DOCTORAL THESIS 2022

DATA-

DRIVEN EXPLORATION OF BENTHIC ENVIRONMENTS

AUTONOMOUS
UNDERWATER VEHICLES
FEATURING
SEMANTIC
PERCEPTION
AND ADAPTIVE
NAVIGATION
INTELLIGENCE



Doctoral Thesis 2022

Doctoral Programme of Information and Communications Technology

DATA-DRIVEN EXPLORATION OF BENTHIC ENVIRONMENTS WITH AUTONOMOUS UNDERWATER VEHICLES FEATURING SEMANTIC PERCEPTION AND ADAPTIVE NAVIGATION INTELLIGENCE

Eric Guerrero Font

Thesis supervisor: Dr. Gabriel Oliver Codina Thesis supervisor: Dr. Francisco Jesús Bonin Font Thesis tutor: Dr. Gabriel Oliver Codina

Doctor by the Universitat de les Illes Balears

Declaration of Authorship

This Thesis has been submitted to the Escola de Doctorat, Universitat de les Illes Balears, in fulfillment of the requirements for the degree of *Doctor en Tecnologies de la Informació i les Comunicacions*. I hereby declare that, except where specific reference is made to the work of others, the content of this dissertation is entirely my own work, describes my own research and has not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university.

Eric Guerrero Font

Palma de Mallorca, March 6, 2023

Acknowledgements

The work presented in this document would not have been possible without the support of many people, to whom I want to address some lines of my deepest gratitude.

En primer lugar, me gustaría agradecer a mi familia por el apoyo incondicional que he recibido durante toda mi vida y que me ha dado impulso para llegar hasta este punto y el ímpetu para continuar mirando hacia adelante y perseguir nuevos retos. A mis padres y abuelas por animarme a explorar y a su vez por darme las herramientas para descubrir mi camino. A mis hermanos por compartir aventuras espaciales, por enseñarme a trascender el entorno y a compartir el aquí y el ahora. Finalment, agrair a n'Estel i a tots els amics que m'envolten, que són família, i que aporten una component constant d'equilibri i color al dia a dia.

En segon lloc, agrair als meus companys de batalla amb el Turbot AUV, a l'equip de robotica submarina. Als meus codirectors de tesi, al Gabriel Oliver per la seva motivació intrínseca per oferir ajuda i per compartir coneixements, i al Francesc Bonin pels consells, pels riures i per estar disponible en tot moment. Al Toni Martorell, company de peripècies, sempre atent i predisposat per emprendre. A la Yolanda Gonzalez, per la seva capacitat de engranar i iluminar el grup. Al Miguel Abadal, la Caterina Muntaner i el Bo Miquel Nordfeldt pel seu sí permanent i la integritat de voler fer les coses bé. Així com al Miquel Massot i al Pep Lluís Negre, ex-integrants del grup, referents i pares adoptius al començament d'aquesta aventura. Gràcies a tots per fer de la feina un repte y un gaudiment al mateix temps.

En tercer lloc, agrair a tots els companys del grup de recerca SRV i de la UIB en general que han donat soport tècnic i moral per el desenvolupament d'aquesta tesi. A tot l'equip de sistemes, especialment a la Inés Álvarez i l'Alberto Ballesteros. A tot l'equip de robòtica aèria, especialment a l'Emilio García, el Xisco Bonnín i el Joan Pep Company. A la Julia Navarro i a altres persones que han anat passant com ara el Giovanni i l'Andrea. Compartir infinits cafès, dinars i cerveses amb companys amb qui comparteixes camí professional crea una complicitat que fa que a la feina et sentis com a casa.

En quart lloc, voldria donar un especial agraïment a totes les persones que de forma esporàdica i sense ser conscients han contribuït de forma substancial al desenvolupament d'aquesta tesi. A la gent de l'IMEDEA, particularment a la Fiona Tomàs i la Julia Máñez per la multitud de campanyes de camp compartides. A la gent del CIRS, pel recolzament donat en la gestió d'incidencies amb el Turbot AUV. Y a Juan Poyatos, por su proactividad en el emprendimiento de proyectos y colaboración en experimentos de campo.

Finally, I would like to thank the anonymous reviewers that contributed to improve the different publications derived from this work, and all the institutions that have partially funded my research. With particular gratitude to the Government of the Balearic Islands, Spain that supported this research through grant FPI/2031/2017. (CAIB).

