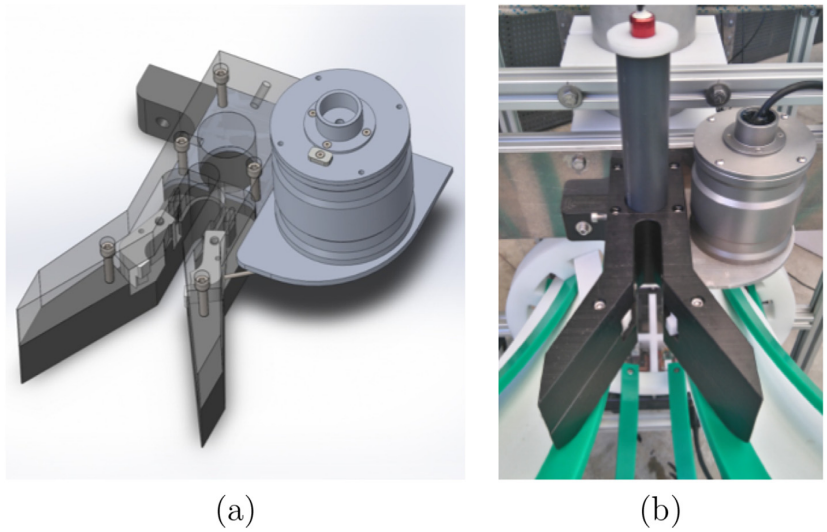


Fig. 4. (a) CAD model of the latching mechanism and (b) picture of the real latching mechanism.

interface and the AUV guidance module that controls the various phases of the autonomous docking. Section 5 presents the experiments that have been performed together with an analysis of the results obtained with the system fully integrated in the LOON infrastructure. Finally, Section 6 reports the conclusions and points out future work directions.

2. Hardware

The DS has been designed especially for the Sparus II AUV (Carreras et al., 2013) (see Fig. 2). Sparus II AUV is a lightweight (i.e., 50 kg) torpedo-shaped vehicle with partial hovering capabilities. It has two horizontal and one vertical thrusters, allowing surge, heave and yaw movements, and a mission-specific payload area. Its flexibility and easy operation makes the Sparus II AUV a multipurpose platform adaptable to a high variety of mapping applications in shallow water (i.e., up to 200 m).

The DS provides active and passive guidance for docking, a latching mechanism to maintain the vehicle docked while in standby mode, high bandwidth data communications, and visual feedback.

The DS has been designed to be as small as possible and lightweight for easy transportation, deployment, and recovery. It is constructed mainly from corrosion resistance aluminium and polyoxymethylene (POM). The structure consists of two main parts, the docking part and the base (see Fig. 3). The entrance of the docking part has fourteen funnel-shaped rails that passively guide the vehicle to the dock position using vehicle's thrust. The size of the funnel can handle translation misalignment of 40 cm and rails arrangement can correct vehicle misalignment's in roll and pitch up to 30deg. Rail guides are made from flexible POM which can absorb collisions, thus minimizing vehicle back bouncing.

The base connects with the docking part with a single axis rotation mechanism that allows the docking part to align with water currents without moving the base. Current implementation requires manual input for the rotation (i.e., by using a diver), however, in the future an active automatic mechanism will be implemented. The base feet can be adjusted up to 20 cm in height in order to level the DS and accommodate ballast weights to keep the centre of gravity low.

A latching system (see Fig. 4) has been developed to prevent the vehicle from exiting the DS due to water currents thus allowing it to enter in low power mode once docked. When the vehicle has entered the DS in its final position, two claws grip on the antenna does not allow the AUV to exit. The operation of the claws is controlled from a servo motor. There is a solid join between the motor and the right claw and from this the motion is propagated to the left claw via flexible wires. The servo motor, when activated, has enough torque to withstand opening forces if the

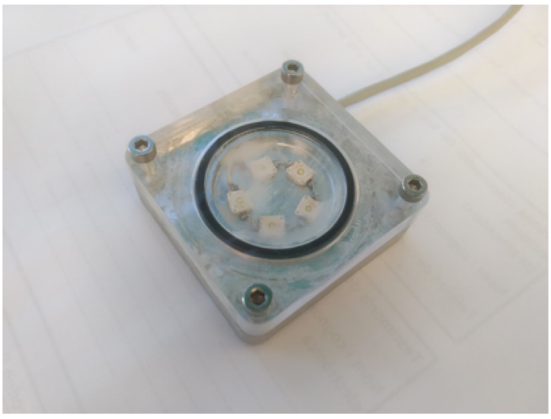


Fig. 5. Light beacon hardware design.

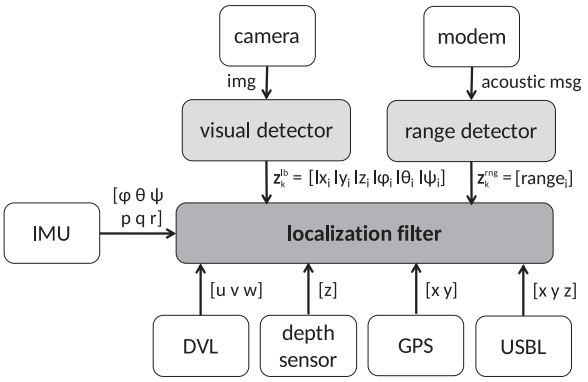


Fig. 6. Set of sensors that take part in the navigation algorithm of the AUV software architecture.

vehicle tries to exit due to water currents or even with its thrusters. However, if it is deactivated, such in the event of a DS power loss, the claws are disengaged and with relatively small thrust the AUV can safely exit.

Two active systems provide mid and short range localization information to the AUV in order to guide it to the DS using the algorithms that will be described in Section 3.

For mid range detection, an Evologics acoustic modem is installed on the top of the DS, performing two main functions: a transponder function providing range measurements for homing and a modem function to monitor the AUV during survey operations and to allow the user to send basic commands to the AUV remotely.

For short range guidance, when the DS is within the AUV visual range of its camera, light beacons are used to provide localization aid. There are four beacons installed around the entrance ring in a symmetrical fashion.

Each individual light beacon consists of five high-intensity LEDs to create a homogeneous lighting effect inside a waterproof housing (see Fig. 5). Each beacon provides flashing light at approximately 110deg field of view (FoV)FOV at 1 Hz period with 90% on - 10% off duty cycle. The system is operated at 24 V, and the maximum power consumption is 22 W, when all markers are lit. The beacons can be remotely activated from top-side.

Two high-bandwidth data communications interfaces have been included in the DS: one between the top-side and the DS, and one between the DS and the AUV.

It is very common to install the DS in long distance from the top-side base. Establishing communications between them can be achieved either by wireless connection via a surface buoy or by fixed wire. Given that in the area where the system was demonstrated (i.e., the LOON testbed) there are restrictions for installing permanent surface buoys, we chose to implement a wired connection with a long cable laying on the seafloor. Nevertheless, the DS can be easily modified to support wireless connection.

For the wired communication scheme we are using VDSL2 technology. It supports, via a single twisted pair, up to 100Mbit/s downstream rates at 500 m and graduate decreasing to 1-4Mbit/s over distances of 4–5 km. We have also included an Ethernet connection, though it can only be reliably used for distances shorter than 100 m.

For the AUV to DS communication a WiFi modem has been adapted to fit in a waterproof housing and is positioned on the latching mechanism. When the AUV is docked, its WiFi antenna is close enough to the modem ($\approx$ 7 cm) to allow a strong wireless data link. With this system we are able to communicate with up to 45Mbit/s between the AUV and the top-side.

On the DS there is a watertight container that houses all the necessary electronic components for its operation. That includes a pressure sensor, an inertial measurement unit (IMU) and an Ethernet switch connecting the Evologics modem, the WiFi modem, and a Raspberry Pi 3 single board computer. The Raspberry Pi 3 is locally running ROS (Quigley et al., 2009) and provides control for the latch mechanism and the LEDs. It is also in charge to broadcast through the modem the DS depth and orientations, obtained from the IMU, in order to check that the DS has been properly leveled during its installation. Besides, there is also a webcam that points to the center of DS imaging the AUV when it is docked. There is also an independent underwater analog video camera that is placed on the side of the DS, providing a general overview of the DS and the AUV when approaching.

The DS is connected to top-side via two independent cables, one for power and one for data, to minimize electromagnetic interferences. It is powered with 220VAC via an isolation transformer in order to minimize power loss. The data cable is an Ethernet CAT 5e cable, where one pair is used for the VDSL2, one for the analog video, and the last four are used for the short range Ethernet link.

3. AUV navigation and DS localization

The navigation filter of the Sparus II AUV is based on the well known extended Kalman filter (EKF). It combines the information on depth returned by the pressure sensor, velocities from the Doppler velocity log (DVL) and attitude from the attitude and heading reference system (AHRS) to provide a dead reckoning (DR) navigation. This navigation drifts over time and needs absolute measurements to correct it. Those measurements can come from either global positioning system (GPS) when on surface, ultra-short base-line (USBL), or acoustic ranges or visual detections with respect to a known landmark (see Fig. 6).

A feature-based EKF-SLAM navigation filter is used in the proposed scenario where the only landmark will be the DS. The state vector for the implemented filter is the following:

$$\boldsymbol{x} = [\, x \quad y \quad z \quad u \quad v \quad w \quad \boldsymbol{l}_1 \quad \ldots \quad \boldsymbol{l}_N \,], \tag{1}$$

where $[x\ y\ z]$ and $[u\ v\ w]$ are the position and linear velocity vectors of the AUV, and $\boldsymbol{l}_i$ is the landmark i pose vector defined as:

$$\boldsymbol{l}_i = [\, lx_i \quad ly_i \quad lz_i \quad l\phi_i \quad l\theta_i \quad l\psi_i \,]. \tag{2}$$

Because range-only measurements cannot estimate the landmark orientation, a landmark must be initialized by a visual detection. However, once in the state vector, both range-only and visual-based updates can be applied.

The navigation filter uses a constant velocity model with attitude input:

$$\boldsymbol{z}_k^{rng} = \left[\sqrt{rng^2 - (z_V - z_{DS})^2}\right]$$

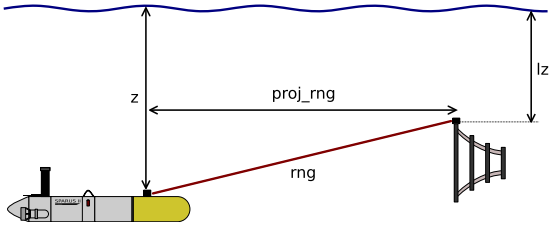


Fig. 7. Projection of the range measurement.

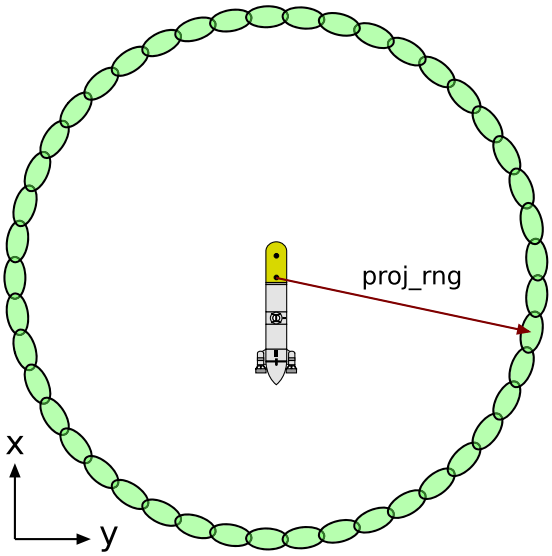


Fig. 8. Initialization of the SOG filter (2σ bounds).

$$\widehat{\boldsymbol{x}}_k = \begin{bmatrix} \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ z_{k-1} \end{bmatrix} + \mathscr{R}(\phi_k\theta_k\psi_k)\left(\begin{bmatrix} u_{k-1} \\ v_{k-1} \\ w_{k-1} \end{bmatrix} t + \begin{bmatrix} n_{u_{k-1}} \\ n_{v_{k-1}} \\ n_{w_{k-1}} \end{bmatrix} \frac{t^2}{2}\right) \\ u_{k-1} + n_{u_{k-1}} \\ v_{k-1} + n_{v_{k-1}} \\ w_{k-1} + n_{w_{k-1}} \\ l_{1_{k-1}} \\ \dots \\ l_{N_{k-1}} \end{bmatrix}, \tag{3}$$

where t is the sample time, $[n_u\ n_v\ n_w]$ is the noise vector and $[\phi_k\ \theta_k\ \psi_k]$ are the Euler angles used as the filter input $\boldsymbol{u}_k$. Each sensor measurement is modelled as:

$$\boldsymbol{z}_k = H\widehat{\boldsymbol{x}}_k + \boldsymbol{s}_k, \tag{4}$$

where $\boldsymbol{z}_k$ is the measurement itself, H is the observation matrix that relates the state vector with the sensor measurement ($h(\widehat{\boldsymbol{x}}_k)$ if the measurement is not linear), and $\boldsymbol{s}_k$ is the sensor noise. To differentiate between range and light beacon measurements we will use $\boldsymbol{z}_k^{rng}$, $h^{rng}()$ and $\boldsymbol{z}_k^{lb}$, H^{lb} respectively.

In case of range-only measurements, the observation equation provides the expected range measurement $h^{rng}(\widehat{\boldsymbol{x}}_k)$ which is given by the norm of the difference between the vehicle and beacon positions at time k.

$$h^{rng}(\widehat{\boldsymbol{x}}_k) = ||(x, y, z) - (lx_i, ly_i, lz_i)||. \tag{5}$$

In the case of a visual detection measurements, updates are linear being

$$\boldsymbol{z}_k^{lb} = [lx_i\ ly_i\ lz_i\ l\phi_i\ l\theta_i\ l\psi_i], \tag{6}$$

and

$$H^{lb} = \begin{bmatrix} -\mathscr{R}(\phi_k\theta_k\psi_k)^T & 0_{3\times3} & \mathscr{R}(\phi_k\theta_k\psi_k)^T & 0_{3\times3} & \dots \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & I_{3\times3} & \dots \end{bmatrix}, \tag{7}$$

where $[lx_i\ ly_i\ lz_i]$ is the relative position of the landmark with respect to the vehicle, $[l\phi_i\ l\theta_i\ l\psi_i]$ is the landmark orientation with respect the inertial frame and $\mathscr{R}(\phi_k\theta_k\psi_k)$ is the vehicle orientation rotation matrix at time k.

When the position of the DS is unknown or known with a high uncertainty, the range-only localization method described in Section 3.1 is used to obtain an approximate location. Once the vehicle homes to the vicinity of the DS, the light beacon detection system will be used to initialize the landmark in the EKF-SLAM navigation filter and to provide accurate updates (see Section 3.2).

3.1. DS range-only localization

Range-only localization is a highly non-linear problem. Given 1D measurements (range), the vehicle must be localized in a higher dimensional space (3D). With an unknown position of the beacon and after the first measurement, the probability distribution has the shape of a spherical shell with thickness equal to the range measurement uncertainty. Kalman filter (KF) represents a location as a Gaussian with a mean and a covariance matrix, that can be used to express a location as a sphere but not as a spherical shell. Therefore the use of KF is not appropriate for this problem.

Several range-only localization methods have been reported in the literature (Vaganay et al., 2000; Newman and Leonard, 2003; Olson et al., 2006; Webster et al., 2009; Wang et al., 2013; Blanco et al., 2008). However, those methods are demonstrated offline after the vehicle is recovered and no online localization is performed.

In order to simplify the problem from 3D localization to 2D, we project the measured ranges *rng* (8) to the horizontal plane according to the depths of the DS z_{DS} and vehicle z_V (see Fig. 7). The beacon depth is known *a priori*, since it is measured during the DS deployment or by the internal pressure sensor installed in the DS, which broadcasts measures through the acoustic modem. Likewise, AUV depth information is known very precisely from its pressure sensor, only having to take into account the tide if necessary, for which appropriate models are already available (Ray, 1999).

$$\boldsymbol{z}_k^{rng} = \left[\sqrt{rng^2 - (z_V - z_{DS})^2}\right] \tag{8}$$

During the estimation of the beacon position, we rely on the on-board DR navigation filter explained in Section 3. The DR navigation drift is not taken into account because the time needed to localize the DS is small enough, as demonstrated in Vallicrosa and Ridao (2016) where a 3D sum of Gaussian (SoG) filter with active localization (AL) was used to successfully localize an acoustic beacon in a real scenario.

At known depth, a range measurement describes a beacon as being in any position on a circumference around the AUV with a radius equal to the projected range and thickness equal to the uncertainty of the measurement. To cover this big space of possibilities one might use a particle filter (PF) to represent the static beacon position, however, the PF solution would leave empty spaces without coverage. To avoid that, a much

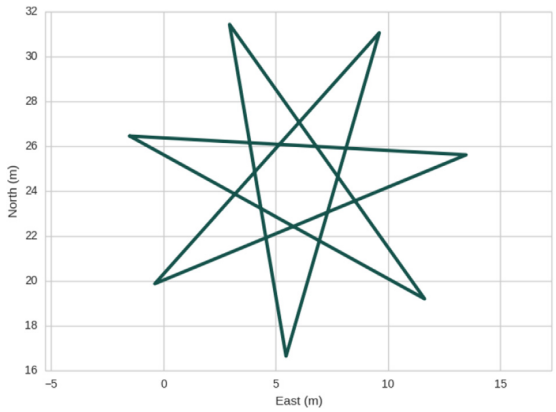


Fig. 9. Star shaped trajectory used for beacon localization.

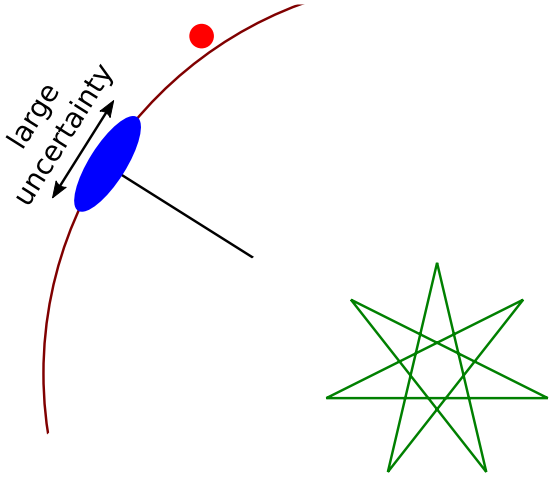


Fig. 10. After performing the star-shaped trajectory (green), the beacon is localized (blue) with big uncertainty on the axis tangent to the circumference described by the range measurements (dark red). An extra waypoint (red) is commanded to the vehicle to correct this uncertainty. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

larger number of particles could be used, but then the problem would become computationally intractable. A more elaborate option is the use of a SOG filter (Blanco et al., 2008), which is similar to the PF but uses weighted Gaussians instead of weighted particles. It represents the believed beacon position $\mathscr{B}$ according to the odometry $\boldsymbol{x}_k$ and the measurements $\boldsymbol{z}_k^{rng}$:

$$p(\mathscr{B}|\boldsymbol{x}_k, \boldsymbol{z}_k^{rng}) \approx \sum_{i=1}^{N} v_k^i \mathscr{N}(\boldsymbol{z}_k^{rng}; \boldsymbol{\mu}_k^i, \Sigma_k^i), \tag{9}$$

where v_k^i is the weight associated with each Gaussian, and $\boldsymbol{\mu}_k^i$ and Σ_k^i its mean and covariance matrix.

The Gaussians in the SOG can cover all the probability space if they are correctly distributed. Moreover, an EKF is used to correct their position according to the measurements, thus improving its performance.

The SOG is initialized with the first range measurement (see Fig. 8). The filtering is carried out in two main steps. First, the range measurement is used to update each of the Gaussians ($\boldsymbol{\mu}_k^i$, Σ_k^i) with an EKF. Second, the weights are updated with the innovation of the EKF

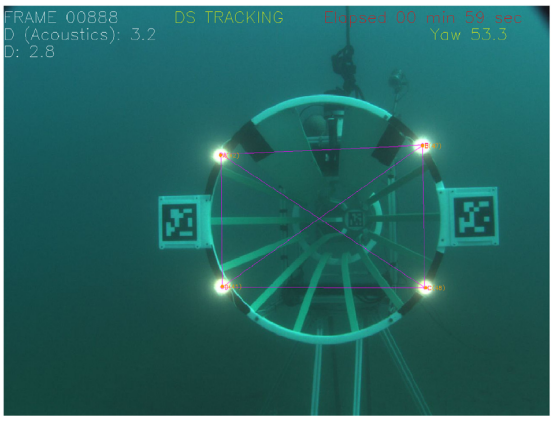


Fig. 11. Screenshot of the output while tracking the DS. In orange the position of the light markers according to the image analysis. In purple, the position of the markers in the DS according to the estimated relative position. AR markers were used as a complementary resource during the final approach when the light beacons are not in the FOV of the camera. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

$\boldsymbol{y}_k^i = \boldsymbol{z}_k^{rng} - h^{rng}(\boldsymbol{\mu}_k^i)$:

$$v_k^i = v_{k-1}^i \cdot \exp\left(-\left(\boldsymbol{y}_k^i\right)^2\right), \tag{10}$$

being i the index of the Gaussian and v_{k-1} the previous weight. This computed weight is always in the $[0, 1]$ range. The weights of the Gaussians with a small innovation are significantly greater than those having a large innovation. Thus, after some updates, the Gaussians which are not consistent with the observed ranges become negligible while those consistently compatible will influence the estimated pose of the beacon.

When the vehicle follows an observable path (Vaganay et al., 2000), the beacon is localized in a few seconds. In this work, a simple approach using a star shaped trajectory is used to avoid symmetries and locate the beacon (see Fig. 9). The trajectory is scaled proportionally to the first measured range and it is aborted as soon as the beacon is localized with an uncertainty below a user-provided threshold.

The obtained localization of the DS after the star trajectory, suffers from a large uncertainty on the tangent of the circumference defined by the range measurements (see Fig. 10). To reduce this uncertainty before attempting to approach the DS, the vehicle is commanded to a new waypoint computed on the principal axis of the localized beacon. Range updates obtained from this position further reduce the uncertainty of localization, ending with a much better localization result.

3.2. Visual pose estimation

Acoustic localization can be very effective from medium to long distances, but it is not so advantageous at short distances when high precision operation is required for successfully completing the docking maneuver. To achieve a level of performance capable of ensuring the vehicle's safety during the terminal homing, visual sensing is used to provide updates with small uncertainty and high update rates.

The proposed solution consists in placing a set of active light beacons in distinct and known positions of the DS (see Fig. 11). Using a standard camera it is possible to detect the lights in the images and estimate the pose between the DS and the camera. It is worth noting that differently from range-only localization, this method is able to provide information on the relative orientation of the DS, involving the full 6 DoFs (three relative translations and three rotations).

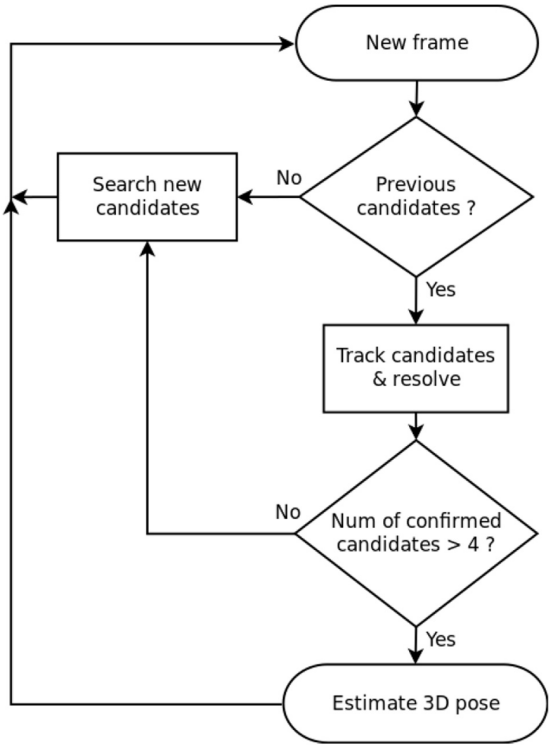


Fig. 12. Diagram of the detection algorithm. The algorithm starts by detecting spots in the images as candidate lights. These candidates are then tracked during several frames before deciding if they are accepted as a light beacon or rejected. Once there are four accepted candidates each one is associated to one of the light beacons and the relative pose of the DS is estimated.

With the aim of facilitating the detection of the light beacons and avoiding wrong identifications, the lights follow a known blinking pattern (as explained in Section 2). This pattern allows the actual lights to be correctly identified even in the presence of reflections or permanently illuminated areas in the scene.

The implemented detection technique is similar to the one detailed in Bosch et al. (2016), where we used it to track multiple AUVs for cooperative navigation in real sea conditions, and proved to be both effective and robust. It is based on three main steps: first there is a selection of candidate spots in the image based on gradient information. Then, at every incoming frame, the candidates are tracked taking into account the camera motion and we check its agreement along time with the known blinking pattern. Finally, each candidate is associated with a beacon according to the known geometry of the beacons installed in the docking station.

When the four lights have been detected (see Fig. 12), the relative landmark pose $\widehat{\boldsymbol{l}}$ of the DS that best fits the observation of the light beacons in the image, $\boldsymbol{q}_i$, is found using non-linear least squares minimization. This is done by searching for the values of the variable $\boldsymbol{l}$ that minimize the re-projection error of the beacons; that is, the difference between the real observation and the projection of the light beacon derived from the variable $\boldsymbol{l}$ and the calibration parameters of the camera.

$$\widehat{\boldsymbol{l}} = \arg\min_{\boldsymbol{l}} \sum_{i}^{n} (f_{\boldsymbol{l}}(\boldsymbol{p}_i) - \boldsymbol{q}_i)^2. \tag{11}$$

The variable $\boldsymbol{l}$ contains the complete pose of the DS with respect to the camera $\boldsymbol{l} = \begin{bmatrix} l_x^c & l_y^c & l_z^c & l_\phi^c & l_\theta^c & l_\psi^c \end{bmatrix}^T$. The function f computes the image projection of each marker i given $\boldsymbol{l}$ and the position of the marker in the DS reference frame, $\boldsymbol{p}_i$. This function uses the pinhole camera model (Zhang, 2000; Hartley and Zisserman, 2004), and assumes known intrinsic calibration parameters. Although an approximate linear solution can be found for four or more light markers using a different pose parametrization, we are interested in the above parametrization since it can be directly used in the docking problem. The problem is solved with the Levenberg-Marquardt algorithm available in the Ceres library (Agarwal et al., 2012). As with all iterative methods, it needs an initial guess of the variables, which can be approximated from the range measurements between acoustic modems. Further details on the pose estimation problem and its performance with a varying number of light beacons can be found in Gracias et al. (2015).

For the proper operation of the navigation filter it is essential to have an estimate of the pose uncertainty. A first-order approximation of the pose covariance $\Sigma_{\widehat{l}}$ can be computed from the assumed covariance Σ_q of the pixel location of the beacons in the image and from the Jacobian $J(\widehat{\boldsymbol{l}}) = \frac{\partial q}{\partial l}(\widehat{\boldsymbol{l}})$ that relates small changes in the pose parameter with small changes in the observations. The Levenberg–Marquardt implementation provides an estimate of this Jacobian at the end of the minimization. The pose covariance estimate is given by:

$$\Sigma_{\widehat{l}} = \left(J(\widehat{\boldsymbol{l}})^T \Sigma_q^{-1} J(\widehat{\boldsymbol{l}}) \right)^{-1}. \tag{12}$$

The uncertainty obtained $\Sigma_{\widehat{l}}$ excludes the uncertainty in the transformation between the camera and the AUV and the uncertainty related with the camera calibration.

In order to have an approximate value of the uncertainty of the localization of a light in the image Σ_q we assume the detected lit region on the image follows a 2D Gaussian distribution, and that its area covers the 95% of lit pixels. In other words, we find an equivalent radius for the blob as it was a perfect circle and we assume this radius is 2σ. After analysing multiple series of light beacon images gathered at different distances and lighting conditions this value has been set as constant for the range of operation of the detector $\sigma = 1$ *pixel*.

A complementary system has been developed to help the vehicle in the last few meters of the terminal homing, when due to the small distance to the DS, the light beacons are not in the field of view of the camera. This system consists in detecting augmented reality (AR) markers placed in known positions of the DS (see Fig. 11), and estimating the relative pose of the DS in a analogous way as done with the light beacons. Notice that when the vehicle is well aligned to the docking during the terminal homing, it is capable of docking reliably using only its own navigation and the last estimated pose of the DS obtained from the light beacons localization. However, in cases where there is significant navigation drift (e.g., bad DVL readings due to slopes or rocky areas), the vehicle might not present a good alignment with the dock entry therefore causing some light beacons to drop off the field of view. In such situations, the side markers become useful to locate the DS during the last meters. It is important to note that the pose estimation based on AR markers is complementary but not a substitute to the light beacon approach given that it only works when there is ambient or artificial light to make the markers visible and that the distance under which the AR markers are detectable is significantly smaller (see Section 5, Results). The detection of the AR markers was done using the ARUCO library (Garrido-Jurado et al., 2014). Given that the library does not provide any uncertainty values on the location of the detected marker corners, a conservative value of $\sigma = 2$ *pixel* was chosen. This uncertainty was then propagated in the same way as in the case of the light beacons to obtain $\Sigma_{\widehat{l}}$.

4. Vehicle operation and control

The goal of the LOON-DOCK project is to have a vehicle persistently deployed so that a user can connect to it and use it remotely to carry out

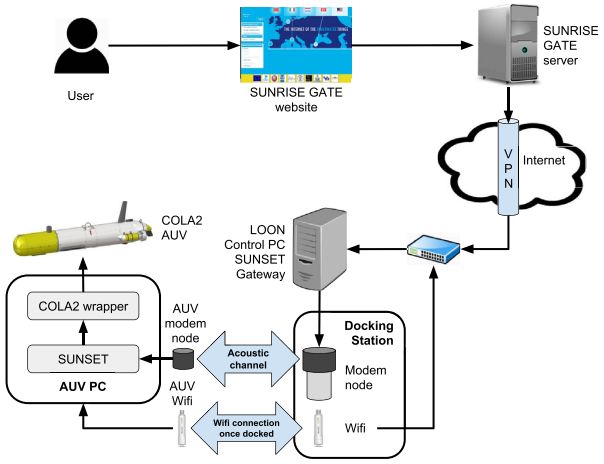


Fig. 13. Diagram of the connections to remotely operate the DS/AUV system in the LOON testbed.

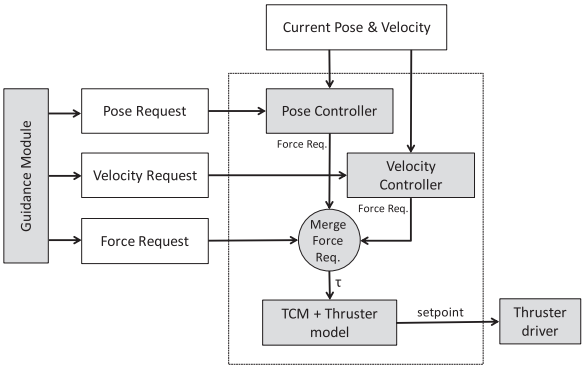


Fig. 14. Girona 500 AUV control scheme.

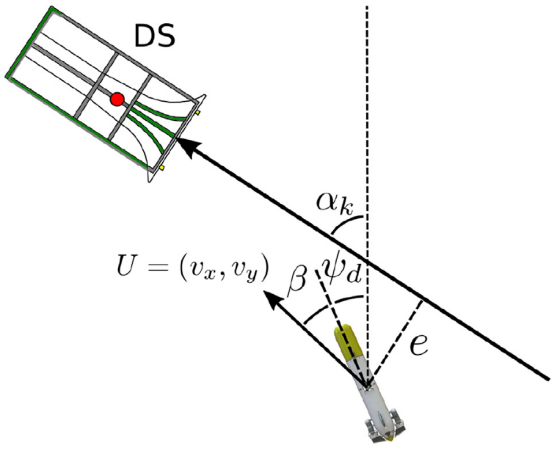


Fig. 15. Angles and distances involved in the LOS controller.

survey-like missions. Therefore, the user must be able to define, upload and delete mission plans to the AUV and start, stop or abort its execution. These plans have been defined to include the following motion commands: go to a position, keep the current position, execute a lawnmower like trajectory, and dock or undock. Moreover, the user can request the AUV position and state as well as information related with the AUV payload in real time. Acoustic modems are used to carry out all these communications, however, when the AUV is docked, the user can download the data gathered in a previous mission using the WiFi connection installed in the DS.

The following sections present the communication framework used to remotely operate the AUV and the high-level and low-level controllers used to carry out the mission plans defined by the users.

4.1. Remote operation

As shown in Fig. 13, a user can connect to the control PC in the LOON test-bed using the web-based interface named SUNRISE GATE (Petrioli et al., 2014), passing through a virtual private network (VPN) and a gateway that ensure the privacy and security of the network (Alves et al., 2014). The commands that the user executes through the web interface are received by the control PC and translated to SUNSET commands (Petrioli et al., 2015), which is the protocol used for acoustic communications. Then, they are sent through an acoustic node (i.e., the modem mounted in the DS and wired to the control PC) to another acoustic node (i.e., the modem of the AUV). Because the AUV uses a ROS-based architecture, (Quigley et al., 2009), once it receives these commands, they are translated from SUNSET to ROS. Finally, a *wrapper* module executes the necessary vehicle primitives available in the Sparus II control architecture named component oriented layer-based architecture for autonomy (COLA2) (Palomeras et al., 2012).

Both the SUNRISE GATE and the SUNSET communications were previously developed in the context of the SUNRISE project. Therefore, the main effort to allow the execution of remote commands to Sparus II AUV has been to map the SUNSET commands to the primitives in the COLA2 architecture used in the vehicle.

4.2. AUV guidance and control

To execute the plans defined by a user, the COLA2 architecture implements a high-level guidance module with the following motion commands: *keep position*, *go to waypoint*, perform a *survey-like pattern* and *dock/undock*. To achieve these motion commands the control architecture has a low-level module including a velocity (linear and angular) and a pose (position and orientation) tracking control schemes (see Fig. 14).

The objective for the velocity control scheme is to minimize the error between a commanded velocity setpoint and the actual vehicle velocity. Analogously, the pose tracking controller seeks to minimize the error between a desired pose setpoint and the actual vehicle pose. Both pose and velocity controllers generate an output force (τ_i) that is merged with other forces that can be generated by other high-level controllers (e.g., safety controllers, teleoperation, etc.). The resulting force (τ_d) must be then achieved by the vehicle thrusters. To compute the thrust that each propeller must yield, τ_d is multiplied by the inverse of the thruster control matrix (TCM), being TCM a matrix that codifies the amount of force that each thruster produces per DoF. A complete description of the implemented low-level controller module can be found in Palomeras et al. (2015) for an AUV that runs the same architecture than Sparus II AUV.

To implement the available motion commands, the guidance module uses the pose and velocity control schemes as follows:

Station Keeping: To keep the current position, the pose of the AUV in the moment to enable this motion command is sent as a desired setpoint to the pose controller.

Go to waypoint: To move the AUV to a specific waypoint, the guidance module computes the orientation error (ψ_e) between the vehicle's current position and the desired waypoint and sends the desired yaw (ψ_d) to the pose controller together with the desired depth (z_d). When, ψ_e is smaller than a defined error, the desired surge (u_d) is computed, proportionally to ψ_e and a maximum speed, and is sent to the velocity controller.

Survey like pattern: A lawn-mower pattern is generated from the user

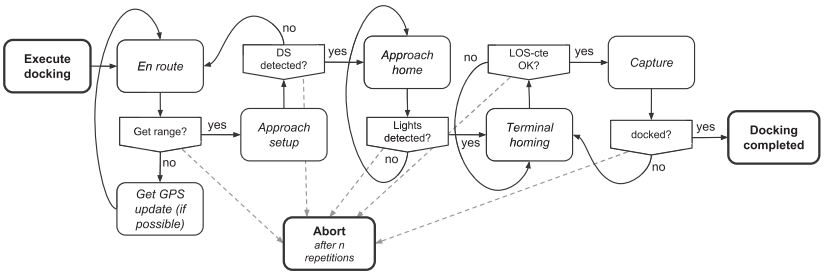


Fig. 16. State diagram for undocking, mission execution, and docking while handling more common failures.

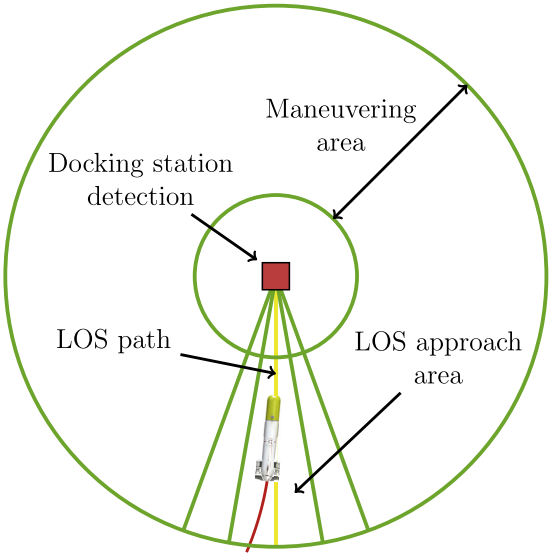


Fig. 17. Terminal homing phase.

input specifying the area to be covered (i.e., three waypoints) and the desired track spacing. The generated transects are then followed using a line-of-sight (LOS) controller with cross tracking error (LOS-cte), as the one described in Fossen (2011). A proportional controller using the cross-track error e in conjunction with the sideslip (drift) angle β (13) allows steady state convergence of the controller under constant currents (see Fig. 15). The LOS-cte controller sends the computed yaw reference ψ_d and the desired depth z_d to the pose controller while the desired constant surge velocity u_d is achieved by the velocity control scheme.

$$\psi_d = \alpha_k + arctan(-K_p\, e) - \beta \tag{13}$$

$$\beta = arctan2(v_y, v_x)$$

Docking: Finally, for the docking motion command, the guidance module executes a state machine like the one presented in Fig. 16:

i Given an estimation of the DS position and orientation, a waypoint 40 m in front of it is computed and the AUV navigates to it (*en route*). Notice that this point is only used to start sensing ranges from the modem and therefore the accurate positioning of the AUV in that point is not critical. It is set at 40 m in order to account for up to 30 m of navigation drift. In this way there is enough margin to avoid potential collisions with the DS while still ensuring that ranges would be sensed with no problem (even at shallow water, i.e., less than 15 m depth, ranges can be heard at more than 70 m distance). In the event that no ranges were detected by the modem from this point, the AUV surfaces to get a GPS update, if possible, and moves again to the computed location.

ii Once ranges are available, the acoustic localization algorithm (see Section 3.1) is enabled in order to estimate the DS position (*approach setup*).

iii Once the acoustic localization finalizes, if a candidate position for the DS is obtained, the AUV is sent several meters in front of it and a visual-based searching procedure, consisting in turning the vehicle left and right while approaching the DS up to a safety distance, is started to localize the light beacons (*approach home*).

iv If the light beacons are localized (see Section 3.2), the DS landmark is initialized in the EKF-SLAM filter and the vehicle is sent again to some distance with respect to the DS entrance to carry out the *terminal homing*. This phase can be classified as a pose-based visual servoing (PBVS) and contains two main steps (see Fig. 17): move closer to the LOS path while not losing the visual contact with the DS and follow the LOS path until reaching the DS entrance. Initially, if the vehicle is not aligned with the DS, a series of movements might be required to position the robot in an appropriate location. These maneuvers always try to keep the light beacons inside the camera FOV using the position of both the AUV and the DS estimated by the navigation filter. In the second step, the already presented LOS algorithm with cross tracking error (LOS-cte) is executed to guide the AUV up to the DS entrance. If the AUV gets out of the LOS approach area while getting closer to the DS it returns to the first step.

v If the *terminal homing* finalizes successfully, the *capture* phase starts applying a force profile to the thrusters to gently introduce the vehicle inside the DS. A similar force profile, but on the opposite direction, is applied to undock the vehicle.

If all phases conclude satisfactory, to verify that the vehicle has been correctly docked, several elements are considered: a camera in the DS pointing to the funnel, the WiFi connection between the DS and the AUV, and also a test to check that the AUV orientation is static despite applying some thrust.

If any of the phases fails, the same phase or the previous one is repeated as shown in Fig. 16. If after three repetitions the phase keeps failing, the docking command is aborted.

5. Experiments and results

All the elements described in Section 3 were tested incrementally. Preliminary tests were performed in a 16 × 8x5m water tank at the University of Girona. Even though this water tank has a rather limited space, it provides a convenient and controlled environment with a supervision room that offers direct view to the water. In there we could test the light beacons localization, the acoustic communications using SUNSET, the vehicle remote operation from the SUNRISE GATE, and the final phases of the docking maneuver (i.e., *approach home*, *terminal homing* and *capture*). Later, additional experiments were carried out in a harbour near Girona. The SOG filter and further tests on the light beacon localization system were performed at this harbour under more real conditions. Once

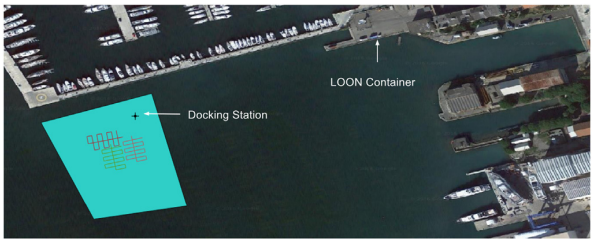


Fig. 18. View of the operational area, where the docking station was deployed and the experiments took place. All tests were controlled from a lab container located next to the LOON container.

all the phases of the docking command were checked individually, more complete tests, involving survey trajectories started and/or ended in the dock, were carried out using the acoustic communications between the DS and the AUV.

Final experiments were performed at the LOON testbed, located in La Spezia (Italy), during a three day campaign. Fig. 18 shows the location of the LOON container, where the LOON control PC running the SUNSET gateway was located. The blue area shows the region that was designated for conducting the experiments and where the DS was installed at around 9 m depth. Fifteen mission plans, all of them including docking commands, were executed by the AUV at this testbed.

In all tests the DS was correctly detected and only in three (20%) was not possible to conclude the docking maneuver due to the presence of strong water currents that affected the DS area during some hours in a direction that was not aligned with the docking station. In two of these three cases, after a predefined number of re-attempts, the system returned an error indicating that the docking could not be performed and in the remaining one, the vehicle finalized the maneuver but detected that the vehicle was not correctly docked. As described in Section 2, the DS can rotate to be aligned with the water currents. However, in the DS current state, this is a manual adjustment that has to be done by a diver. When the current was aligned with the DS axis or was relatively low (below 0.15 m/s), the docking maneuver was always successful despite very poor visibility (below 3 m most of the time).

Several tests were done to assess the remote operation of the vehicle. First, SUNSET commands were sent through the SUNSET user interface directly from the LOON control PC. Once checked that the AUV could be completely commanded by sending acoustic orders through the SUNSET protocol, the vehicle was remotely operated through the SUNRISE GATE web interface demonstrating the complete integration with the LOON test-bed (see Fig. 19). The DS capability to latch the AUV and to transmit large amounts of data once this was docked was also tested during the trials allowing to keep the AUV docked, in suspension mode, for several hours as well as to transmit previously gathered data files up to 45Mbits/s.

Fig. 20 summarizes one of the full missions performed by the AUV. The vehicle started by undocking from the DS (purple line) and went to the first waypoint of the survey trajectory. It executed the grid survey (orange line) while gathering multibeam data from 5 m altitude. At the end of it, the docking motion command was triggered. Even though the AUV had left the DS, and therefore it could receive acoustic range updates during the whole mission, we forced the AUV to localize the DS using the SOG filter with AL. Thus, the AUV started the star pattern trajectory that is performed to increase the observability of the DS ranges.

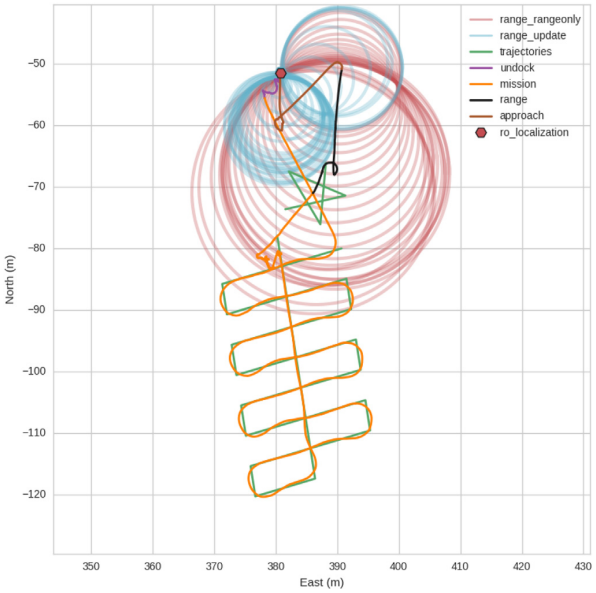


Fig. 20. Visual mission summary.

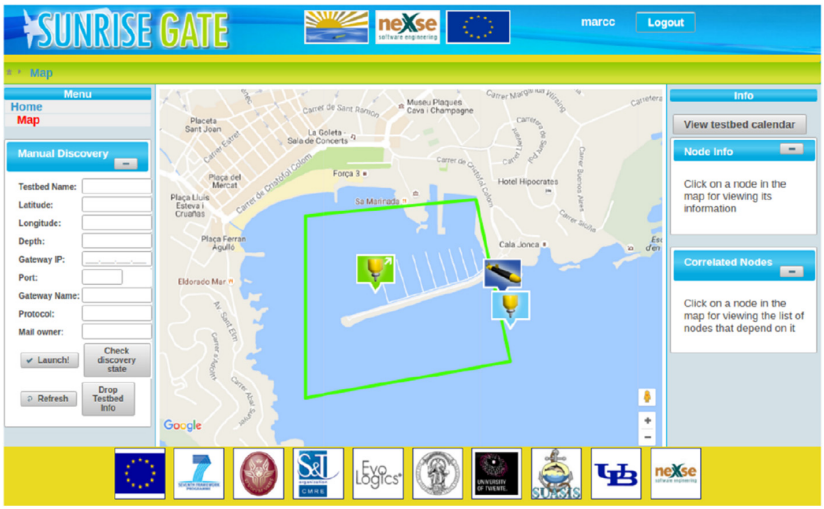


Fig. 19. SUNRISE GATE web interface screenshot.