List of Figures

Spatial image resolution comparison	2
Autonomous navigation framework	3
Inspection and exploration framework schemes	4
Images from two benthic environments covered by $P.$ $oceanica$ seagrass	6
Image mosaic from a benthic environment covered by $P.$ oceanica seagrass	6
Example of optical spaceborne systems	10
Visual inspection of benthic environments by scuba divers	13
Lawn mower path followed for image recording with scuba divers	14
Example of tethered image systems	15
Example of AUVs	16
Bathymetry estimation via GP from Ma et al. [106]	18
RIG-tree framework from Hollinger et al. [89]	19
VGG16-FCN8 architecture used for online image segmentation	23
Online image processing to generate semantic point cloud data	24
Segmented images and semantic point clouds	24
	26
Robots used for the semantic modelling experiments	29
Cartographies from FT1 region	30
Sample images from FT1, FT2 and FT3	30
Cartographies from FT2 and FT3 regions	31
Resulting image segmentation for the three segmentation models considered	35
Comparison of the SGPMC models with $\phi_{M_{12}}$ or $\phi_{M_{32}}$ kernels	36
Meadow bounds representation of the raw data obtained from FT1	37
Results obtained for the validation of the SGPMC- $beta$ model with kernel $m12.\ldots$	38
Resulting maps of FT1 experiment	39
Resulting maps of FT2 experiment	39
Resulting maps of FT3 experiment	39
Decision-time adaptive replanning (DAR) method	41
One iteration of the proposed DF-MCTS algorithm	48
Two sets of action candidates	49
Prediction of the pre-trained SGP model used for testing the DF-MCTS strategy	52
Results obtained for the two DF-MCTS variations	53
Paths obtained after 20 DF-MCTS executions	53
Mean path information and path distance obtained	54
	Autonomous navigation framework. Inspection and exploration framework schemes. Images from two benthic environments covered by P . oceanica seagrass. Image mosaic from a benthic environment covered by P . oceanica seagrass. Example of optical spaceborne systems. Visual inspection of benthic environments by scuba divers Lawn mower path followed for image recording with scuba divers Example of tethered image systems. Example of AUVs. Bathymetry estimation via GP from Ma et al. [106]. RIG-tree framework from $Hollinger$ et al. [89]. VGG16-FCN8 architecture used for online image segmentation Online image processing to generate semantic point cloud data. Segmented images and semantic point clouds. Grid samples projected in the 3D space. Robots used for the semantic modelling experiments. Cartographies from FT1 region. Sample images from FT1, FT2 and FT3. Cartographies from FT2 and FT3 regions. Resulting image segmentation for the three segmentation models considered. Comparison of the SGPMC models with $\phi_{M_{12}}$ or $\phi_{M_{32}}$ kernels. Meadow bounds representation of the raw data obtained from FT1 Results obtained for the validation of the SGPMC-beta model with kernel $m12$. Resulting maps of FT1 experiment. Resulting maps of FT3 experiment. Resulting maps of FT3 experiment. Pocision-time adaptive replanning (DAR) method. One iteration of the proposed DF-MCTS algorithm. Two sets of action candidates. Prediction of the pre-trained SGP model used for testing the DF-MCTS strategy. Results obtained for the two DF-MCTS variations.

List of Figures iv

4.8	Aerial image and groundtruth used for testing the DAR method	. 55
4.9	Results obtained for an adaptive replanning in area A1	. 56
4.10	Executed path and resulting data density simulated in 2D the target areas $A1$ and $A2$.	. 58
4.11	Results obtained in 2D simulation in the target areas A1 and A2	. 59
4.12	Executed path and resulting data density simulated in 2D the target areas A3	. 60
4.13	Executed path and resulting data density simulated in 2D the target areas A4	61
4.14	Results obtained in 2D simulation in the target areas A3 and A4	62
5.1	Geo-localized image scheme	
5.2	Hovering AUVs	. 68
5.3	Navigation and communication payload of the AUV Turbot	. 69
5.4	Operation modes	. 70
5.5	USBL system	. 71
5.6	Localization layer.	. 73
5.7	In situ image recording experiments	. 75
5.8	Trilateration scheme	. 76
5.9	SDS experiment in detail	. 76
5.10	Ranging locations and distances used for the trilateration	. 77
5.11	Surge and sway speeds	. 77
5.12	First and last images of the static dataset SDS	. 78
5.13	Angular speeds	. 79
5.14	Depth and altitude.	. 79
5.15	Altitude measurements	. 80
5.16	Roll and pitch	. 81
5.17	Acoustic channel frequencies. From top to bottom SDS, DS1 and DS3	. 82
5.18	Absolute localization error	. 82
5.19	USBL measurements error	. 83
5.20	Pixel MSE distribution considering uncertainty from the position, scale, and orientation.	. 84
5.21	Blur distance example	. 85
5.22	Images recorded during the field-tests	. 86
5.23	Estimated blurring distances	. 87
5.24	Resulting pixel size	. 88
5.25	Correlation between the burring and the navigation	. 89
5.26	Correlation with the navigation descriptors	. 89
5.27	Correlation between surge, yaw rotation, altitude and the maximum pixel size	. 89
6.1	Data-driven exploration scheme	. 92
6.2	Threads of the AVIG architecture.	
6.2	StoneFish simulator.	
	Target area for the simulation experiments.	
6.4	•	
6.5	Synthetic simulation environment.	
6.6	Sample images recorded in simulation.	
6.7	Sequential spatial data generated in simulation	
6.8	Modelling time and coverage percent in simulation using DAR	
6.9	Speed in the forward direction.	
	Paths followed and raw data density obtained in simulation	
0.11	Results of the DAR, SBSRE* and LM in simulation	104