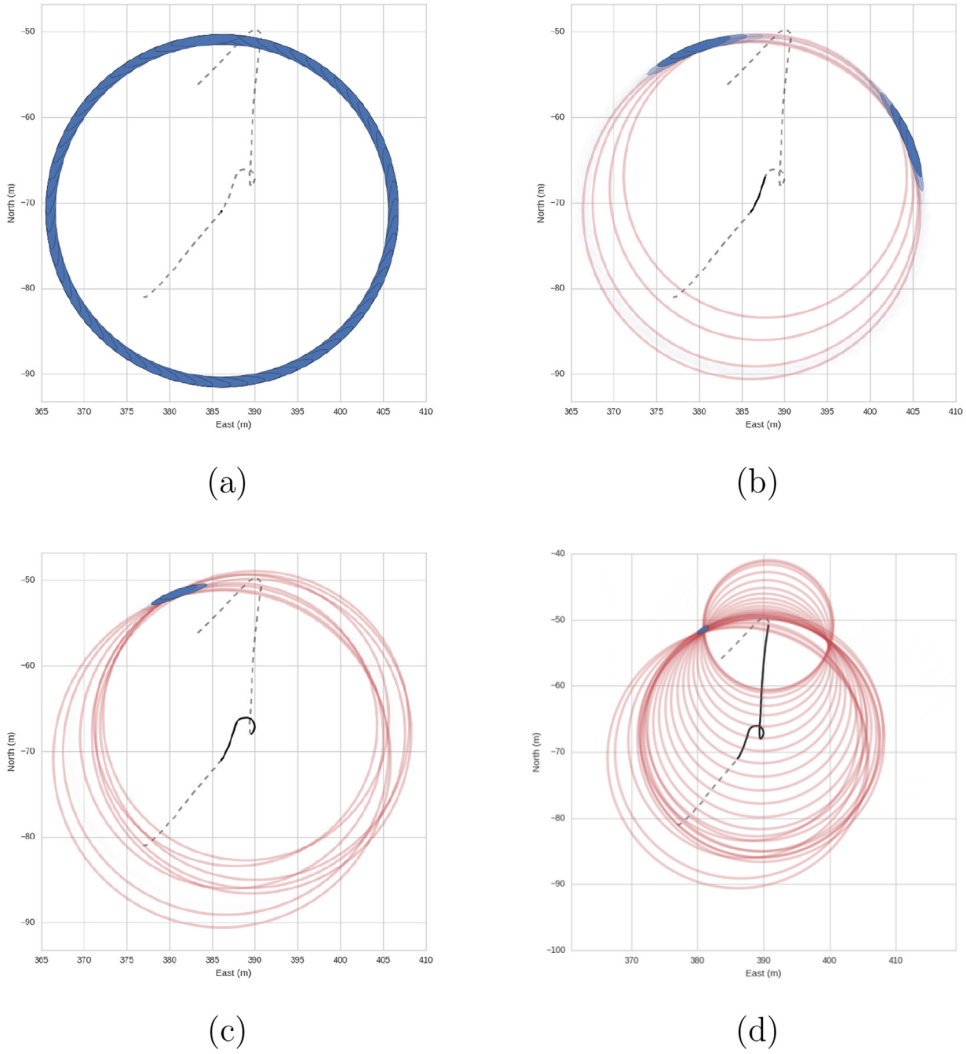


Fig. 21. Evolution of the range-only localization algorithm to detect the docking station position. See text for details.

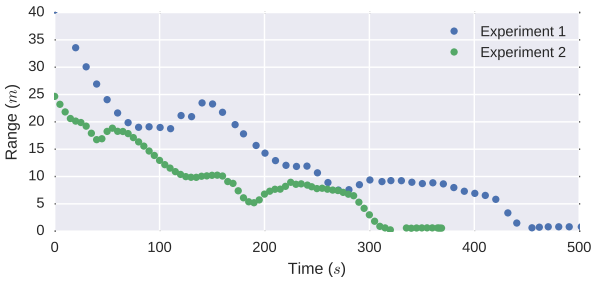
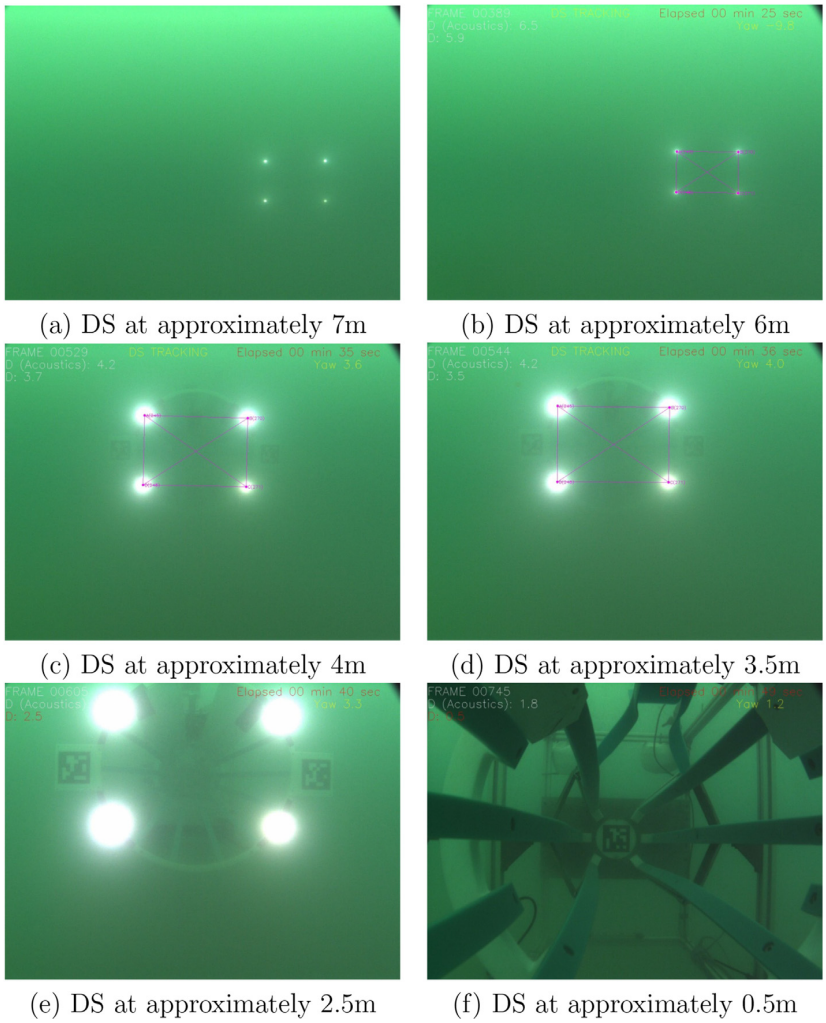


Fig. 22. Range measurements over time for two different missions.

The range-only localization process can be seen in more detail in Fig. 21. The algorithm starts by initializing a set of Gaussians that cover all the position probability space (Fig. 21a) according to the first range measurement. The vehicle initiates the star pattern trajectory and the incoming range measurements quickly narrow the probability of the DS position to two possible locations that are in agreement with all the received measurements (Fig. 21b). After performing the first turn, new incoming ranges narrow down the uncertainty of the docking station position below the established threshold ($\sigma_{th} = 1.5$). At this point, the vehicle moves to one side of the determined position in order to decrease the uncertainty of the ellipse (down to $\sigma_{DS} = 0.7$) in its main axis (Fig. 21c and d). It is worth noting that this successful localization benefited from very consistent range measurements provided through SUNSET, arriving every 5s.

Range measurements can be affected by various types of non-Gaussian noise like surface bounces, wrap returns and background noise (Yoerger et al., 2007). In the presented experiments, the measured ranges are not affected by those problems due to the small distance between the AUV and the DS and because there were no obstacles in the vicinity, leaving a direct reception for modem communication and range measurements. This correct reception can be observed in Fig. 22 where measured ranges over time, during the localization and approach to the DS, show a smooth shape without outliers. Given the favorable conditions, the range update rate was constant at 0.2 *Hz*.

In the final approach to the docking station, the optical tracking system is started in order to have a better precision on the DS location as well as a faster update rate. At this point, the docking command executes



(a) DS at approximately 7m (b) DS at approximately 6m

(c) DS at approximately 4m (d) DS at approximately 3.5m

(e) DS at approximately 2.5m (f) DS at approximately 0.5m

Fig. 23. Docking maneuver images from Sparus II AUV on-board camera. The lights were visible from at least 7 m away from the DS, while the AR markers were only visible at 3 m.

a search pattern in which the AUV approaches the estimated position of the DS from 10 m to 4 m, and every 2 m the AUV performs a sweeping movement turning from 90deg to the left to 90deg to the right. This behavior terminates when the DS is detected. With this simple search pattern, we ensure that the vision-based system will be able to detect the light beacons, even when the error in the SOG filter estimation is up to 2.8 m (i.e., 95% of cases when $\sigma = 0.7$).

The maximum distance of operation for the optical tracking depends strongly on the visibility conditions. For this particular experiment, the visibility was considered poor: the structural elements of the DS were not visible at distances greater than 3 m. However, the light beacons could be properly detected up to a maximum distance of 7 m, allowing to complete the docking approach without difficulty. Visibility conditions can be appreciated in Fig. 23.

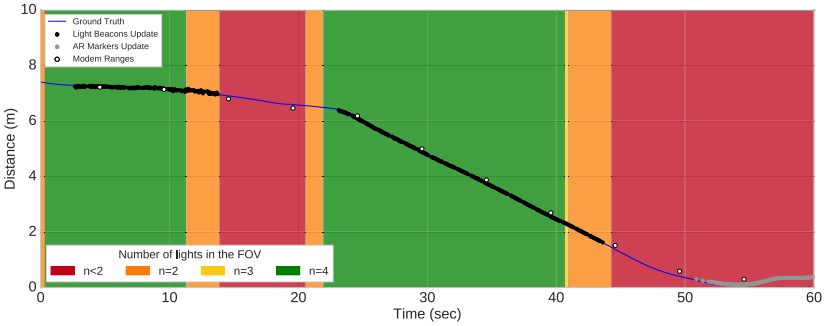


Fig. 24. DS tracking performance during the docking command execution in one of the missions. The different colored dots represent three different updates: White dots correspond to acoustic updates, black dots are visual updates computed using light beacons while grey dots are visual updates computed using AR markers. The blue line represents the relative distance between the DS and the AUV according to the navigation of the vehicle and was computed offline once the position of the DS was known precisely. The colored background reflects how many light beacons were inside the FOV of the camera according to the orientation of the vehicle. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

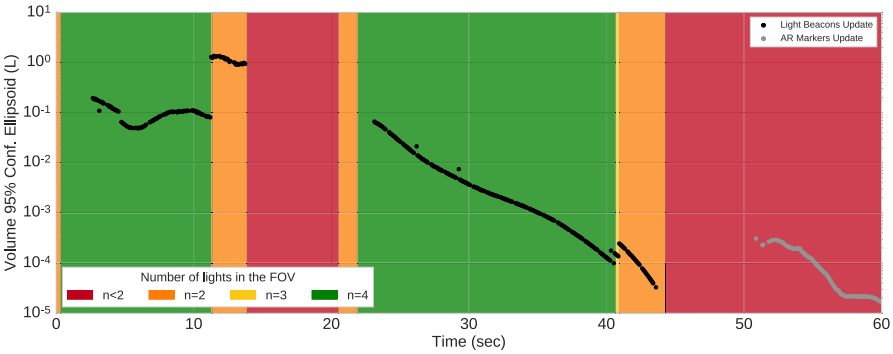


Fig. 25. Uncertainty of the optical tracking updates during the docking maneuver in one of the missions. Black dots correspond to light beacons updates while grey dots are for AR markers updates. The colored background reflects how many light beacons were inside the camera FOV according to the vehicle's orientation.

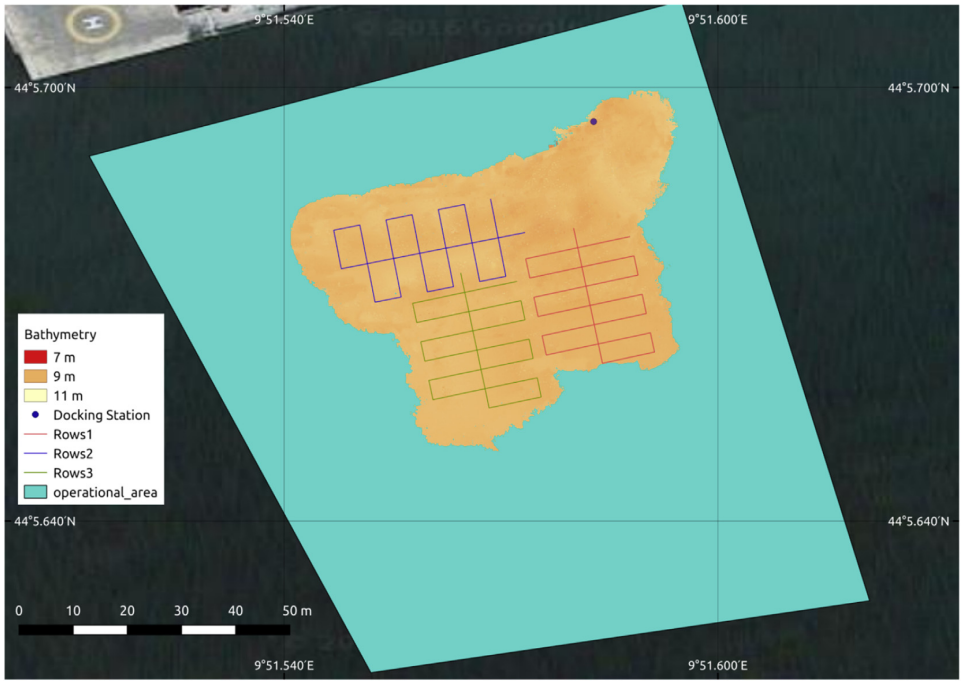


Fig. 26. Bathymetric data collected during different experiments in the LOON testbed.

Once the vision-based system is able to estimate consistently several times the DS pose, a landmark is initialized in the EKF localization system and every time that a new vision-based measure or a range measure is obtained, the position of both AUV and DS is updated in the navigation filter (see Section 3). After introducing the landmark in the EKF, the guidance module moves the AUV to a waypoint placed 10 m in front of the DS pose available in the state vector and initializes the terminal homing phase. While approaching the DS following the LOS path, if the estimated distance by the optical tracking is under 1.5 m and there is ambient light, the approach used to detect the DS changes and the algorithm tries to estimate the relative position of it using the AR markers instead of the light beacons. When the system was on tracking mode, it was able to provide updates at a maximum of 15 *Hz*, which corresponds to the capture frame rate of the camera. Because acoustic ranges, light beacons, and AR marker measurements are equally used to update the DS position in the EKF-SLAM filter, the position of these three elements with respect to the DS frame must be properly calibrated.

Fig. 24 compares the distance between the AUV and the DS estimated according to the acoustic ranges, light beacons and AR markers. All methods show a high degree of agreement with the trajectory estimated offline, once the DS position was known precisely, with differences smaller than 30 cm. The background colors in the plot represent the number of light beacons inside the FOV of the camera, this is, depending on the orientation of the AUV a different number of light beacons are inside the FOV of the camera. Due to the fact that the orientation of the DS is known *a priori*, even when only two of the light beacons have been identified it is possible to estimate the relative pose. For this particular mission the optical track was lost between seconds 15 to 25 due to a reorientation of the AUV to correctly approach the DS. From second 45 onwards, when the distance is smaller than 1.5 m, the optical tracking was based on AR markers.

From the plot, we can conclude that the optical tracking worked when the lights were in the camera FOV and in a distance of less than 7 m. The tracking was lost at 1.5 m, moment in which the tracking method changed to AR markers detection.

The uncertainty of the DS position according to the visual estimates is shown in Fig. 25. The plot includes the uncertainty in the detection of the DS in the image, but excludes the uncertainty in the transformation

between the camera and the AUV and the uncertainty related with the camera calibration. Peaks in the uncertainty plot appear when only two or three lights are visible instead of four.

Fig. 26 shows the bathymetric data collected along several survey missions that were autonomously commanded and executed in different locations. Although the data might not be very valuable in terms of mapping utility, because the area was extremely flat and with scarcity of features, it serves as a proof of concept and testifies to the whole concept of the LOON-DOCK: being able to remotely command and retrieve large amounts of data from a persistently deployed AUV through a DS.

6. Conclusions and future work

This paper has presented the docking solution implemented in the context of the LOON-DOCK project in order to demonstrate the remote operation and data transmission of a survey-AUV from Internet.

We have designed and built a docking station for Sparus II AUV, providing passive and active guidance mechanisms, a latching system, high-bandwidth data communications and visual feedback. The active guidance for docking is based on two complementary and cost-effective systems: two acoustic modems using only range information within a SOG filter to detect the DS from mid distances and a vision-based system, composed of a set of light beacons and visual markers, used in the terminal homing phase. The combination of a mid-accuracy mid-range method with a high-accuracy short-range method allows us to reliably dock the AUV.

The system has been extensively tested, with trials ranging from a controlled water tank environment to more realistic sea operation conditions. Successful validation of all the involved parts (i.e., AUV navigation, control, acoustic communications, and autonomous docking capabilities) has been conducted despite the presence of currents and very poor water visibility.

The main downside identified in our docking approach arises in the presence of strong water currents that are not aligned with the docking station. Sparus II, being a torpedo-shaped vehicle underactuated in sway, can only deal with lateral currents by changing its heading. This has two negative effects: first, by adapting the heading to compensate the currents, the vehicle is more prone to lose sight of the light beacons during the terminal homing, and second, if the misalignment with the docking is larger than 30°, the vehicle cannot physically enter inside the funnel receptacle. For this reason, as a future work, it is important to deal with the automatic alignment of the DS with the currents. Notice that the proposed design is already able to rotate the DS funnel receptacle and only a system to measure the water currents and an actuator to move it accordingly would have to be added. Besides, a battery recharging mechanism will be also integrated in the future to complete the system and enable its persistent deployment through longer time spans.

Acknowledgement

This work was supported by the EU funded project LOON-DOCK/SUNRISE (Sensing, monitoring and actuating in the UNderwater world through a federated Research InfraStructure Extending the Future Internet) FP7-ICT-2013-10-611449.

References

Agarwal, Sameer, Mierle, Keir, et al., 2012. Ceres solver. http://ceres-solver.org.

Allen, Ben, Austin, Tom, Forrester, Ned, Goldsborough, Rob, Kukulya, Amy, Packard, Greg, Purcell, Mike, Stokey, Roger, 2006. Autonomous docking demonstrations with enhanced remus technology. In: OCEANS 2006. IEEE, pp. 1–6.

Alves, J., Potter, J., Guerrini, P., Zappa, G., LePage, K., Sept 2014. The loon in 2014: test bed description. In: 2014 Underwater Communications and Networking (UComms), pp. 1–4. https://doi.org/10.1109/UComms.2014.7017141.

Bellingham, James G., 2016. Autonomous underwater vehicle docking. In: Springer Handbook of Ocean Engineering. Springer, pp. 387–406.

Blanco, J.-L., Fernandez-Madrigal, J.-A., González, Javier, 2008. Efficient probabilistic range-only slam. In: Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on. IEEE, pp. 1017–1022.

Bosch, Josep, Gracias, Nuno, Ridao, Pere, Istenič, Klemen, Ribas, David, 2016. Close-range tracking of underwater vehicles using light beacons. Sensors. ISSN: 1424-8220 16 (4), 429. http://www.mdpi.com/1424-8220/16/4/429.

Brignone, L., Perrier, M., Viala, C., 2007. A fully autonomous docking strategy for intervention auvs. In: OCEANS 2007-Europe. IEEE, pp. 1–6.

Carreras, Marc, Candela, C., Ribas, D., 2013. Sparus II, design of a lightweight hovering AUV. In: 5th International Workshop on Marine Technology (MARTECH), ISBN 2011728827, pp. 163–164.

Cowen, Steve, Briest, Susan, Dombrowski, James, 1997. Underwater docking of autonomous undersea vehicles using optical terminal guidance. In: OCEANS'97. MTS/IEEE Conference Proceedings, vol. 2. IEEE, pp. 1143–1147.

Curtin, Thomas B., Bellingham, James G., Catipovic, Josko, Webb, Doug, 1993. Autonomous oceanographic sampling networks. Oceanography 6 (3), 86–94.

Feezor, Michael D., Sorrell, F. Yates, Blankinship, Paul R., 2001. An interface system for autonomous undersea vehicles. IEEE J. Ocean. Eng. 26 (4), 522–525.

Fossen, Thor I., 2011. Handbook of Marine Craft Hydrodynamics and Motion Control. John Wiley and Sons.

Garrido-Jurado, S., Mu noz Salinas, R., Madrid-Cuevas, F.J., Marín-Jiménez, M.J., 2014. Automatic generation and detection of highly reliable fiducial markers under occlusion. Pattern Recogn. ISSN: 0031-3203 47 (6), 2280–2292. https://doi.org/10.1016/j.patcog.2014.01.005. http://www.sciencedirect.com/science/article/pii/S0031320314000235.

Gracias, Nuno, Bosch, Josep, Karim, Mohammad Ehsanul, 2015. Pose estimation for underwater vehicles using light beacons. In: Proc. Of the 4th IFAC Workshop on Navigation, Guidance and Control of Underwater Vehicles NGCUV 2015, vol. 48, pp. 70–75.

Hartley, R.I., Zisserman, A., 2004. Multiple View Geometry in Computer Vision, second ed. Cambridge University Press, ISBN 0521540518.

Hong, Young-Hwa, Kim, Jung-Yup, Oh, Jun-ho, Lee, Pan-Mook, Jeon, Bong-Hwan, Oh, Kyu-Hyun, et al., 2003. Development of the homing and docking algorithm for auv. In: The Thirteenth International Offshore and Polar Engineering Conference. International Society of Offshore and Polar Engineers.

Jacobson, John, Cohen, Pierce, Nasr, Amin, Schroeder, Art J., Kusinski, Greg, 2013. Deepstar 11304: laying the groundwork for auv standards for deepwater fields. Mar. Technol. Soc. J. 47 (3), 13–18.

King, Peter, Lewis, Ron, Mouland, Darrell, Walker, Dan, 2009. Catchy an auv ice dock. In: OCEANS 2009, MTS/IEEE Biloxi-Marine Technology for Our Future: Global and Local Challenges. IEEE, pp. 1–6.

Krupinski, Szymon, Maurelli, Francesco, Grenon, Gabriel, Petillot, Yvan, 2008. Investigation of autonomous docking strategies for robotic operation on intervention panels. In: OCEANS 2008. IEEE, pp. 1–10.

Kushnerik, A.A., Vorontsov, A.V., Ph Scherbatyuk, A., 2009. Small auv docking algorithms near dock unit based on visual data. In: OCEANS 2009, MTS/IEEE Biloxi-Marine Technology for Our Future: Global and Local Challenges. IEEE, pp. 1–6.

Li, Dejun, Chen, Yan hu, Shi, Jian guang, Yang, Can jun, 2015. Autonomous underwater vehicle docking system for cabled ocean observatory network. Ocean Eng. ISSN: 0029-8018 109 (Suppl. C), 127–134. https://doi.org/10.1016/j.oceaneng.2015.08.029. http://www.sciencedirect.com/science/article/pii/S0029801815004278.

Li, Dejun, Zhang, Tao, Yang, Canjun, 2016. Terminal underwater docking of an autonomous underwater vehicle using one camera and one light. Mar. Technol. Soc. J. 50, 58–68, 11.

Maire, Frederic D., Prasser, David, Dunbabin, Matthew, Dawson, Megan, 2009. A vision based target detection system for docking of an autonomous underwater vehicle. In: Proceedings of the 2009 Australasian Conference on Robotics and Automation. Australian Robotics and Automation Association.

Manley, Justin, Willcox, Scott, 2010. The wave glider: a persistent platform for ocean science. In: OCEANS 2010 IEEE-Sydney. IEEE, pp. 1–5.

McEwen, Robert S., Hobson, Brett W., McBride, Lance, Bellingham, James G., 2008. Docking control system for a 54-cm-diameter (21-in) auv. IEEE J. Ocean. Eng. 33 (4), 550–562.

Meyer, D., 2016. Glider technology for ocean observations: a review. Ocean Sci. Discuss. 2016, 1–26. https://doi.org/10.5194/os-2016-40. http://www.ocean-sci-discuss.net/os-2016-40/.

Murarka, Aniket, Kuhlmann, Gregory, Gulati, Shilpa, Flesher, Chris, Sridharan, Mohan, Stone, William C., 2009. Vision-based frozen surface egress: a docking algorithm for the endurance AUV. In: Proceedings of the Unmanned Untethered Submersible Technology Conference (UUST), vol. 2009.

Newman, Paul, Leonard, John, 2003. Pure range-only sub-sea SLAM. In: Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on, vol. 2. IEEE, pp. 1921–1926.

Olson, Edwin, Leonard, John J., Teller, Seth, 2006. Robust range-only beacon localization. Oceanic Eng. IEEE J. 31 (4), 949–958.

Palomeras, N., El-Fakdi, A., Carreras, M., Ridao, P., Oct 2012. Cola2: a control architecture for auvs. IEEE J. Ocean. Eng. ISSN: 0364-9059 37 (4), 695–716. https://doi.org/10.1109/JOE.2012.2205638.

Palomeras, Narcís, Carrera, Arnau, Hurtós, Natàlia, Karras, George C., Bechlioulis, Charalampos P., Cashmore, Michael, Magazzeni, Daniele, Long, Derek, Fox, Maria, Kyriakopoulos, Kostas J., et al., 2015. Toward persistent autonomous intervention in a subsea panel. Aut. Robots 1–28.

Park, Jin-Yeong, Jun, Bong huan, Lee, Pan mook, Oh, Junho, 2009. Experiments on vision guided docking of an autonomous underwater vehicle using one camera. Ocean Eng. ISSN: 0029-8018 36 (1), 48–61. https://doi.org/10.1016/j.oceaneng.2008.10.001. http://www.sciencedirect.com/science/article/pii/S0029801808002242. Autonomous Underwater Vehicles.

Petrioli, C., Potter, J., Petroccia, R., November, 17 2013. Sunrise "sensing, monitoring and actuating on the underwater world through a federated research infrastructure extending the future internet. In: Proceedings of EMSO 2013. Rome, Italy.

Petrioli, C., Petroccia, R., Spaccini, D., Vitaletti, A., Arzilli, T., Lamanna, D., Galizial, A., Renzi, E., Sept 2014. The sunrise gate: accessing the sunrise federation of facilities to test solutions for the internet of underwater things. In: 2014 Underwater Communications and Networking (UComms), pp. 1–4. https://doi.org/10.1109/UComms.2014.7017144.

Petrioli, Chiara, Petroccia, Roberto, Potter, John R., Spaccini, Daniele, 2015. The sunset framework for simulation, emulation and at-sea testing of underwater wireless sensor networks. Ad Hoc Netw. 34, 224–238.

Quigley, Morgan, Conley, Ken, Gerkey, Brian P., Faust, Josh, Foote, Tully, Leibs, Jeremy, Wheeler, Rob, Ng, Andrew Y., 2009. Ros: an open-source robot operating system. In: ICRA Workshop on Open Source Software.

Ray, Richard D., 1999. A Global Ocean Tide Model from Topex/poseidon Altimetry: Got99.2. Technical Report. NASA Technical Reports Server (NTRS).

Singh, Hanumant, Bellingham, James G., Hover, Franz, Lemer, S., Moran, Bradley A., Von der Heydt, Keith, Yoerger, Dana, 2001. Docking for an autonomous ocean sampling network. IEEE J. Ocean. Eng. 26 (4), 498–514.

Smith, S.M., Kronen, D., 1997. Experimental results of an inexpensive short baseline acoustic positioning system for auv navigation. In: OCEANS'97. MTS/IEEE Conference Proceedings, vol. 1. IEEE, pp. 714–720.

Stokey, Roger, Allen, Ben, Austin, Tom, Goldsborough, Rob, Forrester, Ned, Purcell, Mike, Alt, Chris Von, 2001. Enabling technologies for remus docking: an integral component of an autonomous ocean-sampling network. IEEE J. Ocean. Eng. 26 (4), 487–497.

Stone, W., Hogan, B., Flesher, C., Gulati, S., Richmond, K., Murarka, A., Kuhlman, G., Sridharan, M., Siegel, V., Price, R.M., et al., 2010. Design and deployment of a four-degrees-of-freedom hovering autonomous underwater vehicle for sub-ice exploration and mapping. Proc. Inst. Mech. Eng. Part M J. Eng. Marit. Environ. 224 (4), 341–361.

Vaganay, J., Baccou, P., Jouvencel, B., 2000. Homing by acoustic ranging to a single beacon. In: OCEANS 2000 MTS/IEEE Conference and Exhibition, vol. 2. IEEE, pp. 1457–1462.

Vallicrosa, G., Bosch, J., Palomeras, N., Ridao, P., Carreras, M., Gracias, N., 2016. Autonomous Homing and Docking for Auvs Using Range-only Localization and Light Beacons, vol. 49. IFAC-PapersOnLine, pp. 54–60 (23).

Vallicrosa, Guillem, Ridao, Pere, 2016. Sum of Gaussian single beacon range-only localization for auv homing. Annu. Rev. Contr. 42, 177–187.

Wang, Sen, Chen, Ling, Hu, Huosheng, Gu, Dongbing, 2013. Single beacon based localization of AUVs using moving Horizon estimation. In: Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on. IEEE, pp. 885–890.

Webster, Sarah E., Eustice, Ryan M., Singh, Hanumant, Whitcomb, Louis L., 2009. Preliminary deep water results in single-beacon one-way-travel-time acoustic navigation for underwater vehicles. In: Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on. IEEE, pp. 2053–2060.

Yoerger, Dana R., Jakuba, Michael, Bradley, Albert M., Bingham, Brian, 2007. Techniques for deep sea near bottom survey using an autonomous underwater vehicle. Int. J. Robot Res. 26 (1), 41–54.

Zhang, Z., Nov 2000. A flexible new technique for camera calibration. IEEE Trans. Pattern Anal. Mach. Intell. ISSN: 0162-8828 22 (11), 1330–1334. https://doi.org/10.1109/34.888718.

5

H-SLAM: Rao-Blackwellized Particle Filter SLAM Using Hilbert Maps

In this chapter, we propose a new SLAM framework for continuous occupancy mapping named H-SLAM. This framework is based on a RBPF with a HM map representation in each particle. As explained in detail in Section 2.3, the HM provides a continuous occupancy representation at constant computational complexity $O(1)$, while significantly reducing the amount of memory needed in traditional OGs to have a similar representation. Range measurements are used to learn the map as well as to provide self consistency measurements following the measurement model in [Thrun et al., 2005]. The system is tested both in simulation and real datasets gathered with Sparus II AUV.

The proposed method has been published in the following paper:

Paper published in **Sensors** journal
Volume: 18, Number: 5, Article: 1386, Published: May 2018
DOI: 10.3390/s18051386
JCR2016 I&I IF 2.677, Q1 (10/58)

sensors

MDPI

Article

H-SLAM: Rao-Blackwellized Particle Filter SLAM Using Hilbert Maps

Guillem Vallicrosa * and Pere Ridao

Underwater Robotics Research Center (CIRS), Computer Vision and Robotics Institute (VICOROB), Universitat de Girona, 17004 Girona, Spain; pere@eia.udg.edu

* Correspondence: gvallicrosa@eia.udg.edu

Received: 28 March 2018; Accepted: 28 April 2018; Published: 1 May 2018

Abstract: Occupancy Grid maps provide a probabilistic representation of space which is important for a variety of robotic applications like path planning and autonomous manipulation. In this paper, a SLAM (Simultaneous Localization and Mapping) framework capable of obtaining this representation online is presented. The H-SLAM (Hilbert Maps SLAM) is based on Hilbert Map representation and uses a Particle Filter to represent the robot state. Hilbert Maps offer a continuous probabilistic representation with a small memory footprint. We present a series of experimental results carried both in simulation and with real AUVs (Autonomous Underwater Vehicles). These results demonstrate that our approach is able to represent the environment more consistently while capable of running online.

Keywords: AUV (Autonomous Underwater Vehicle); SLAM (Simultaneous Localization and Mapping); PF (Particle Filter); 2D

1. Introduction

Robot localization is a fundamental problem in achieving true autonomy. Especially underwater, where global localization systems like Global Positioning System (GPS) are not available, vehicles have often to rely on Dead Reckoning (DR) navigation that drifts over time. This accumulated drift is problematic when constructing maps because a same geophysical feature may appear as a different one when it is re-observed after drifting.

To overcome this drift, systems like the Long Baseline (LBL), the Short Baseline (SBL), the Ultra-Short Baseline (USBL), the GPS Intelligent Buoyss (GIBs), or the single beacon navigation, are commonly used to provide absolute positioning fixes [1–4]. However, these systems require time for deployment and constrain the vehicle to their coverage area.

To avoid the use of external structures, a vehicle equipped with exteroceptive sensors such as sonars can make use of Terrain-Based Navigation (TBN) [5] to bound its navigational drift. However, detailed digital terrain maps are not always available. Moreover, those maps are mainly measured from surface ships, thus degrading their resolution as depth increases.

Another solution, is the use of Simultaneous Localization and Mapping (SLAM) methods [6,7], which do not require any external structures and neither a pre-obtained digital map. As in TBN, SLAM needs the use of exteroceptive sensors, mainly cameras or sonars. Although underwater cameras suffer from low visibility in turbid waters, they provide higher resolution and faster refresh rate while they are much cheaper than sonars. On the other hand, sonar sensors have lower resolution and refresh rate, but measure up to hundreds of meters regardless of water visibility issues.

Some of the most successful SLAM methods in the literature use a feature-based approach for SLAM [8–10]. Uniquely identifiable features are detected and associated to continuously correct the navigational drift and the learned map. However, underwater environments make robust feature

extraction difficult, especially on sonar measurements, and a featureless method should be used. Featureless methods can rely on scan-matching, frequency registration, . . . , where relations between different scans are obtained. Those relations are represented in a graph-like structure that can be solved/optimized with any of the state of the art back ends [11–13]. Another method is to rely on Particle Filters (PFs) where each particle carries its own map and is weighted against it for self-consistency of its measurements [14].

1.1. Underwater SLAM State of the Art

Focusing specifically on the underwater environment, multiple works have achieved successful SLAM implementations, either with optical imaging sensors or acoustic sonar sensors.

Optical imagery has been used to construct two-dimensional (2D) underwater photomosaics that correct the inherent DR drift and enable an overview of extended areas of the seafloor [15–22]. Additionally, in scenarios with a high three-dimensional (3D) component, optical imagery has also been used for 3D reconstructions [23–27].

Regarding sonar sensors, the Forward-Looking Sonar (FLS) provide a strong alternative to optical imagery mosaicking in low visibility conditions [28–31]. Although FLS provide a longer measurement range, its Field of View (FOV) is limited and the change of orientation greatly affects the perceived appearance of measured objects.

Multibeam echosounders are commonly used to obtain 2.5D elevation maps of the seafloor thanks to their wide swath and long range of measurements. Typically used on surface ships to map the seafloor, they are also used in Autonomous Underwater Vehicles (AUVs) to obtain a better resolution closer to the bottom [32–35].

Finally, mechanical scanning sonars and single beam echosounders have also been used for SLAM in man-made environments with line features [36]. Even in fully 3D environments like caves, with occupancy grids [37], as well as with scan-matching algorithms [38].

SLAM underwater is usually computed after the AUV is recovered from water and its data downloaded. After observing the obtained result, another mission can be scheduled to explore potential targets or cover the gaps of the first mission. This process can be inefficient and costly. However having the SLAM solution online, could enable autonomous exploration [39] or autonomous intervention [40] capabilities for the AUVs.

To the best of the authors knowledge the only underwater SLAM algorithms that have been tested online are [31,37]. The first uses multiple single beam echosounders and provides an Occupancy Grid (OG) map using an efficient Deferred-Reference Octree representation to avoid huge copies in its PF. While the second one uses a FLS Fourier-based registration with a pose-graph representation with loop-closing detection. While FLS mosaicking does not provide a useful representation of the environment for path planning, the OG grid map provides the perfect candidate for online path planning. OG describe the environment as free, occupied and unknown zones with certain probability. This information can be used to plan safe paths and autonomous exploration.

In our proposal, we want to work with occupancy maps because in future work they can be used for online path planning. To work with occupancy maps, we need to work with particle filters, where each particle carries their own version of the map. In [37] they reduced the memory footprint from OG maps by using an octree structure, but increased the computational complexity of the cell-query/update operation from constant $O(1)$ to logarithmic $O(\log(n))$. We propose a new SLAM framework, named Hilbert Maps SLAM (H-SLAM) which reduces the memory footprint of traditional OG maps while keeping the computational complexity constant $O(1)$. Moreover, they offer a continuous occupancy representation that can be queried at any resolution.

1.2. Contribution

The main contributions of this paper are:

1. Bring the map representation named Hilbert Maps (HMs) to the underwater environment.

2. Implement a new SLAM framework, the H-SLAM.
   (a) Use sonar measurements with HM representation.
   (b) PF based.
   (c) Capable of running online on an AUV.
3. Simulated experiments and results of the method proposed.
   (a) Experiment with a known map. Localization only (TBN).
   (b) Full SLAM experiment.
4. Real experiments and results of the method proposed.
   (a) Datasets obtained by an AUV.

1.3. Paper Organization

The paper is organized as follows. Section 2 describes the HM representation and the specifics on how to use it for map localization. Section 3 presents the Rao-Blackwellized Particle Filter (RBPF) used in conjunction with the HM representation for the H-SLAM framework. Section 4 describes the datasets used for testing the algorithms while Section 5 discusses the results obtained with them. Finally, in Section 6, we present the conclusions.

2. Hilbert Maps

HMs where recently introduced in [41] to offer a continuous probabilistic representation of the space given a collection of range sensor measurements. In other words, it offers a continuous occupancy map representation. Unlike traditional OGs, there is no cell resolution, so any point in the space can be queried. Moreover, it captures spatial relationships between measurements, thus being more robust to outliers and possessing better generalization performance and exploiting that environments have some inherent structure. For example, if two close points are observed occupied the space between them will have a higher probability of being occupied than free while no other measurements are obtained on the neighbourhood.

Developed as an alternative to the Gaussian Process Occupancy Maps (GPOMs) [42], they offer similar advantages at a smaller computational cost. While GPOMs have a cubic computational cost $O(n^3)$, HMs computational cost is constant $O(1)$. Instead of training the classifier directly on the training points $\mathbf{x}$, HMs project them to a finite set of features or inducing points $\mathbf{\Phi}(\mathbf{x})$, where a simple logistic regression classifier is learned. Those features dot product approximates popular kernels in the Gaussian Process (GP) framework $k(\mathbf{x}, \mathbf{x}') \approx \mathbf{\Phi}(\mathbf{x})^T \mathbf{\Phi}(\mathbf{x}')$, like the Radial-Basis Function. Furthermore, the logistic regression can be trained and updated using Stochastic Gradient Descent (SGD), making computation theoretically independent from the number of observations.

Given a dataset $\mathcal{D} = \{\mathbf{x}_i, y_i\}$ where $\mathbf{x}_i \in \mathbb{R}^D$ is a point in the 2D or 3D space and $y_i \in \{-1, 1\}$ is the label corresponding to the occupancy of the point $\mathbf{x}_i$. HMs learn the discriminative model $p(y|\mathbf{x}, \mathbf{w})$ on the dataset through SGD. Once the model is learned, one can use the parameters $\mathbf{w}$ to predict the probability of occupancy of any query point $\mathbf{x}_*$ as

$$p(y_* = 1|\mathbf{x}_*, \mathbf{w}) = \frac{1}{1 + \exp(-\mathbf{w}^T \mathbf{\Phi}(\mathbf{x}_*))} \in [0, 1]. \quad (1)$$

The most important parameters that define a HM are the learning rate of the SGD and features used. Regarding the learning rate η_t, it can be constant or decaying with time. Regarding the features, many different features have been applied to HMs [41,43,44], and the basic parameters common to them are the *feature_resolution* f_{res}, that defines how distant each feature are from each other, and the *radius_neighbourhood* r_{th} that defines how far a feature affects its surroundings (Figure 1). The closer

the features are, the smaller the details that can be represented. The lower the radius, the less features affect the same point in space. The feature used in this work is a simple triangle feature defined as

$$\Phi(\mathbf{x}) = \begin{cases} \frac{r_{th}-r}{r_{th}} & \text{if } r < r_{th} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where $r = ||\mathrm{f_i} - \mathbf{x}||_2$ and $\mathrm{f_i}$ is the position of the feature i.

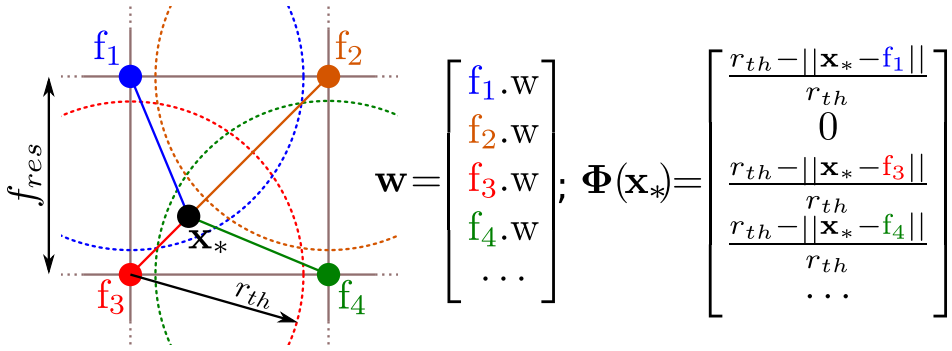
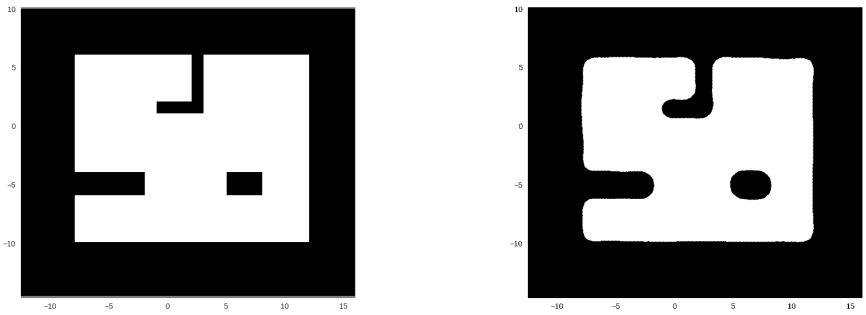


Figure 1. Schematic of a Hilbert Map. Features $\mathrm{f_i}$ are spread at f_{res} distance in a square grid and the neighbourhood that they affect is defined by the radius r_{th}. When predicting the occupancy of a point $\mathbf{x}_*$, one must gather all the feature weights and multiply it by the value of the feature in that point $\mathbf{\Phi}(\mathbf{x}_*)$ according to (1). In the example shown, the query point is outside $\mathrm{f_2}$ neighbourhood and thus, its contribution is zero.

Being a continuous representation features can be much farther than cells in a traditional OG, but achieve a similar representation at a much lower memory footprint. For example the map described in Figure 2 extends 28.5 × 24.5 m which for an occupancy grid at 0.1 m resolution takes around 70,000 cells to represent. If represented by doubles (8 bytes/double), it takes ≈545.5 kB. However a HM representation at 0.5 m feature resolution, takes ≈21.8 kB (a 0.04% of the memory) providing similar representation at 0.1 m queries.



(**a**) OG representation at 0.1 m resolution.

(**b**) HM representation with features at 0.5 m, queried at 0.1 m resolution.

Figure 2. Comparison between OG and HM representation queried at same resolution. Notice that rounded corners are not the most desirable representation for structured environments, but for underwater scenarios is not usually a drawback.

Sensors **2018**, *18*, 1386

Hilbert Map Learning and Raycasting

Learning a map from range sensors measurements and querying a point in the map, are both clearly defined in the seminal work of HMs [41]. To include range measurements, they are first discretized into single points. The point at the end of the range is labeled free if the range is maximum and occupied otherwise. Then, the rest of the ray (from vehicle position to measured range) is sampled randomly and labeled free every 1 or 2 m to properly cover the ray (Figure 3). Those points and labels are learned into the HM.

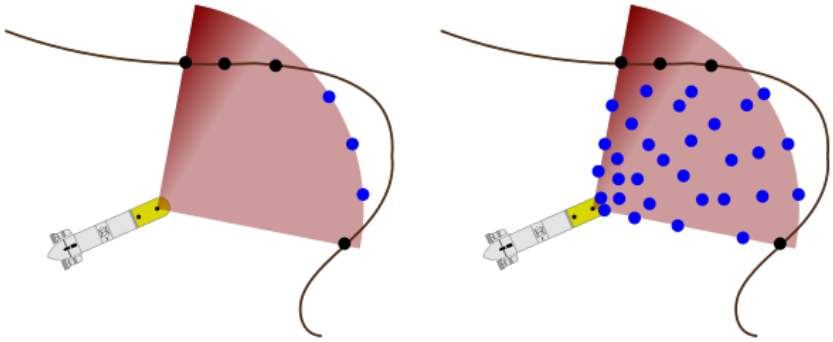


Figure 3. (**left**) Original range measurements made with a sonar. (**right**) Sampled points for map learning (black points are occupied and blue points are free).

However, to develop a SLAM framework based on HMs, it lacks a necessary raycast method to compare the real range measurements with the expected range measurements that the vehicle would have according to the learned map. On grided OG maps, the cells are queried through the ray path until an occupancy value bigger than a threshold is found [45]. Our HM raycasting method is inspired by the one developed on GPOMs [46].

The raycast starts from the vehicle position in the HM and points in the same relative direction as the real measurement. Points at increasing distance from the vehicle are queried in the HM to obtain the occupancy value (Figure 4). This distance is defined as the query resolution. When a query point has an occupancy value bigger than a threshold, this point is considered a hit (occupied) and no more points are queried. To get the exact position where the threshold was crossed, a linear interpolation between the hit point and the point previous to the hit point is computed. Finally, the raycasted range is the distance between the vehicle position and the result of the linear interpolation.

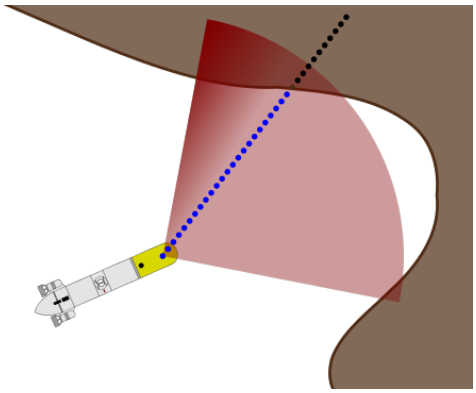


Figure 4. Example of raycast where queries are made at specific resolution.

3. Rao-Blackwellized Particle Filter with Hilbert Maps

AUVs are often loaded with a handful of sensors to provide proper positioning. Depth sensor, Attitude and Heading Reference System (AHRS) and Doppler Velocity Log (DVL) provide excellent positioning except for the x and y axis in the absence of GPS, SBL, LBL, USBL or GIB. To assess this positioning one can represent the state of the vehicle as a RBPF [47]. Here, states directly observable using vehicle sensors are removed from the PF and are tracked by a single Extended Kalman Filter (EKF) shared by all particles whose state vector is

$$\mathbf{x}_k^{ekf} = [z_k \;\; u_k \;\; v_k \;\; w_k]^T, \tag{3}$$

where $z_k k$ is the depth of the vehicle in the world frame, $[u_k \;\; v_k \;\; w_k]$ are the velocities in the vehicle frame at the time k. The vehicle orientation ϕ_k, θ_k roll and pitch and the yaw rate $\dot{\psi}_k$ in the world frame are taken as inputs $\mathbf{u}_k$ of the EKF prediction model and are not estimated. The remaining states are estimated by the PF, where each particle is defined as

$$\mathbf{x}_k^{pf,i} = \left\{ \begin{bmatrix} x_k^i & y_k^i & \psi_k^i \end{bmatrix}^T, \; w_k^i, \; m_k^i \right\}, \tag{4}$$

where i is the particle index and $[x_k^i \;\; y_k^i \;\; \psi_k^i]^T$ are the positions and the yaw in the world frame, w_k^i is the weight of the particle and m_k^i is the HM of the particle.

The particle filter is initialized from the on-board DR filter if an absolute positioning system is available. Otherwise the filter is initialized at the origin for x, y and uses the current sensor measurements to initialize the state model.

3.1. State Propagation

At each sensor measurement, the EKF is predicted to the time of the observation. A simple constant velocity model is used for the prediction as

$$\mathbf{x}_{k+1}^{ekf} = f\left(\mathbf{x}_k^{ekf}, \mathbf{u}_k, \mathbf{n}_k\right) = \begin{bmatrix} z_k + \cos(\theta_k)\cos(\phi_k)\left(w_k t + n_{w_k}\frac{t^2}{2}\right) \\ u_k + n_{u_k} t \\ v_k + n_{v_k} t \\ w_k + n_{w_k} t \end{bmatrix} \tag{5}$$

where t is the time increment from the previous prediction, $\mathbf{u}_k = [\phi_k \;\; \theta_k \;\; \dot{\psi}_k]^T$ is the input control vector and $\mathbf{n}_k = [n_{u_k} \;\; n_{v_k} \;\; n_{w_k}]^T$ are the acceleration noises in the linear velocities. Note that noises in roll and pitch $[n_{\phi_k} \;\; n_{\theta_k}]$ are so small that can be considered negligible and are not taken into account. Covariance is also predicted as

$$P_{k+1} = F_k P_k F_k^T + W_k Q_k W_k^T \tag{6}$$

where $F_k = \left.\frac{\partial f(\mathbf{x}_k^{ekf}, \mathbf{u}_k, \mathbf{n}_k)}{\partial \mathbf{x}_k^{ekf}}\right|_{\mathbf{x}_k^{ekf} = \hat{\mathbf{x}}_k^{ekf}, \mathbf{n}_k = \mathbf{0}}$, $W_k = \left.\frac{\partial f(\mathbf{x}_k^{ekf}, \mathbf{u}_k, \mathbf{n}_k)}{\partial \mathbf{n}_k}\right|_{\mathbf{x}_k^{ekf} = \hat{\mathbf{x}}_k^{ekf}, \mathbf{n}_k = \mathbf{0}}$, and $Q_k = diag\{\sigma_u \;\; \sigma_v \;\; \sigma_w\}$.

Each particle is also predicted forward by randomly sampling the uncertainties of u_k, v_k from the EKF and a user specified yaw rate uncertainty $\sigma_{\dot{\psi}}$. The velocities and their covariances are transformed for each particle from the body frame to the world frame $\{W\}$ as

$$\begin{bmatrix} {}^W\dot{x}_k^i \\ {}^W\dot{y}_k^i \\ {}^W\dot{z}_k^i \end{bmatrix} = Rot(\phi_k, \theta_k, \psi_k^i) \begin{bmatrix} u_k \\ v_k \\ w_k \end{bmatrix} \tag{7}$$

$${}^W P_{\dot{x}_k, \dot{y}_k, \dot{z}_k}^i = Rot(\phi_k, \theta_k, \psi_k^i) P_{u_k, v_k, w_k} Rot(\phi_k, \theta_k, \psi_k^i)^T \tag{8}$$

Sensors **2018**, *18*, 1386 7 of 19

where $Rot(\phi_k, \theta_k, \psi_k^i)$ is a rotation matrix given the attitude Euler angles and P_{u_k,v_k,w_k} is the 3×3 sub-matrix of P_k containing the velocity uncertainties. Those obtained values are used to predict each particle positions as

$$x_{k+1}^i = x_k^i + \mathcal{N}\left({}^W\dot{x}_k^i, {}^WP_{\dot{x}_k,\dot{x}_k}^i\right) t \tag{9}$$

$$y_{k+1}^i = y_k^i + \mathcal{N}\left({}^W\dot{y}_k^i, {}^WP_{\dot{y}_k,\dot{y}_k}^i\right) t \tag{10}$$

$$\psi_{k+1}^i = \psi_k^i + \mathcal{N}\left(\dot{\psi}_k, \sigma_{\dot{\psi}}\right) t \tag{11}$$

where $\dot{\psi}$ is taken from $\mathbf{u}_k$.

3.2. State Update

Once the prediction has been computed up to the time of the sensor measurement, the EKF state can be updated with the common EKF update equations. The measurement function is defined as

$$\mathbf{z}_k = H_k \mathbf{x}_k^{ekf} + \mathbf{v}_k \tag{12}$$

where $\mathbf{z}_k$ is the measurement, H_k defines which states are observed and $\mathbf{v}_k$ is the noise of the measurement.

Depending on the different measurements $\mathbf{z}_k$ provided by the different sensors (see Section 4.2) the H_k matrix will change. For example, the depth sensor provides depth measures and it is defined as

$$\begin{bmatrix} z \end{bmatrix}_{depth} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \mathbf{x}_k^{ekf} + \begin{bmatrix} \sigma_{depth} \end{bmatrix} \tag{13}$$

DVL sensor provides velocities in the vehicle frame, and thus it is defined as

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix}_{DVL} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x}_k^{ekf} + \begin{bmatrix} \sigma_u \\ \sigma_v \\ \sigma_w \end{bmatrix} \tag{14}$$

Finally, AHRS sensor provides orientation in roll and pitch, and angular rate in yaw $[\phi \ \ \theta \ \ \dot{\psi}]$ that are saved in the input control vector $\mathbf{u}_k$.

3.3. Weighting, Learning and Resampling

Once a sonar measurement is received, it is segmented according to the returned intensities to obtain a single range and occupancy value. If no significant intensity is found, the range is set to the maximum range value and the measure is set to *free*. Otherwise, the range is set to the range of the highest intensity and the measure is set to *occupied*.

If it is an *occupied* measurement, its range r_k^{meas} is compared with each particle map m_k^i to update the particle weight. The expected range measurement $r_k^{i,cast}$ is obtained by casting a ray as described in Section 2, from the particle position in their respective HM m_k^i. The weight update per each particle is proportional to the difference of those ranges

$$w_{k+1}^i \propto w_k^i \exp\left(-\frac{\left(r_k^{meas} - r_k^{i,cast}\right)^2}{\sigma_r^2}\right), \tag{15}$$

where σ_r is the range measurement covariance. This can be thought as a measure of self-consistency of the each particle HM.

After particle weighting, the measurement is learned in each m_k^i to be used in future weightings and to properly reconstruct the environment. These ranges are first sampled and then learned as points as explained in Section 2.

Finally, the well known Sequential Importance Resampling (SIR) is used each time the number of effective particle N_{eff} falls below half of the number of particles ($N_{eff} < N/2$) [48].

Please note that in the case of TBN, particles carry no HMs and there is a single shared HM. This shared map is only learned beforehand and never updated. The learning step is suppressed in this case.

4. Datasets

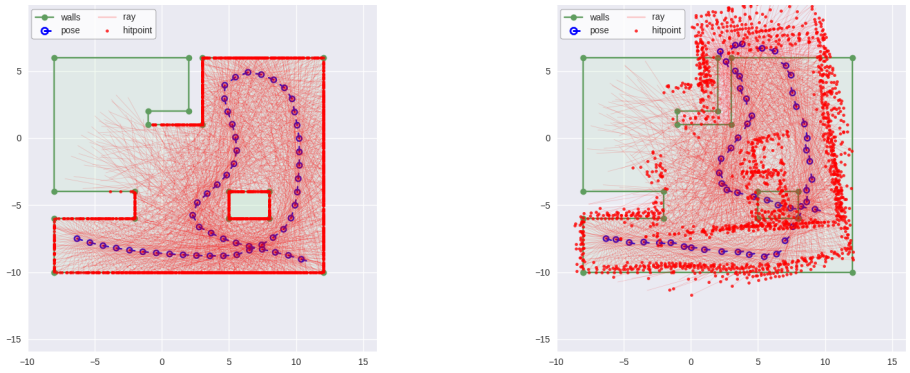
The proposed H-SLAM framework was tested on several datasets. First on a synthetic dataset to ensure correct implementation and to be able to compare against ground truth, and then with two underwater datasets, one structured and one non-structured, gathered by an AUV.

4.1. Simulated Dataset

This dataset is used as a proof-of-concept of the algorithms. The dataset is generated from a set of 53 vehicle poses in a 2D map where 36 range measurements spaced 10° around the vehicle are obtained for each pose (Figure 5a). The increments between the poses are obtained, then linear and angular gaussian noises are added to obtain the odometry measurements. The range measurements are also corrupted by gaussian noise (Figure 5b). When predicting particles, odometry increments $[\Delta x \ \ \Delta y \ \ \Delta\psi]$ are combined with gaussian noise $[\sigma_{lin} \ \ \sigma_{lin} \ \ \sigma_{ang}]$ to obtain particle positions.

This dataset is used for both TBN and SLAM. For the TBN case, the original map is sampled at 0.2 m resolution and those points are used to learn its HM representation (Figure 6). Then this map is used to localize the particles. On the SLAM case, only the noisy odometries and ranges are used as input to the filter because each particle learns its own HM m_k^i.

Using only odometry increments and ranges simplifies the filter explained in Section 3. Each particle state is propagated by compounding their current position with the noisy odometry increments.

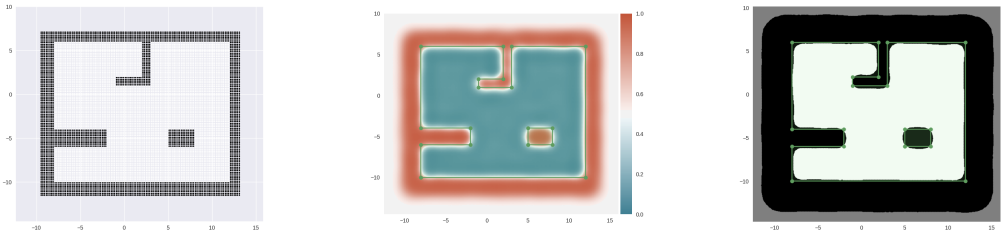


(**a**) Ground truth.

(**b**) Noisy odometry and measurements.

Figure 5. Simulated dataset of an indoor environment. Vehicle starting position on the bottom left.

Sensors **2018**, *18*, 1386



(**a**) Sampled points with label (black occupied, white free).

(**b**) Learned Hilbert Map (colored by occupancy probability).

(**c**) Thresholded Hilbert Map (black occupied, white free, grey unknown).

Figure 6. (**a**) Sampled points from the simulated scenario; (**b**) Learned HM for TBN; (**c**) Learned HM thresholded at $p(occ) > 0.5$ for the occupied, $p(occ) < 0.5$ for free, and $p(occ) = 0.5$ for unknown, to better identify the different areas.

4.2. Real-World Datasets

These datasets were obtained with Sparus II AUV [49] equipped with a Tritech SeaKing Profiling Sonar for range measurements. The Sparus II AUV provides depth information from a pressure sensor, velocities and altitude from a DVL, and attitude from an AHRS. The profiler is mounted at the payload space of the AUV (Figure 7).

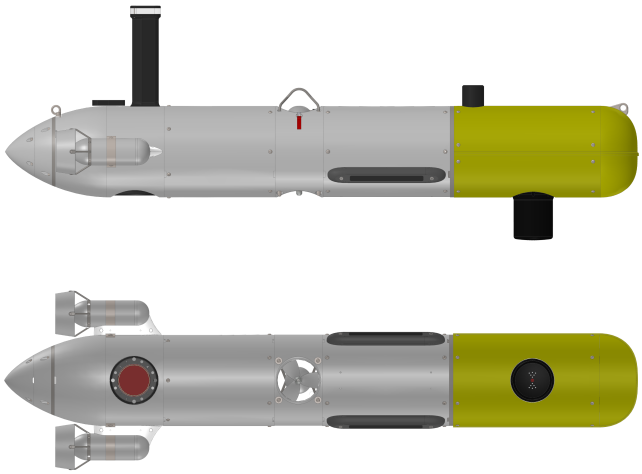


Figure 7. Sparus II AUV side view (**up**) and bottom view (**bottom**). Profiler is mounted on the payload area (in yellow) at the bottom of the vehicle.

With those sensors the AUV is capable to provide a DR navigation that drifts over time as can be observed in the following datasets. Both datasets were taken along Sant Feliu de Guixols' coast (Figure 8) at a constant depth, during the experiments regarding [50] trials.

No GPS or USBL were available to provide global corrections to the navigation drift or to provide a ground truth to compare with. The profiler provides a 120° FOV in the front of the vehicle at 1.8° angular increments. This forward-looking configuration complicates the SLAM in the sense that until a loop is closed, same locations are not measured again.

Each ray measurement provides ranges from 0 m to 10 m at 0.025 m resolution with their corresponding intensity values. Those rays are thresholded according to a minimum and maximum range, and a minimum return intensity to obtain a range measurement to be used in the H-SLAM filter.

The first dataset was taken on the man-made breakwater structure outside of the harbour. The three most eastern blocks of around 14 × 14 m with a spacing of 5 m were surveyed with the AUV (Figure 9). The dataset contains a total of 12,412 range measurements over 15 min mission at 1.5 m constant depth. As can clearly be observed on the figure, when the vehicle returns to the starting point the drift is clearly noticeable. This dataset contains three loop closes, where same features are re-observed after going around each of the three blocks.

Figure 8. Location of both real-world datasets along Sant Feliu de Guixols' coast (source: OpenStreetMap©).

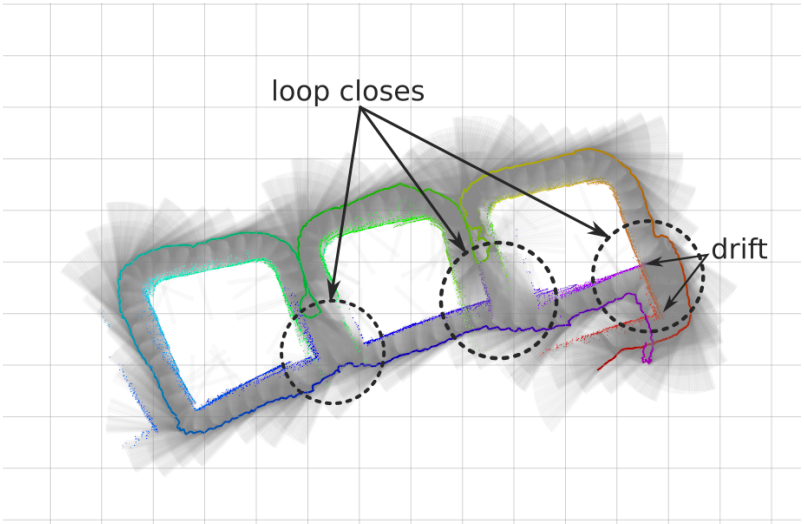


Figure 9. DR trajectory in a continuous line from magenta to red, with the corresponding profiler rays (in black) and their hitpoints (also colored by time) of the breakwater dataset. Grid cells at the background are 5 m wide.

The second dataset was taken on the natural rock structure next to the so-called *Punta del Molar*. Like the previous dataset, the AUV navigated around the rock (Figure 10). The dataset contains a total of 14,417 range measurements over 17 min mission at 2.5 m constant depth. Likewise the first dataset, the drift is clearly observable when the vehicle returns to the starting position.

Sensors **2018**, *18*, 1386

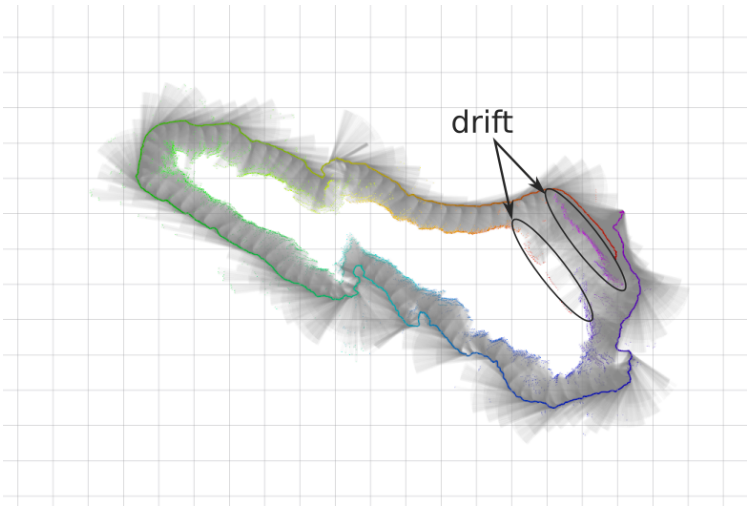


Figure 10. DR trajectory in a continuous line from magenta to red, with the corresponding profiler rays (in black) and their hitpoints (also colored by time) of the rocks dataset. Grid cells at the background are 5 m wide.

5. Results

All the tests on the different datasets were run with the similar parameters to ease the comparison of results (Table 1). Feature resolution and radius of the neighbourhood were increased for the real datasets since they are bigger than the simulated one and have less details. Range covariance was also increased due to the bigger errors obtained when dealing with real sensors.

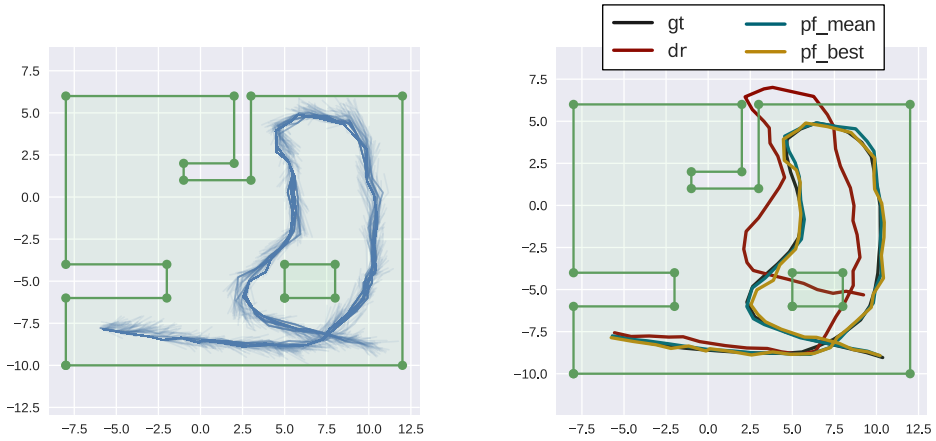
Table 1. Parameters used in the different datasets. On the real datasets prediction covariances are gathered from the covariance matrix P_k^{ekf}.

Parameter	Simulated	Breakwater	Rocks
Feature resolution (m)	0.5	1.0	1.0
Radius neighbourhood r_{th} (m)	1.5	2.0	2.0
Linear covariance σ_{lin} (m)	0.25	-	-
Angular covariance σ_{ang} (degree)	2	-	-
Range covariance σ_r (m)	0.05	0.4	0.4
Number of particles	40	40	40

5.1. Simulated Dataset

The simulated dataset is first used on a TBN experiment, where the HM is first learned from samples as explained on Section 4.1. This map is shared between particles being only queried to modify particle weights according to the differences between measured and casted rays. The results are compared against the ground truth but also against the provided odometry inputs in a DR filter that simply composes them (Figure 11).

As can be observed, the TBN corrects the vehicle trajectory reducing significantly the position error. While the DR filter error keeps increasing, the TBN error is maintained almost constant around 0.4 m (Figure 12).



(**a**) Trajectories for all particles involved in the filter.

(**b**) Comparison between ground truth, DR filter, and best and mean particle trajectories.

Figure 11. Trajectory results of TBN on HMs.

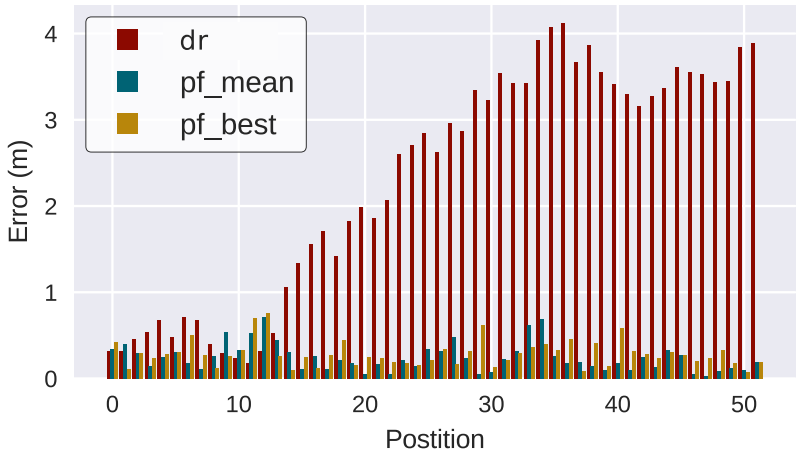


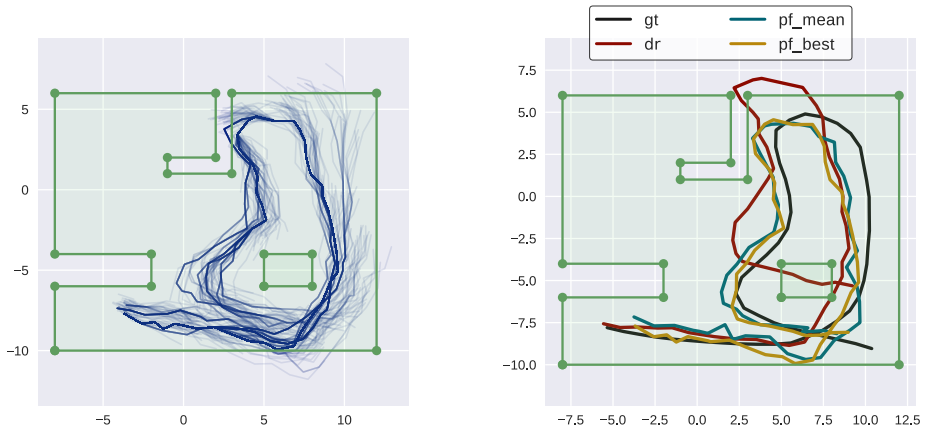
Figure 12. Position error of the filters compared to ground truth (TBN).

As expected, moving to SLAM increases the error and the correction of the trajectory is lower than in the TBN case (Figure 13).

However, errors continue to be bounded although they are much higher due to the nature of map incremental learning and self-consistency checks (Figure 14).

Another way to compare the results is to compare the map learned using ground truth odometry and measurements against the map learned by the DR filter and the H-SLAM filter (Figure 15). In this case, the representation obtained by the H-SLAM is much more close to the ground truth one than the one obtained by the DR filter.

Sensors **2018**, *18*, 1386



(**a**) Trajectories for all particles involved in the filter.

(**b**) Comparison between ground truth, DR filter, and best and mean particle trajectories.

Figure 13. Trajectory results of SLAM on HMs.

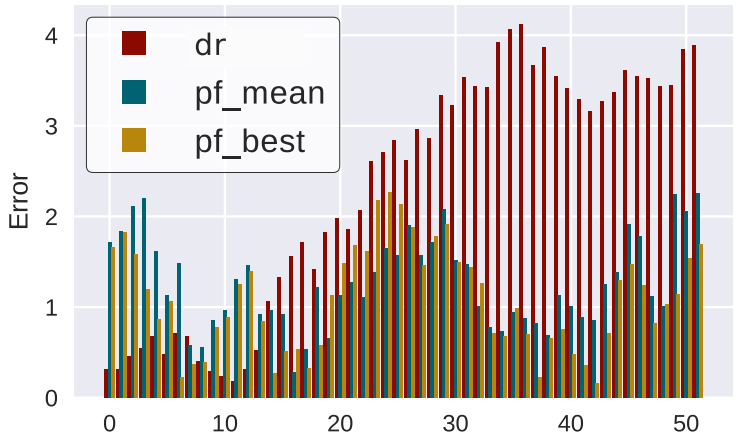
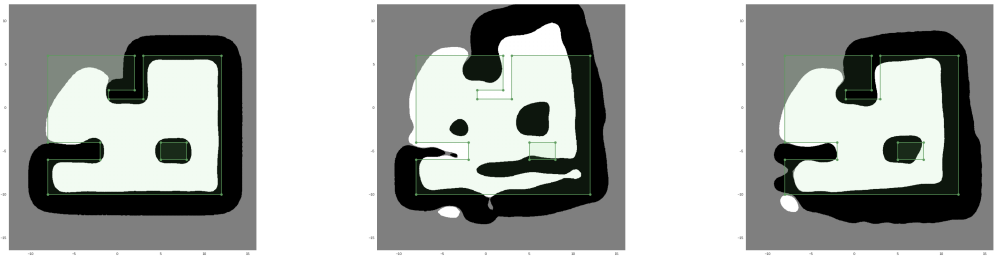


Figure 14. Position error of the filters compared to ground truth (SLAM).



(**a**) Ground truth map.

(**b**) DR map.

(**c**) Best particle map.

Figure 15. Comparison of HMs learned from different vehicle trajectories.

5.2. Breakwater Dataset

On the case of the breakwater dataset, we can observe a quite problematic area in the small corridors between the blocks. A multipath echo is clearly present when looking to the east. This multipath returns a maximum range which is interpreted as a completely free ray. This problem is clearly visible on the rightmost block, causing the HM to represent it hollow.

When the H-SLAM is applied to the breakwater dataset, the result clearly improves over the trajectory, providing more consistent sizes for the blocks and avoiding the double wall at the end of the dataset (Figure 16). Observing the reprojected measurements on the corrected trajectory, no major drifts are observed.

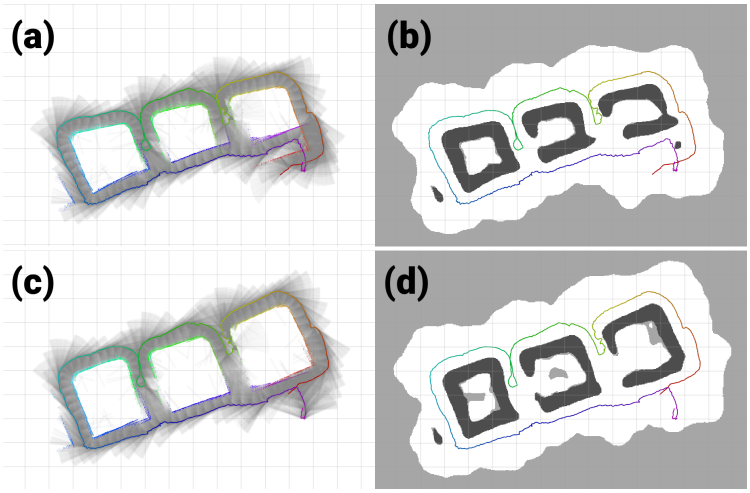


Figure 16. Breakwater dataset results. (**a**,**b**) Original dataset; (**c**,**d**) Corrected dataset; (**a**,**c**) Raw rays and endpoints with the vehicle trajectory colored by time; (**b**,**d**) Learned HM segmented to show free/unknown/occupied values with the vehicle trajectory.

The validity of H-SLAM approach can be seen when comparing the results with the satellite images because they maintain the same structure as they have underwater (Figure 17).

Figure 17. Breakwater HM superimposed with a satellite image (source: Map data ©2018 Google, Inst. Geogr. Nacional, Spain).

Finally, observing the covariance of the particles over time (Figure 18), the three loop closing events described in Section 4.2 produce a clear decrease in uncertainty of the H-SLAM localization.

Sensors **2018**, *18*, 1386

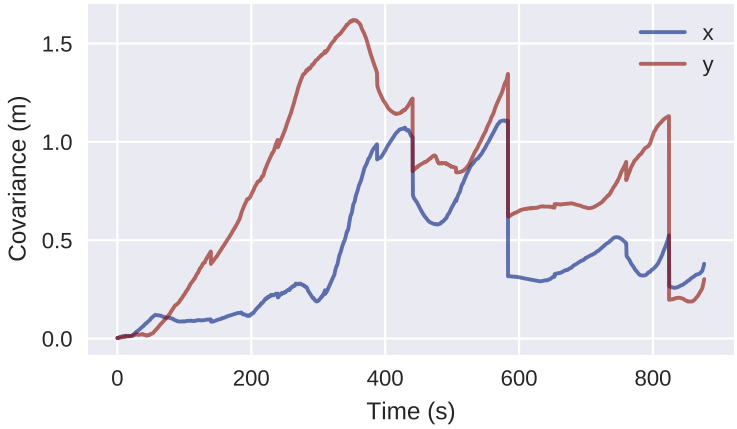


Figure 18. Covariance of the particles over time for the Breakwater dataset.

5.3. Rocks Dataset

On the rocks dataset the same parts of the map are not observed until the trajectory finishes. Incremental corrections are made during the whole dataset and at the end a loop-closing is achieved. Several outliers are observed at the boundaries of the dataset due to proximity to other rock formations (Figure 19).

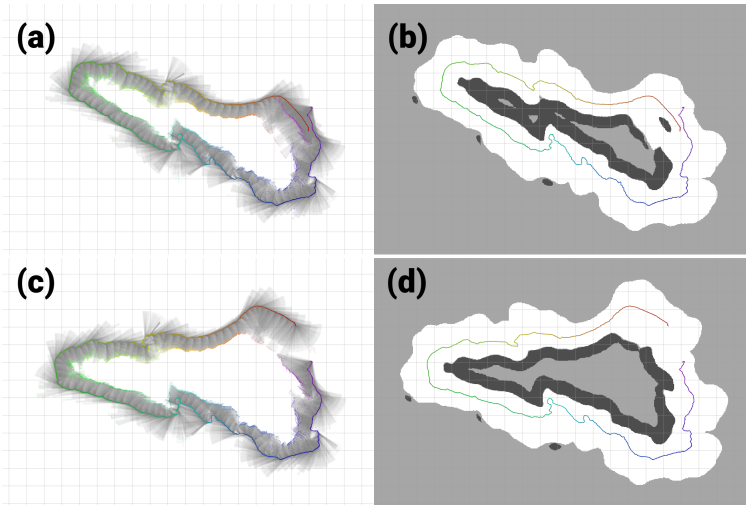


Figure 19. Rocks dataset results. (**a**,**b**) Original dataset. (**c**,**d**) Corrected dataset. (**a**,**c**) Raw rays and endpoints with the vehicle trajectory colored by time. (**b**,**d**) Learned HM segmented to show free/unknown/occupied values with the vehicle trajectory.

Although the natural rock structure does not maintain the same structure underwater, when comparing the results with the satellite images, the validity of H-SLAM can be seen (Figure 20). Furthermore, the small occupied spots on the south-western part of the explored zone are clearly caused by the nearby rock structures.

Finally, observing the covariance of the particles over time (Figure 21), a loop closing event is observed at around 900 s that corresponds to revisiting the initial area.

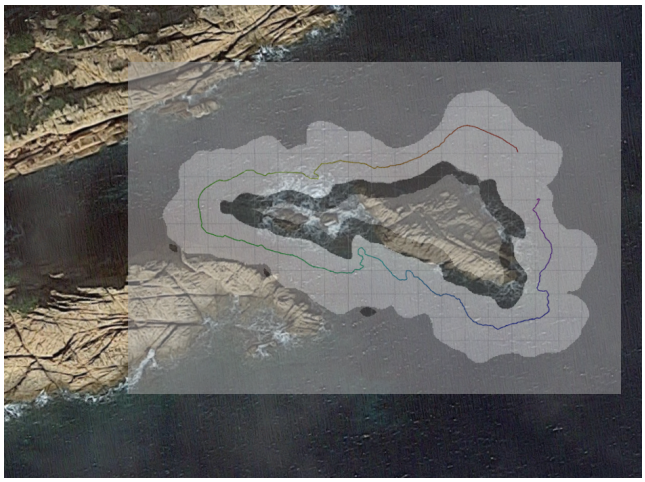


Figure 20. Rocks HM superimposed with a satellite image (source: Map data ©2018 Google, Inst. Geogr. Nacional, Spain).

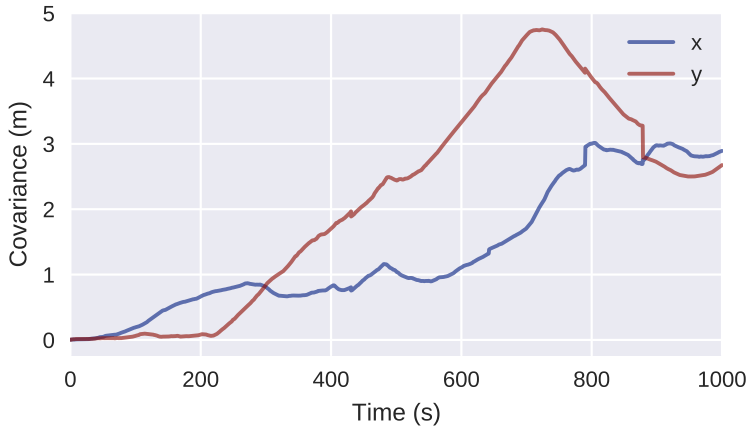


Figure 21. Covariance of the particles over time for the Rocks dataset.

5.4. Performance

H-SLAM was not run online when obtaining the datasets, but from previously obtained datasets saved in a *rosbag* file. This file, part of the Robot Operating System (ROS) [51], allows to replay data exactly as how it was obtained. In this case, the algorithm was run as fast as possible through the *bagfile* to compare the time it took to gather data (total available time for execution) against the time needed to compute the H-SLAM solution (Table 2).

Table 2. Computing time comparison with dataset collection time.

	Breakwater	**Rocks**
Time to obtain dataset	14 min 36 s	16 min 54 s
Time to run H-SLAM	02 min 26 s	03 min 42 s

As can be observed, the computing time is much lower an thus making the algorithm capable of running online on the AUV, even with many more particles than the 40 used on the tests.

6. Conclusions

In this work, we have presented new SLAM framework named H-SLAM for AUVs equipped with sonars. The combination of a RBPF with a HM representation of the environment provided trajectory corrections that increased the consistency of the recorded measurements both in simulation and in real datasets. Moreover, the computing time required is much lower than the time it took to collect the datasets, being capable of being used online on an AUV.

In the simulated datasets, the RBPF provided a significant correction when used for TBN with a known map, and a lesser correction when used for SLAM. However the final map was much more consistent than the one obtained by the DR filter.

In the real datasets, significantly more consistent maps were also obtained. Especially on the breakwater dataset, the multiple closing loops allowed to obtain a correct trajectory and map that matches the satellite image of the structure.

7. Future Work

The algorithms have been tested at constant depth providing continuous occupancy maps in 2D. Future work must better reflect the nature of underwater environments, extending H-SLAM to the 3D case. Moreover, multipath errors observed on the real datasets should be filtered out. Our idea is check the range measurements persistence over time before using them in H-SLAM.

Author Contributions: The work presented in this paper has been carried out in collaboration with all authors. G.V. designed, implemented and tested the system with the collaboration of P.R. who was the project leader in charge of the direction and supervision. All authors discussed the results obtained together and reviewed the manuscript.

Funding: Research funded by Ministerio de Educación, Cultura y Deporte (PhD grant ref. FPU12/05384 and ARCHROV project ref. DPI2014-57746-C3-3-R), and by the European Comission (EUMR project ref. H2020-INFRAIA-2017-1-twostage-731103).

Acknowledgments: The authors would like to thank Lionel Ott and Fabio Ramos, for kindly providing code for the original HM framework, and to Stefan Williams for hosting Guillem in the ACFR during the preliminary development of this work.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Mandt, M.; Gade, K.; Jalving, B. Integrating DGPS-USBL position measurements with inertial navigation in the HUGIN 3000 AUV. In Proceedings of the 8th Saint Petersburg International Conference on Integrated Navigation Systems, Saint Petersburg, Russia, 28–30 May 2001; pp. 28–30.
2. Thomas, H.G. GIB buoys: An interface between space and depths of the oceans. In Proceedings of the 1998 Workshop on Autonomous Underwater Vehicles, Cambridge, MA, USA, 20–21 August 1998; pp. 181–184.
3. Batista, P.; Silvestre, C.; Oliveira, P. Single beacon navigation: Observability analysis and filter design. In Proceedings of the American Control Conference (ACC), Baltimore, MD, USA, 30 June–2 July 2010; pp. 6191–6196.
4. Vallicrosa, G.; Ridao, P. Sum of gaussian single beacon range-only localization for AUV homing. *Ann. Rev. Control* **2016**, *42*, 177–187. [CrossRef]
5. Melo, J.; Matos, A. Survey on advances on terrain based navigation for autonomous underwater vehicles. *Ocean Eng.* **2017**, *139*, 250–264. [CrossRef]
6. Durrant-Whyte, H.; Bailey, T. Simultaneous localization and mapping: Part I. *IEEE Robot. Autom. Mag.* **2006**, *13*, 99–110. [CrossRef]
7. Bailey, T.; Durrant-Whyte, H. Simultaneous localization and mapping (SLAM): Part II. *IEEE Robot. Autom. Mag.* **2006**, *13*, 108–117. [CrossRef]
8. Dissanayake, M.G.; Newman, P.; Clark, S.; Durrant-Whyte, H.F.; Csorba, M. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Trans. Robot. Autom.* **2001**, *17*, 229–241. [CrossRef]

9. Davison, A.J.; Reid, I.D.; Molton, N.D.; Stasse, O. MonoSLAM: Real-time single camera SLAM. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 1052–1067. [CrossRef] [PubMed]
10. Mur-Artal, R.; Tardós, J.D. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Trans. Robot.* **2017**, *33*, 1255–1262. [CrossRef]
11. Kaess, M.; Ranganathan, A.; Dellaert, F. iSAM: Incremental smoothing and mapping. *IEEE Trans. Robot.* **2008**, *24*, 1365–1378. [CrossRef]
12. Kümmerle, R.; Grisetti, G.; Strasdat, H.; Konolige, K.; Burgard, W. g2o: A general framework for graph optimization. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, 9–13 May 2011; pp. 3607–3613.
13. Salas-Moreno, R.F.; Newcombe, R.A.; Strasdat, H.; Kelly, P.H.; Davison, A.J. Slam++: Simultaneous localisation and mapping at the level of objects. In Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, OR, USA, 23–28 June 2013; pp. 1352–1359.
14. Montemerlo, M.; Thrun, S.; Koller, D.; Wegbreit, B. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In Proceedings of the AAAI National Conference on Artificial Intelligence, Edmonton, AB, Canada, 28 July–1 August 2002.
15. Eustice, R.; Pizarro, O.; Singh, H.; Howland, J. UWIT: Underwater Image Toolbox for optical image processing and mosaicking in MATLAB. In Proceedings of the 2002 International Symposium on Underwater Technology, Tokyo, Japan, 16–19 April 2002; pp. 141–145.
16. Gracias, N.R.; Van Der Zwaan, S.; Bernardino, A.; Santos-Victor, J. Mosaic-based navigation for autonomous underwater vehicles. *IEEE J. Ocean. Eng.* **2003**, *28*, 609–624. [CrossRef]
17. Ridao, P.; Carreras, M.; Ribas, D.; Garcia, R. Visual inspection of hydroelectric dams using an autonomous underwater vehicle. *J. F. Robot.* **2010**, *27*, 759–778. [CrossRef]
18. Escartín, J.; Garcia, R.; Delaunoy, O.; Ferrer, J.; Gracias, N.; Elibol, A.; Cufi, X.; Neumann, L.; Fornari, D.J.; Humphris, S.E.; et al. Globally aligned photomosaic of the Lucky Strike hydrothermal vent field (Mid-Atlantic Ridge, 37°18.5′ N): Release of georeferenced data, mosaic construction, and viewing software. *Geochem. Geophys. Geosyst.* **2008**, *9*. [CrossRef]
19. Singh, H.; Howland, J.; Pizarro, O. Advances in large-area photomosaicking underwater. *IEEE J. Ocean. Eng.* **2004**, *29*, 872–886. [CrossRef]
20. Bingham, B.; Foley, B.; Singh, H.; Camilli, R.; Delaporta, K.; Eustice, R.; Mallios, A.; Mindell, D.; Roman, C.; Sakellariou, D. Robotic tools for deep water archaeology: Surveying an ancient shipwreck with an autonomous underwater vehicle. *J. F. Robot.* **2010**, *27*, 702–717. [CrossRef]
21. Elibol, A.; Gracias, N.; Garcia, R.; Gleason, A.; Gintert, B.; Lirman, D.; Reid, P. Efficient autonomous image mosaicing with applications to coral reef monitoring. In Proceedings of the IROS 2011 Workshop on Robotics for Environmental Monitoring, San Francisco, CA, USA, 25–30 September 2011.
22. Eustice, R.; Singh, H.; Leonard, J.J.; Walter, M.R.; Ballard, R. Visually Navigating the RMS Titanic with SLAM Information Filters. *Robot. Sci. Syst.* **2005**, *6*, 57–64.
23. Williams, S.; Mahon, I. Simultaneous localisation and mapping on the great barrier reef. In Proceedings of the 2004 IEEE International Conference on Robotics and Automation, New Orleans, LA, USA, 26 April–1 May 2004; Volume 2, pp. 1771–1776.
24. Pizarro, O.; Eustice, R.; Singh, H. Large area 3D reconstructions from underwater surveys. In Proceedings of the OCEANS'04, MTTS/IEEE TECHNO-OCEAN'04, Kobe, Japan, 9–12 November 2004; Volume 2, pp. 678–687.
25. Nicosevici, T.; Gracias, N.; Negahdaripour, S.; Garcia, R. Efficient three-dimensional scene modeling and mosaicing. *J. F. Robot.* **2009**, *26*, 759–788. [CrossRef]
26. Zhang, H.; Negahdaripour, S. EKF-based recursive dual estimation of structure and motion from stereo data. *IEEE J. Ocean. Eng.* **2010**, *35*, 424–437. [CrossRef]
27. Johnson-Roberson, M.; Pizarro, O.; Williams, S.B.; Mahon, I. Generation and visualization of large-scale three-dimensional reconstructions from underwater robotic surveys. *J. F. Robot.* **2010**, *27*, 21–51. [CrossRef]
28. Kim, K.; Neretti, N.; Intrator, N. Mosaicing of acoustic camera images. *IEE Proc. Radar Sonar Navig.* **2005**, *152*, 263–270. [CrossRef]
29. Negahdaripour, S.; Firoozfam, P.; Sabzmeydani, P. On processing and registration of forward-scan acoustic video imagery. In Proceedings of the 2nd Canadian Conference on Computer and Robot Vision, Victoria, BC, Canada, 9–11 May 2005; pp. 452–459.

30. Aykin, M.D.; Negahdaripour, S. On Feature Matching and Image Registration for Two-dimensional Forward-scan Sonar Imaging. *J. F. Robot.* **2013**, *30*, 602–623. [CrossRef]
31. Hurtós, N.; Ribas, D.; Cufí, X.; Petillot, Y.; Salvi, J. Fourier-based Registration for Robust Forward-looking Sonar Mosaicing in Low-visibility Underwater Environments. *J. F. Robot.* **2015**, *32*, 123–151. [CrossRef]
32. Roman, C.; Singh, H. Improved vehicle based multibeam bathymetry using sub-maps and SLAM. In Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, AB, Canada, 2–6 August 2005; pp. 3662–3669.
33. Barkby, S.; Williams, S.B.; Pizarro, O.; Jakuba, M.V. A featureless approach to efficient bathymetric SLAM using distributed particle mapping. *J. F. Robot.* **2011**, *28*, 19–39. [CrossRef]
34. Barkby, S.; Williams, S.B.; Pizarro, O.; Jakuba, M.V. Bathymetric particle filter SLAM using trajectory maps. *Int. J. Robot. Res.* **2012**, *31*, 1409–1430. [CrossRef]
35. Palomer, A.; Ridao, P.; Ribas, D. Multibeam 3D underwater SLAM with probabilistic registration. *Sensors* **2016**, *16*, 560. [CrossRef] [PubMed]
36. Ribas, D.; Ridao, P.; Tardós, J.D.; Neira, J. Underwater SLAM in man-made structured environments. *J. F. Robot.* **2008**, *25*, 898–921. [CrossRef]
37. Fairfield, N.; Kantor, G.; Wettergreen, D. Real-Time SLAM with Octree Evidence Grids for Exploration in Underwater Tunnels. *J. F. Robot.* **2007**, *24*, 3–21. [CrossRef]
38. Mallios, A.; Ridao, P.; Ribas, D.; Hernández, E. Scan matching SLAM in underwater environments. *Auton. Robot.* **2014**, *36*, 181–198. [CrossRef]
39. Hernández, J.D.; Vidal, E.; Greer, J.; Fiasco, R.; Jaussaud, P.; Carreras, M.; García, R. AUV online mission replanning for gap filling and target inspection. In Proceedings of the OCEANS 2017-Aberdeen, Aberdeen, UK, 19–22 June 2017; pp. 1–4.
40. Palomeras, N.; Carrera, A.; Hurtós, N.; Karras, G.C.; Bechlioulis, C.P.; Cashmore, M.; Magazzeni, D.; Long, D.; Fox, M.; Kyriakopoulos, K.J.; et al. Toward persistent autonomous intervention in a subsea panel. *Auton. Robot.* **2016**, *40*, 1279–1306. [CrossRef]
41. Ramos, F.; Ott, L. Hilbert maps: Scalable continuous occupancy mapping with stochastic gradient descent. *Int. J. Robot. Res.* **2016**, *35*, 1717–1730. [CrossRef]
42. O'Callaghan, S.T.; Ramos, F.T. Gaussian process occupancy maps. *Int. J. Robot. Res.* **2012**, *31*, 42–62. [CrossRef]
43. Guizilini, V.; Ramos, F. Large-scale 3D scene reconstruction with Hilbert Maps. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016; pp. 3247–3254.
44. Guizilini, V.C.; Ramos, F.T. Unsupervised Feature Learning for 3D Scene Reconstruction with Occupancy Maps. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; pp. 3827–3833.
45. Amanatides, J.; Woo, A. A fast voxel traversal algorithm for ray tracing. *Eurographics* **1987**, *87*, 3–10.
46. Hata, A.Y.; Wolf, D.F.; Ramos, F.T. Particle filter localization on continuous occupancy maps. In *International Symposium on Experimental Robotics*; Springer: Roppongi, Tokyo, Japan, 3–6 October 2016; pp. 742–751.
47. Doucet, A.; De Freitas, N.; Murphy, K.; Russell, S. Rao-Blackwellised particle filtering for dynamic Bayesian networks. In Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence, San Francisco, CA, USA, 30 June–3 July 2000; Morgan Kaufmann Publishers Inc.: San Mateo, CA, USA, 2000; pp. 176–183.
48. Thrun, S. Probabilistic Robotics. *Commun. ACM* **2002**, *45*, 52–57. [CrossRef]
49. Carreras, M.; Hernández, J.D.; Vidal, E.; Palomeras, N.; Ribas, D.; Ridao, P. Sparus II AUV—A Hovering Vehicle for Seabed Inspection. *IEEE J. Ocean. Eng.* **2018**. [CrossRef]
50. Vidal, E.; Hernández, J.D.; Istenič, K.; Carreras, M. Optimized environment exploration for autonomous underwater vehicles. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018. (to be published)
51. Quigley, M.; Conley, K.; Gerkey, B.; Faust, J.; Foote, T.; Leibs, J.; Wheeler, R.; Ng, A.Y. ROS: An open-source Robot Operating System. In Proceedings of the ICRA Workshop on Open Source Software, Kobe, Japan, 17 May 2009; Volume 3, p. 5.

6

Results & Discussion

In this chapter we present the main results and discussions derived from this thesis. First we start with a summary of the completed work in Section 6.1. Then we move to the results of the range-only localization algorithm in Section 6.2 and the results of the continuous occupancy mapping algorithm in Section 6.3.

6.1 Summary of completed work

It is known that to have true autonomous vehicles, a good online localization is needed. In this thesis we have proposed two online localization methods for AUVs. The first one works in a restricted environment to find a beacon using range-only measurements. The other works in an unrestricted environment and provides continuous occupancy maps.

In Chapter 2 we provided basic background on the sensors and AUVs that are used in the real-world tests carried throughout this thesis. Moreover, a detailed description of HMs for occupancy representation is provided to extend the description already provided in Chapter 5.

In Chapter 3 we addressed the problem of range-only localization for homing. On long term deployed AUVs with DSs, AUVs must be always able to return to their DS for battery recharging and/or data uploading. We proposed two algorithms based on SOG filter already defined in previous works [Vallicrosa et al., 2014, Vallicrosa et al., 2015] and tested them to compare their performances.

In Chapter 4 we took one of the algorithms proposed in Chapter 3 and made a simpler but more robust implementation. This algorithm was coupled with a vision-based localization system to provide continuous position updates for a complete homing and docking procedure. The development of this work started in simulation [Vallicrosa et al., 2016], then moved to simple trials in the water tank and harbor [Hurtós et al., 2017], to finally be extensively tested on the LOON testbed during the final trials of LOON-DOCK/SUNRISE project.

In Chapter 5 we addressed the problem of occupancy mapping. This work started as online Terrain-Based Navigation (TBN) navigation based on Octomap [Vallicrosa et al., 2013a, Vallicrosa et al., 2013c, Vallicrosa et al., 2013b]. Then shifted to propose a new SLAM framework, named H-SLAM, combining a RBPF with HM continuous occupancy representation. This framework was tested in several 2D datasets.

All of the algorithms developed in the various chapters are implemented efficiently to be run online in an AUV. This is the basic premise maintained throughout this thesis, an essential step to increase the autonomy of AUVs.

6.2 Online Range-Only Localization for Homing

We proposed two different algorithms tested with an unknown beacon depth (3D) and with known depth (2D) using the projection of range measurements:

SOG-DR was proposed to locate an acoustic beacon given only range measurements and the DR navigation provided by the AUV's navigation filter. This algorithm assumes that the drift during localization is small. A SOG filter is implemented where each Gaussian represents a static possible position of the beacon localization. At each

range measurement an EKF update is triggered to each Gaussian, then their weights are updated according to the innovation of the EKF update.

SOG-SLAM was proposed to locate simultaneously an acoustic beacon and an AUV, given range measurements and navigation sensor measurements (pressure cell, DVL, and AHRS). This algorithm runs a complete navigation filter in each Gaussian of the SOG that also incorporates one possible position of the beacon. At each navigation sensor measurement, the Gaussian is predicted and updated with the new data. At each range measurement an EKF update is triggered to each Gaussian, then their weights are updated according to the innovation of the EKF update. Finally, a custom resampling step is tested where first copy of a Gaussian is identical, but successive copies are sampled from their own beacon location uncertainty.

We have also proposed two initialization methods for SOG filters when the first range measurement is received:

Naive initialization is based on a user specified parameter, the tangential uncertainty σ_t. This parameter defines how much space each Gaussian covers on the spherical shell. In the 2D case it simply divides the circumference by $2\sigma_t$ to obtain the angular increment between each Gaussian, providing a uniform coverage of the possible beacon positions (more details can be found in Chapter 3). This initialization was tested with the SOG-DR algorithm.

Geodesic Grid initialization provides a uniform coverage for the 3D case. By using the subdivisions of a regular icosahedron, the spherical shell can be covered with 12, 42, 162, 642, ..., Gaussians. This also has the advantage of an upper bound in the maximum number of Gaussians used in the localization whatever the value of the first range measurement. This upper bound can be linked to the computational power of the AUV to reduce its computational burden. The position of each Gaussian is precomputed for the unitary sphere and then scaled when the first range measurement is received. This initialization was tested with the SOG-SLAM algorithm.

To evaluate the SOG filter localization performance, an equivalent Gaussian formulation was proposed. The equivalent Gaussian is computed after each range update by obtaining the Unscented Transform of each Gaussian in the SOG and then computing the mean and covariance of those Unscented points. As can be seen in Chapter 3, this equivalent Gaussian gives a result very similar to sampling a large number of points from each Gaussian and computing their mean and covariance. This ensures a fast and reliable computation. The equivalent Gaussian is compared with the user provided uncertainty localization threshold. When this Gaussian has a lower uncertainty in its biggest axis than

the threshold, the beacon is considered localized. This is the stopping condition of the range-only localization algorithms.

On top of the SOG filter, an AL algorithm was proposed to autonomously decide which actions contribute more to reduce the filter uncertainty. A set of actions as 3D motions are proposed. This method takes advantage of the fact that the EKF uncertainty update does not depend on the measure itself. Then, each Gaussian in the SOG evaluates each of the actions and checks its future uncertainty. This uncertainty is equivalent to the entropy of the system, and its minimization is equivalent to the D-optimality criterion. A voting system is used, where each Gaussian in the SOG votes for the action that reduces more its future entropy. The most voted action is the one selected as the next movement.

A total of 12 experiments were proposed to evaluate these methods (Chapter 3). Four of them were teleoperated, while the remaining eight were autonomously run thanks to the AL algorithm. As stated in Chapter 3 results, the entropy of the filter is continuously decreasing after each AL action is taken. The localization errors follow the same trend as the entropy, and significant reductions are observed when localization symmetries are resolved.

To assess accurate localization position, the localized beacon is approached by the AUV. When the forward-looking camera precisely localizes the beacon, this measure is assumed as ground truth and compared with the range-only localization.

All localization errors in the experiments fall below 4 m, while in 9 of them the error is below 2.5 m. Although the 3 experiments with errors bigger than 2.5 m might seem significant, the beacon was correctly localized with the camera, ensuring correct localization for docking.

Those experiments where later re-run online on the recorded data, to compare the different algorithms given the exact same inputs. SOG-DR and SOG-SLAM showed no substantial difference in performance when the beacon depth is unknown (3D). However, when depth is known (2D), localization performs better than in the previous case, in several of the experiments.

6.2.1 Full homing and docking strategy

Although the DS range-only localization was initially developed for the Spanish national project COMAROB, it was not until the start of European project LOON-DOCK/SUNRISE that a full homing and docking strategy was implemented. The LOON-DOCK project aimed to extend the Littoral Ocean Observatory Network (LOON) testbed by incorporating a DS with an AUV to be able to demonstrate data transmission from AUV missions to the Internet.

A cost effective solution was required to achieve this goal and the online range-only localization for homing was selected together with a vision-based localization. Range-only localization located the DS from a distant location, and then was approached until the

vision-based navigation could provide better estimates for the final docking maneuvers.

From the previously presented algorithms, the SOG-DR with known depth was selected for having a similar performance but being simpler to work with. Instead of teleoperating the AUV or using AL, a predefined trajectory that ensured observability was used. This trajectory was scaled according to the first range measurement and was run until the specified uncertainty threshold was reached. An additional step was added to correct the tangential uncertainty observed in all localizations. This was achieved by moving the AUV to a point in the axis of major uncertainty, e.g. giving reductions from $\sigma_t = 1.5$ to $\sigma_t = 0.7$ (more details in Chapter 4).

This wider LOON-DOCK project also required the development of a cost-effective DS for Sparus II AUV, a control strategy to achieve successful docking and the integration between range-only localization and vision-based localization to provide continuous updates on the DS localization. Additionally, modifications to the software architecture were done to be able to communicate through the project infrastructure.

A total of 15 experiments were proposed to evaluate the whole homing and docking procedure (Chapter 4). Twelve of the experiments were a complete success, while one successfully docked but was not correctly reported to the control system, and the last two (after several retries) were aborted due to strong water currents in the vicinity of the DS. The limited maneuverability of a torpedo shape AUV such as Sparus II cannot cope with those kinds of lateral water currents when docking. This could be solved by a DS that automatically aligns itself to the currents as discussed in Chapter 4. When current was aligned the docking maneuver was always successful even when visibility was below 3 m. In the results we can also observe an strong agreement between the range measurements and the visual detections.

Range measurements were also integrated into the on-board navigation system provided by the AUV's software architecture COLA2. These measurements can provide navigation corrections once the DS position is known. For example, when the AUV starts a new mission from the DS, its position is known and can be used to consistently update the navigation filter while range measurements are available. Although this is used in Chapter 4, it is explained in more detail in [Vallicrosa et al., 2016].

6.3 Online Continuous Occupancy Mapping

We proposed a new SLAM framework named H-SLAM for continuous occupancy mapping. It uses a RBPF representation where directly observable states (depth and linear velocities) are tracked through a common EKF filter while position in the xy-plane and heading are tracked by the PF. Each particle also has a weight and carries its own map solution according to its particular path. The final map solution and trajectory is obtained by selecting the most weighted particle. The comparison of obtained measurements with

each particle map serves as a measure of self consistency that rewards the most consistent maps with higher particle weights.

For the map representation, HMs were selected. They offer a continuous occupancy representation that can be very useful for future works combining it with online path planning. They also offer a high memory reduction compared to traditional OG maps while maintaining their computational complexity constant $O(1)$ as is detailed in Section 2.3. A simple triangle feature mapping that is fast to compute was chosen for this work. This feature mapping also has a very sparse nature that reduces its computation time. A raycasting method was also developed in order to be able to compare real range measurements with expected measurement from the HM. This method is essential in the particle weighting step of the H-SLAM.

H-SLAM was tested in simulation before moving to real-world datasets. Simulation datasets provided precise ground truth to compare with the obtained results. First a TBN algorithm was tested. In this case, particles carried no map and there was a single HM that was shared among all particles and learned beforehand. This test proved very successful maintaining the localization error below 0.5 m even though the simulated noise was high.

Afterwards the complete H-SLAM was tested on the same simulated dataset. In this case the localization performed worse than in the TBN case as expected. Nonetheless it provided a significantly more consistent map that the one obtained with DR navigation. Moreover, the position errors were bounded below 2 m.

Finally, two real-world datasets obtained by Sparus II AUV equipped with a MSIS profiler sonar were tested. The first dataset contains a trajectory around three square blocks that are part of a breakwater structure. A significant navigation drift was observed in the DR navigation at the end of the path. Three loop closing events also took place during the circumnavigation of the blocks. Those loop closes proved really useful when computing H-SLAM, clearly dropping the uncertainty of the particles position after being observed and maintaining it mostly below 1 m. The final map represents correctly the three square blocks shape. However, multi-path problems in the sonar measurements were observed. Those decreased the quality of map representation by showing free space inside the blocks.

The second real-world dataset contains a circumnavigating trajectory around a natural rock formation. A significant navigation drift was also observed at the end of the trajectory. A single loop closing event is available almost at the end of the trajectory when the initial area is re-observed. This is a more complicated dataset for the SLAM problem in general. H-SLAM constantly increased the particles uncertainty (up to 4.8 m) until the loop-closing event. Having no ground truth available makes it difficult to assess the correct performance of the solution. This is why the final map was compared to satellite images of the rock formation, that provided a clear match.

Although H-SLAM was not tested in real-time, the time needed to compute the map

and the trajectory was compared to the time it took to gather the datasets. In both cases, the H-SLAM computation time represents less than 22% of the time to gather the data. This ensures the possibility of running the algorithm in real-time, and even to increase the number of particles used. Note that the computer where H-SLAM was tested has similar performance to the one in the AUVs.

6.3.1 Effect of H-SLAM parameters

The tunning of the filter parameters is very important to obtain a good result when working with HMs. The most important parameters are the feature resolution f_{res} (distance between features) and the radius of their neighborhood r_{th}. The first contributes to the level of detail that the HM can achieve. Different resolutions lead to finer or coarser HMs (Fig. 6.1).

Figure 6.1: Comparison of different feature resolutions f_{res} while number of particles is 20 and the radius of the neighborhood is defined as $r_{th} = 2f_{res}$. From left to right, resolutions are 0.5, 1.0, 1.5, 2.0 and 2.5 m.

The second, defines to how many neighboring features each measured point contributes. The bigger the radius, the more features in the vicinity of a point. This also increases the computation time because more neighbors, means more weight updates per point (Fig. 6.2).

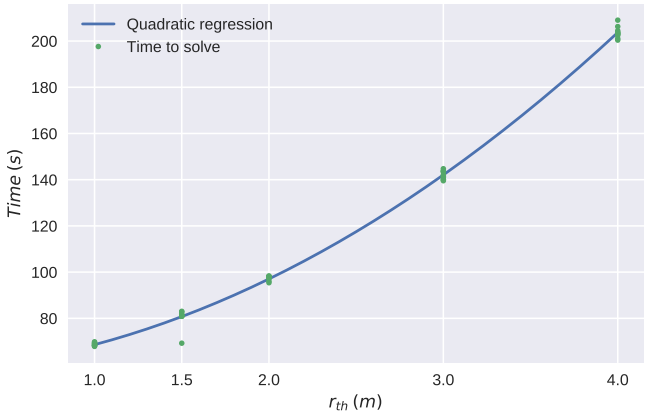


Figure 6.2: Computation time versus r_{th} with 20 particles and $f_{res} = 1.0$.

Bigger radius implies that feature values are more interlinked between neighboring inducing points. If the radius is set equal or below f_{res} we will obtain completely inde-

pendent features, while if it is too high points will affect features that are much farther, losing details in the HM (Fig.6.3).

Figure 6.3: Comparison of different radius of neighborhood r_{th} while the feature resolution f_{res} is kept constant at 1 m resolution. From left to right, radius are 1.0, 1.5, 2.0, 3.0 and 4.0 m.

Finally, another important parameter is the number of particles. Once f_{res} and r_{th} are fixed, the number of particles increases the computational time linearly (Fig.6.4).

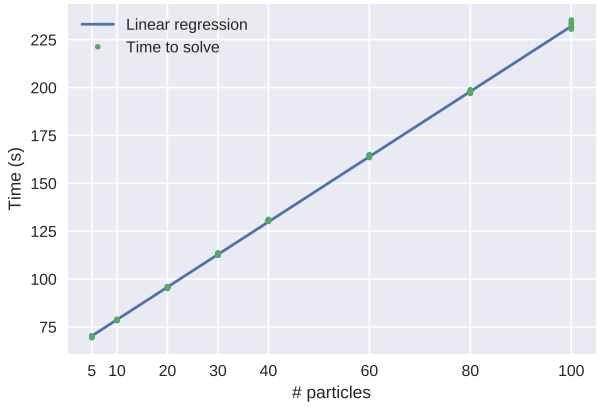


Figure 6.4: Computation time versus number of particles with $f_{res} = 1.0$ and $r_{th} = 2.0$.

Figure 6.4 shows the computation time from different number of particles over 10 iterations each. From this data, the maximum number of particles that can be used in the breakwater dataset (876 s) can be extrapolated to 478 (with $f_{res} = 1.0$ and $r_{th} = 2.0$). However this is a value that cannot be known beforehand and lower particle counts must be selected to ensure online computation of H-SLAM.

In conclusion f_{res} and r_{th} have to be tuned according to the level of detail of the environment to explore, and then a number of particles that ensures online processing must be selected.

7

Conclusions & Future Work

This chapter concludes this thesis by reviewing the main contributions in Section 7.1 and discussing compelling areas for future work in Section 7.2.

7.1 Conclusions and Contributions

In this thesis we have presented two algorithms for online localization. The first one locates a beacon using range-only measurements, while the second provides continuous occupancy maps of its environment. We divided the outcomes of these algorithms in the following contributions:

Online Range-Only Localization for Homing We have proposed two SOG-based methods for range-only localization (SOG-DR and SOG-SLAM). Given the high-nonlinearity of the problem, the SOG filter has the advantages of both the PF and the EKF filters, to provide correct localization for homing (PFs can deal with non-linear problems, and EKFs cover more space than a single particle). Quantitative comparisons between them showed similar performance in localization.

SOG initialization through Geodesic Grids We have proposed a initialization method for SOG filters to ensure full coverage of equally distributed Gaussians in a sphere given a single range measurement. Moreover, it allows to select different resolutions depending on the available computational power to ensure online computations of the SOG filter.

SOG equivalent Gaussian with Unscented Transform We have proposed a fast method to evaluate the precision of a SOG representation by means of the Unscented Transform. An fast approximated equivalent Gaussian is computed to evaluate the SOG uncertainty. This was useful to check the localization uncertainty that provided the SOG filter to determine when a beacon can be considered correctly localized or still requires more range measurements from different positions.

Range-Only Active Localization We have proposed an AL algorithm on top of the Online Range-Only Localization for Homing. Given a set of possible actions, AL choses the action that further decreases the SOG entropy and therefore its uncertainty. AL has proven useful to rapidly and autonomously locate an acoustic modem without human intervention.

Integration with other localization methods Online Range-Only Localization has been integrated with a visual based method to develop a reliable full homing and docking procedure. Range-Only Localization has been accurate enough even in bad visibility conditions where the visual method required close proximity and therefore a precise location of the DS.

Hilbert Maps We have brought HM representation to the underwater domain for the first time in literature. The use of this continuous occupancy representation greatly reduces the memory requirements compared to traditional OG while maintaining

$O(1)$ computational complexity. A raycasting method was developed to be able to compare real range measurements with expected measurements in the HM.

H-SLAM We have proposed a new SLAM framework combining RBPF with HM representation that has obtained correct reconstructions of 2D environments using a sonar sensor.

Experimental evaluation This thesis has provided multiple experimental evaluations of the proposed methods, demonstrating their applicability on real data from different vehicles. Range-Only Localization was tested in real-time, while H-SLAM was tested online on real data gathered in real-world environments.

7.2 Future work

This thesis cannot be considered a final and definitive solution for AUV range-only localization and occupancy mapping. However, it does contribute a further step towards better and more capable AUVs. In doing so, this thesis has established the basis for challenging future work that will continue extending AUV capabilities.

Long-term deployment of an AUV Although range-only localization and the complete homing and docking strategy have been thoroughly tested, we think that a more long-term deployment should be tested in order to assess reliability of the system without any human intervention. However, to be able to run these tests, it is imperative to have a DS that is able to charge the AUV's batteries. That will clearly demonstrate true autonomy for long term deployment operations and open new uses for AUVs such as continuous monitoring and inspection.

H-SLAM framework for full 3D environments Although HM representation supports both 2D and 3D environments, its capability was only evaluated for 2D real-world datasets. The author's current effort focuses on implementing H-SLAM in 3D and demonstrating it on a full 3D dataset.

Filter outliers in sonar range measurements In Chapter 5, we observed sonar measurements suffer from multi-path problems caused by the geometry of the environment. A sonar measurements filter is needed to ensure that outliers are discarded and never incorporated in a map. This is quite a challenging problem that we are trying to solve by temporally checking the consistency of measurements.

Test other sonar sensors with H-SLAM framework In Chapter 5, only a profiler sonar was used to construct the maps. We think that the use of other sensors, e.g. a multibeam sonar can improve the precision of map representation and easily represent 3D environments, specially if coupled with a pan&tilt unit.

Real-time H-SLAM in the AUV In Chapter 5, all experiments related to H-SLAM were tested online on previously recorded data. The next logical step is to test the algorithms in real-time.

Use H-SLAM framework with online path planning The combination of a continuous occupancy representation with online path planning methods can greatly improve the autonomy of AUVs (e.g. enabling autonomous exploration of full 3D environments like underwater caves).

Bibliography

[Amanatides et al., 1987] Amanatides, J., Woo, A., et al. (1987). A fast voxel traversal algorithm for ray tracing. In *Eurographics*, volume 87, pages 3–10.

[Amat et al., 1999] Amat, J., Monferrer, A., Batlle, J., and Cufi, X. (1999). GARBI: a low-cost underwater vehicle. *Microprocessors and Microsystems*, 23(2):61–67.

[Ancuti et al., 2017] Ancuti, C. O., Ancuti, C., Vleeschouwer, C. D., Neumann, L., and Garcia, R. (2017). Color transfer for underwater dehazing and depth estimation. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 695–699.

[Arsenio and Ribeiro, 1998] Arsenio, A. and Ribeiro, M. I. (1998). Active range sensing for mobile robot localization. In *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on*, volume 2, pages 1066–1071. IEEE.

[Aulinas et al., 2010] Aulinas, J., Lladó, X., Salvi, J., and Petillot, Y. R. (2010). Feature based SLAM using Side-Scan salient objects. In *OCEANS*, pages 1–8. IEEE.

[Aykin and Negahdaripour, 2013] Aykin, M. D. and Negahdaripour, S. (2013). On Feature Matching and Image Registration for Two-dimensional Forward-scan Sonar Imaging. *Journal of Field Robotics*, 30(4):602–623.

[Barkby et al., 2011] Barkby, S., Williams, S. B., Pizarro, O., and Jakuba, M. V. (2011). A featureless approach to efficient bathymetric SLAM using distributed particle mapping. *Journal of Field Robotics*, 28(1):19–39.

[Barkby et al., 2012] Barkby, S., Williams, S. B., Pizarro, O., and Jakuba, M. V. (2012). Bathymetric particle filter SLAM using trajectory maps. *The International Journal of Robotics Research*, 31(12):1409–1430.

[Batlle et al., 2005] Batlle, J., Ridao, P., Garcia, R., Carreras, M., Cufi, X., El-Fakdi, A., Ribas, D., Nicosevici, T., Batlle, E., Oliver, G., et al. (2005). URIS: Underwater Robotic Intelligent System. *Automation for the Maritime Industries*, pages 177–203.

[Baumgartner and Wales, 2006] Baumgartner, L. J. and Wales, N. S. (2006). *Assessment of a Dual-frequency Identification Sonar (DIDSON) for application in fish migration studies*. NSW Department of Primary Industries.

[Bingham et al., 2010] Bingham, B., Foley, B., Singh, H., Camilli, R., Delaporta, K., Eustice, R., Mallios, A., Mindell, D., Roman, C., and Sakellariou, D. (2010). Robotic tools for deep water archaeology: Surveying an ancient shipwreck with an autonomous underwater vehicle. *Journal of Field Robotics*, 27(6):702–717.

[Bishop et al., 2007] Bishop, A. N., Fidan, B., Anderson, B. D., Dogancay, K., and Pathirana, P. N. (2007). Optimality analysis of sensor-target geometries in passive localization: Part 1-Bearing-only localization. In *Intelligent Sensors, Sensor Networks and Information, 2007. ISSNIP 2007. 3rd International Conference on*, pages 7–12. IEEE.

[Blanco et al., 2008a] Blanco, J.-L., Fernandez-Madrigal, J.-A., and González, J. (2008a). Efficient probabilistic range-only slam. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 1017–1022. IEEE.

[Blanco et al., 2008b] Blanco, J.-L., González, J., and Fernandez-Madrigal, J.-A. (2008b). A pure probabilistic approach to range-only SLAM. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 1436–1441. IEEE.

[Bosch et al., 2015] Bosch, J., Gracias, N., Ridao, P., and Ribas, D. (2015). Omnidirectional Underwater Camera Design and Calibration. *Sensors*, 15(3):6033–6065.

[Campos et al., 2016] Campos, R., Gracias, N., and Ridao, P. (2016). Underwater Multi-Vehicle Trajectory Alignment and Mapping Using Acoustic and Optical Constraints. *Sensors*, 16(3):387.

[Carrera et al., 2015] Carrera, A., Palomeras, N., Hurtós, N., Kormushev, P., and Carreras, M. (2015). Cognitive system for autonomous underwater intervention. *Pattern Recognition Letters*, 67:91 – 99. Cognitive Systems for Knowledge Discovery.

[Carreras et al., 2013] Carreras, M., Candela, C., Ribas, D., Mallios, A., Magí, L., Vidal, E., Palomeras, N., and Ridao, P. (2013). Sparus II, design of a lightweight hovering AUV. In *5th International Workshop on Marine Technology (Martech)*. SARTI.

[Carreras et al., 2018] Carreras, M., Hernández, J. D., Vidal, E., Palomeras, N., Ribas, D., and Ridao, P. (2018). Sparus II AUV - A Hovering Vehicle for Seabed Inspection. *IEEE Journal of Oceanic Engineering*.

[Carrillo et al., 2012] Carrillo, H., Reid, I., and Castellanos, J. A. (2012). On the comparison of uncertainty criteria for active SLAM. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 2080–2087. IEEE.

[Centelles et al., 2017] Centelles, D., Moscoso, E., Vallicrosa, G., Palomeras, N., Sales, J., Martí, J. V., Marín, R., Ridao, P., and Sanz, P. J. (2017). Wireless HROV control with compressed visual feedback over an acoustic link. In *OCEANS 2017 - Aberdeen*, pages 1–7.

[Chen et al., 2011] Chen, J., Gong, Z., Li, H., and Xie, S. (2011). A detection method based on sonar image for underwater pipeline tracker. In *Second International Conference on Mechanic Automation and Control Engineering (MACE)*, pages 3766–3769.

[Cieslak et al., 2015] Cieslak, P., Ridao, P., and Giergiel, M. (2015). Autonomous underwater panel operation by GIRONA500 UVMS: A practical approach to autonomous underwater manipulation. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 529–536.

[Desa et al., 2006] Desa, E., Madhan, R., and Maurya, P. (2006). Potential of autonomous underwater vehicles as new generation ocean data platforms. *Current Science*, 90(9):1202–1209.

[Elibol et al., 2011] Elibol, A., Gracias, N., Garcia, R., Gleason, A., Gintert, B., Lirman, D., and Reid, P. (2011). Efficient autonomous image mosaicing with applications to coral reef monitoring. In *IROS 2011 workshop on robotics for environmental monitoring*.

[Escartín et al., 2008] Escartín, J., Garcia, R., Delaunoy, O., Ferrer, J., Gracias, N., Elibol, A., Cufi, X., Neumann, L., Fornari, D. J., Humphris, S. E., et al. (2008). Globally aligned photomosaic of the Lucky Strike hydrothermal vent field (Mid-Atlantic Ridge, 37 18.5 N): Release of georeferenced data, mosaic construction, and viewing software. *Geochemistry, Geophysics, Geosystems*, 9(12).

[Eustice et al., 2002] Eustice, R., Pizarro, O., Singh, H., and Howland, J. (2002). UWIT: Underwater Image Toolbox for optical image processing and mosaicking in MATLAB. In *Underwater Technology, 2002. Proceedings of the 2002 International Symposium on*, pages 141–145. IEEE.

[Eustice et al., 2005] Eustice, R., Singh, H., Leonard, J. J., Walter, M. R., and Ballard, R. (2005). Visually Navigating the RMS Titanic with SLAM Information Filters. In *Robotics: Science and Systems*, volume 2005, pages 57–64.

[Fabresse et al., 2013] Fabresse, F. R., Caballero, F., Maza, I., and Ollero, A. (2013). Undelayed 3D RO-SLAM based on Gaussian-mixture and reduced spherical parametrization. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 1555–1561. IEEE.

[Fairfield et al., 2007] Fairfield, N., Kantor, G., and Wettergreen, D. (2007). Real-Time SLAM with Octree Evidence Grids for Exploration in Underwater Tunnels. *Journal of Field Robotics*, 24(1-2):03–21.

[Gadre, 2007] Gadre, A. S. (2007). *Observability analysis in navigation systems with an underwater vehicle application*. PhD thesis, Virginia Tech.

[Gadre and Stilwell, 2004] Gadre, A. S. and Stilwell, D. J. (2004). Toward underwater navigation based on range measurements from a single location. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 5, pages 4472–4477. Ieee.

[Galceran Yebenes, 2014] Galceran Yebenes, E. (2014). *Coverage path planning for autonomous underwater vehicles*. PhD thesis, Universitat de Girona.

[Gracias et al., 2013] Gracias, N., Ridao, P., Garcia, R., Escartin, J., L'Hour, M., Cibecchini, F., Campos, R., Carreras, M., Ribas, D., Palomeras, N., et al. (2013). Mapping the Moon: Using a lightweight AUV to survey the site of the 17th century ship 'La Lune'. In *MTS/IEEE OCEANS-Bergen*, pages 1–8. IEEE.

[Gracias et al., 2003] Gracias, N. R., Van Der Zwaan, S., Bernardino, A., and Santos-Victor, J. (2003). Mosaic-based navigation for autonomous underwater vehicles. *IEEE Journal of Oceanic Engineering*, 28(4):609–624.

[Guizilini and Ramos, 2016] Guizilini, V. and Ramos, F. (2016). Large-scale 3D scene reconstruction with Hilbert Maps. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 3247–3254. IEEE.

[Guizilini and Ramos, 2017] Guizilini, V. C. and Ramos, F. T. (2017). Unsupervised Feature Learning for 3D Scene Reconstruction with Occupancy Maps. In *AAAI*, pages 3827–3833.

[Hata et al., 2016] Hata, A. Y., Wolf, D. F., and Ramos, F. T. (2016). Particle filter localization on continuous occupancy maps. In *International symposium on experimental robotics*, pages 742–751. Springer.

[Hawkes and Nehorai, 1998] Hawkes, M. and Nehorai, A. (1998). Acoustic vector-sensor beamforming and Capon direction estimation. *IEEE Transactions on Signal Processing*, 46(9):2291–2304.

[Hernández et al., 2011] Hernández, E., Carreras, M., Antich, J., Ridao, P., and Ortiz, A. (2011). A topologically guided path planner for an AUV using homotopy classes. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 2337–2343. IEEE.

[Hernández et al., 2015] Hernández, J., Vidal, E., Vallicrosa, G., Pairet, E., and Carreras, M. (2015). Simultaneous mapping and planning for autonomous underwater vehicles in unknown environments. In *OCEANS 2015 Genova*, pages 1–6.

[Hernández et al., 2017] Hernández, J. D., Vidal, E., Greer, J., Fiasco, R., Jaussaud, P., Carreras, M., and García, R. (2017). AUV online mission replanning for gap filling and target inspection. In *OCEANS 2017-Aberdeen*, pages 1–4. IEEE.

[Hernández et al., 2015a] Hernández, J., Vidal, E., Vallicrosa, G., Galceran, E., and Carreras, M. (2015a). Online path planning for autonomous underwater vehicles in unknown environments. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1152–1157.

[Hernández et al., 2016] Hernández, J. D., Istenič, K., Gracias, N., Palomeras, N., Campos, R., Vidal, E., García, R., and Carreras, M. (2016). Autonomous Underwater Navigation and Optical Mapping in Unknown Natural Environments. *Sensors*, 16(8).

[Hernández et al., 2015b] Hernández, J. D., Vallicrosa, G., Vidal, E., Èric Pairet, Carreras, M., and Ridao, P. (2015b). On-line 3D Path Planning for Close-proximity Surveying with AUVs. *IFAC-PapersOnLine*, 48(2):50 – 55. 4th IFAC Workshop onNavigation, Guidance and Controlof Underwater VehiclesNGCUV 2015.

[Hornung et al., 2013] Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., and Burgard, W. (2013). OctoMap: an efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 34(3):189–206.

[Hurtós et al., 2015] Hurtós, N., Ribas, D., Cufí, X., Petillot, Y., and Salvi, J. (2015). Fourier-based Registration for Robust Forward-looking Sonar Mosaicing in Low-visibility Underwater Environments. *Journal of Field Robotics*, 32(1):123–151.

[Hurtós et al., 2017] Hurtós, N., Mallios, A., Palomeras, N., Bosch, J., Vallicrosa, G., Vidal, E., Ribas, D., Gracias, N., Carreras, M., and Ridao, P. (2017). LOON-DOCK: AUV homing and docking for high-bandwidth data transmission. In *OCEANS 2017 - Aberdeen*, pages 1–7.

[Istenic et al., 2017] Istenic, K., Ila, V., Polok, L., Gracias, N., and García, R. (2017). Mission-time 3D reconstruction with quality estimation. In *OCEANS 2017 - Aberdeen*, pages 1–9.

[Johnson-Roberson et al., 2010] Johnson-Roberson, M., Pizarro, O., Williams, S. B., and Mahon, I. (2010). Generation and visualization of large-scale three-dimensional reconstructions from underwater robotic surveys. *Journal of Field Robotics*, 27(1):21–51.

[Kim et al., 2005] Kim, K., Neretti, N., and Intrator, N. (2005). Mosaicing of acoustic camera images. *IEE Proceedings-Radar, Sonar and Navigation*, 152(4):263–270.

[Kim and Kim, 2013] Kim, S. and Kim, J. (2013). Continuous occupancy maps using overlapping local Gaussian processes. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4709–4714.

[Kinsey et al., 2006] Kinsey, J., Eustice, R., and Whitcomb, L. (2006). A Survey of Underwater Vehicle Navigation: Recent Advances and New Challenges. In *Proceedings*

of the 7th IFAC Conference on Manoeuvring and Control of Marine Crafts, Lisbon, Portugal.

[Li and Nehorai, 2011] Li, T. and Nehorai, A. (2011). Maximum likelihood direction-of-arrival estimation of underwater acoustic signals containing sinusoidal and random components. *IEEE Transactions on Signal Processing*, 59(11):5302–5314.

[Mallios et al., 2014] Mallios, A., Ridao, P., Ribas, D., and Hernández, E. (2014). Scan matching SLAM in underwater environments. *Autonomous Robots*, 36(3):181–198.

[Mandić et al., 2016] Mandić, F., Mišković, N., Palomeras, N., Carreras, M., and Vallicrosa, G. (2016). Mobile beacon control algorithm that ensures observability in single range navigation. *IFAC-PapersOnLine*, 49(23):48 – 53. 10th IFAC Conference on Control Applications in Marine SystemsCAMS 2016.

[Moravec, 1988] Moravec, H. P. (1988). Sensor fusion in certainty grids for mobile robots. *AI magazine*, 9(2):61.

[Moreno-Salinas et al., 2012] Moreno-Salinas, D., Pascoal, A. M., and Aranda, J. (2012). Surface sensor networks for Underwater Vehicle positioning with bearings-only measurements. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 208–214. IEEE.

[Negahdaripour et al., 2005] Negahdaripour, S., Firoozfam, P., and Sabzmeydani, P. (2005). On processing and registration of forward-scan acoustic video imagery. In *Computer and Robot Vision, 2005. Proceedings. The 2nd Canadian Conference on*, pages 452–459. IEEE.

[Newman and Leonard, 2003] Newman, P. and Leonard, J. (2003). Pure range-only subsea SLAM. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 2, pages 1921–1926. IEEE.

[Nicosevici et al., 2009] Nicosevici, T., Gracias, N., Negahdaripour, S., and Garcia, R. (2009). Efficient three-dimensional scene modeling and mosaicing. *Journal of Field Robotics*, 26(10):759–788.

[NOAA, 2018] NOAA (2018). Oceans & Coasts — National Oceanic and Atmospheric Administration. `http://www.noaa.gov/oceans-coasts`. [Online; accessed 10-April-2018].

[O'Callaghan et al., 2009] O'Callaghan, S., Ramos, F. T., and Durrant-Whyte, H. (2009). Contextual occupancy maps using Gaussian processes. In *2009 IEEE International Conference on Robotics and Automation*, pages 1054–1060.

[Olson et al., 2006] Olson, E., Leonard, J. J., and Teller, S. (2006). Robust range-only beacon localization. *Oceanic Engineering, IEEE Journal of*, 31(4):949–958.

[O'Callaghan and Ramos, 2012] O'Callaghan, S. T. and Ramos, F. T. (2012). Gaussian process occupancy maps. *The International Journal of Robotics Research*, 31(1):42–62.

[Palomer et al., 2016] Palomer, A., Ridao, P., and Ribas, D. (2016). Multibeam 3D Underwater SLAM with Probabilistic Registration. *Sensors*, 16(4).

[Palomer et al., 2018] Palomer, A., Ridao, P., Youakim, D., Ribas, D., Forest, J., and Petillot, Y. (2018). 3D Laser Scanner for Underwater Manipulation. *Sensors*, 18(4):1–14.

[Palomeras et al., 2016] Palomeras, N., Carrera, A., Hurtós, N., Karras, G. C., Bechlioulis, C. P., Cashmore, M., Magazzeni, D., Long, D., Fox, M., Kyriakopoulos, K. J., et al. (2016). Toward persistent autonomous intervention in a subsea panel. *Autonomous Robots*, 40(7):1279–1306.

[Palomeras et al., 2012] Palomeras, N., El-Fakdi, A., Carreras, M., and Ridao, P. (2012). COLA2: A control architecture for AUVs. *IEEE Journal of Oceanic Engineering*, 37(4):695–716.

[Palomeras et al., 2014] Palomeras, N., Peñalver, A., Massot-Campos, M., Vallicrosa, G., Negre, P. L., Fernández, J. J., Ridao, P., Sanz, P. J., Oliver-Codina, G., and Palomer, A. (2014). I-AUV docking and intervention in a subsea panel. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2279–2285.

[Pizarro et al., 2004] Pizarro, O., Eustice, R., and Singh, H. (2004). Large area 3D reconstructions from underwater surveys. In *OCEANS'04. MTTS/IEEE TECHNO-OCEAN'04*, volume 2, pages 678–687. IEEE.

[Prados et al., 2017] Prados, R., García, R., Gracias, N., Neumann, L., and Vagstol, H. (2017). Real-time fish detection in trawl nets. In *OCEANS 2017 - Aberdeen*, pages 1–5.

[Prats et al., 2012] Prats, M., Garcia, J., Wirth, S., Ribas, D., Sanz, P., Ridao, P., Gracias, N., and Oliver, G. (2012). Multipurpose autonomous underwater intervention: A systems integration perspective. In *20th Mediterranean Conference on Control & Automation (MED)*, pages 1379–1384. IEEE.

[Quigley et al., 2009] Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., and Ng, A. Y. (2009). ROS: an open-source Robot Operating System. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan.

[Ramos and Ott, 2016] Ramos, F. and Ott, L. (2016). Hilbert maps: scalable continuous occupancy mapping with stochastic gradient descent. *The International Journal of Robotics Research*, 35(14):1717–1730.

[Reader et al., 2002] Reader, G. T., Potter, J., and Hawley, J. G. (2002). The evolution of AUV power systems. In *OCEANS '02 MTS/IEEE*, volume 1, pages 191–198.

[Ribas et al., 2007] Ribas, D., Palomeras, N., Ridao, P., Carreras, M., and Hernandez, E. (2007). Ictineu AUV wins the first SAUC-E competition. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 151–156. IEEE.

[Ribas et al., 2012] Ribas, D., Palomeras, N., Ridao, P., Carreras, M., and Mallios, A. (2012). Girona 500 AUV: From Survey to Intervention. *IEEE/ASME Transactions on Mechatronics*, 17(1):46–53.

[Ribas et al., 2008] Ribas, D., Ridao, P., Tardós, J. D., and Neira, J. (2008). Underwater SLAM in man-made structured environments. *Journal of Field Robotics*, 25(11-12):898–921.

[Ridao et al., 2010] Ridao, P., Carreras, M., Ribas, D., and Garcia, R. (2010). Visual inspection of hydroelectric dams using an autonomous underwater vehicle. *Journal of Field Robotics*, 27(6):759–778.

[Ridao et al., 2015] Ridao, P., Carreras, M., Ribas, D., Sanz, P. J., and Oliver, G. (2015). Intervention AUVs: The next challenge. *Annual Reviews in Control*, 40:227 – 241.

[Roman and Singh, 2005] Roman, C. and Singh, H. (2005). Improved vehicle based multibeam bathymetry using sub-maps and SLAM. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 3662–3669. IEEE.

[Ross and Jouffroy, 2005] Ross, A. and Jouffroy, J. (2005). Remarks on the observability of single beacon underwater navigation. In *Proc. Intl. Symp. Unmanned Unteth. Subm. Tech.*

[Salinas, 2013] Salinas, D. M. (2013). *Adaptive sensor networks for mobile target localization and tracking.* PhD thesis, Universidad Nacional de Educación a Distancia. UNED.

[Short et al., 1999] Short, R. T., Fries, D. P., Toler, S. K., Lembke, C. E., and Byrne, R. H. (1999). Development of an underwater mass-spectrometry system for in situ chemical analysis. *Measurement Science and Technology*, 10(12):1195.

[Singh et al., 2004] Singh, H., Howland, J., and Pizarro, O. (2004). Advances in large-area photomosaicking underwater. *IEEE Journal of Oceanic Engineering*, 29(3):872–886.

[Song, 1996] Song, T. L. (1996). Observability of target tracking with bearings-only measurements. *IEEE Transactions on Aerospace and Electronic Systems*, 32(4):1468–1472.

[Song, 1999] Song, T. L. (1999). Observability of target tracking with range-only measurements. *IEEE Journal of Oceanic Engineering*, 24(3):383–387.

[Stachniss, 2009] Stachniss, C. (2009). Information Gain-based Exploration. In *Robotic Mapping and Exploration*, pages 143–160. Springer.

[Tena et al., 2003] Tena, I., Reed, S., Petillot, Y., Bell, J., and Lane, D. M. (2003). Concurrent Mapping and Localisation Using Side-scan Sonar for Autonomous Navigation. In *Proceedings of the 13th International Symposium on Unmanned Untethered Submersible Technology*, Durham, NH, USA.

[Thrun et al., 2005] Thrun, S., Burgard, W., Fox, D., et al. (2005). *Probabilistic robotics*, volume 1. MIT press Cambridge.

[Vaganay et al., 2000] Vaganay, J., Baccou, P., and Jouvencel, B. (2000). Homing by acoustic ranging to a single beacon. In *OCEANS 2000 MTS/IEEE Conference and Exhibition*, volume 2, pages 1457–1462. IEEE.

[Vallicrosa et al., 2016] Vallicrosa, G., Bosch, J., Palomeras, N., Ridao, P., Carreras, M., and Gracias, N. (2016). Autonomous homing and docking for AUVs using Range-Only Localization and Light Beacons. *IFAC-PapersOnLine*, 49(23):54–60. 10th IFAC Conference on Control Applications in Marine SystemsCAMS 2016.

[Vallicrosa et al., 2013a] Vallicrosa, G., Palomer, A., Ribas, D., and Ridao, P. (2013a). Realtime AUV Terrain Based Navigation with Octomap. In *XXXIV Jornadas de Automática*.

[Vallicrosa et al., 2013b] Vallicrosa, G., Palomer, A., Ribas, D., and Ridao, P. (2013b). *Realtime AUV Terrain Based Navigation with Octomap in a Natural Environment*, pages 41–53. Springer International Publishing.

[Vallicrosa et al., 2013c] Vallicrosa, G., Palomer, A., Ribas, D., and Ridao, P. (2013c). Towards realtime AUV SLAM with occupancy grids. In *Martech 2013 5th International Workshop on Marine Technology*, volume 15, pages 59–60.

[Vallicrosa et al., 2015] Vallicrosa, G., Ridao, P., and Ribas, D. (2015). AUV Single Beacon Range-Only SLAM with a SOG Filter. *IFAC-PapersOnLine*, 48(2):26 – 31. 4th IFAC Workshop onNavigation, Guidance and Controlof Underwater Vehicles (NGCUV2015).

[Vallicrosa et al., 2014] Vallicrosa, G., Ridao, P., Ribas, D., and Palomer, A. (2014). Active Range-Only beacon localization for AUV homing. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2286–2291.

[Vander Hook et al., 2014] Vander Hook, J., Tokekar, P., and Isler, V. (2014). Cautious Greedy Strategy for Bearing-only Active Localization: Analysis and Field Experiments. *Journal of Field Robotics*, 31(2):296–318.

[Walter et al., 2008] Walter, M., Hover, F., and Leonard, J. (2008). SLAM for ship hull inspection using exactly sparse extended information filters. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1463–1470. IEEE.

[Wang et al., 2013] Wang, S., Chen, L., Hu, H., and Gu, D. (2013). Single beacon based localization of AUVs using moving Horizon estimation. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 885–890. IEEE.

[Webster et al., 2009] Webster, S. E., Eustice, R. M., Singh, H., and Whitcomb, L. L. (2009). Preliminary deep water results in single-beacon one-way-travel-time acoustic navigation for underwater vehicles. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 2053–2060. IEEE.

[Williams and Mahon, 2004] Williams, S. and Mahon, I. (2004). Simultaneous localisation and mapping on the great barrier reef. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 2, pages 1771–1776. IEEE.

[Williams et al., 2010] Williams, S. B., Pizarro, O., Jakuba, M., and Barrett, N. (2010). AUV benthic habitat mapping in south eastern Tasmania. In *Field and Service Robotics*, pages 275–284. Springer.

[Youakim et al., 2017] Youakim, D., Ridao, P., Palomeras, N., Spadafora, F., Ribas, D., and Muzzupappa, M. (2017). MoveIt!: Autonomous Underwater Free-Floating Manipulation. *IEEE Robotics Automation Magazine*, 24(3):41–51.

[Zhang and Negahdaripour, 2010] Zhang, H. and Negahdaripour, S. (2010). EKF-based recursive dual estimation of structure and motion from stereo data. *IEEE Journal of Oceanic Engineering*, 35(2):424–437.