List of Figures v

6.12	Steps of the DAR method in simulation	105
6.13	Region of the CSJ experiments	107
6.14	Image examples form the CSJ experiment	108
6.15	Semantic point clouds from the CSJ experiment	109
6.16	Semantic point cloud data map from the CSJ experiment	109
6.17	Gathered data used for training	110
6.18	CSJ training samples histogram	111
6.19	Resulting modelling time and coverage percent with the AVIG framework	112
6.20	Path followed and raw data density obtained in field-test following DAR method \dots	112
6.21	Five different instants during the execution of the DAR method in field-test	113
6.22	Results obtained in field-test	114
6.23	Region of the PA experiments	114
6.24	Altitude and depth during the PA field-tests	116
6.25	Altitude measurement histograms during the PA field-tests	116
6.26	Field-test image examples form PA	117
6.27	Environment prediction for the PA field experiments	118
6.28	Training samples histogram for PA1 experiment	119
6.29	Training samples histogram for PA2 experiment	120
6.30	Replanning representations for the PA1 experiment	121
6.31	Replanning representations for the PA2 experiment	124
6.32	Results obtained during the PA field-tests	125

List of Tables

2.1	Spaceborne RS systems specifications	11
3.1	Specifications of the robots used for field-testing the semantic habitat modelling	28
3.2	Field-test specifications	29
3.3	Configurations used for low, medium and high resolution mapping	31
3.4	Results obtained to select the most suitable CNN model selection	34
3.5	Mean results of the 40-fold cross-validation tests for the SGP model selection \dots	36
3.6	Resulting AUC values obtained from field-tests FT1-3	37
4.1	Default parameter configurations for the DF-MCTS experiment	50
4.2	Specific parameter configurations for the DF-MCTS experiment	51
4.3	SGP configurations for the DAR experiment	51
4.4	Scale parameters used for the target areas A1-A4 of the 2D simulation experiments. $$	55
4.5	Configuration parameter the sampling-based strategy	57
5.1	AUV Turbot payload	68
5.2	Communication and absolute position depending on the operation mode	70
5.3	Measurements used for the EKF Localization filter.	74
5.4	Mean value and standard deviation of the sway speed estimation for each dataset. $\ \ldots \ \ldots$	78
5.5	Altitude error distributions	80
5.6	Pose estimation results	83
5.7	Uncertainty results from navigation analysis	84
6.1	Configuration of the DAR method in simulation	98
6.2	SBSRE* scaled parameters for the simulation experiment	96
6.3	Configuration of the LOSCTE controller used for navigation	96
6.4	Modelling thread configuration	100
6.5		106
6.6	Navigation configuration for the CSJ region	107
6.7	Specific replanning and modelling configurations for the CSJ experiment	
6.8	Navigation configuration for the PA region	115
6.9	Specific replanning and modelling configurations for the PA experiment	115

List of Abbreviations

AUC Area under the curve

AUV Autonomous underwater vehicle

AVIG Adaptive visual information gathering

BCM Bayesian committee machineBHM Benthic habitat mappingCNN Convolutional neural network

CP Coverage percentCS Control stationCV Cross-validation

DAR Decision-time adaptive replanning

DE Differential entropy

DF Depth-first

DP Dynamic programmingDVL Doppler velocity logEKF Extended Kalman filter

ES Echosounder
GP Gaussian process

GPS Global positioning system

GT ground truth

IG Information gathering

IGN Instituto Geodésico nacional
 IMU Inertial measurement unit
 IPP Informative path planning
 IG Information gathering

IS In situ

LBS Long baseline

LOSCTE Line of sight with cross tracking error

MBES Multibeam echosounder

MC Monte Carlo

MCMC Markov chain Monte CarloMCTS Monte Carlo tree search

MDE Mean of the differential entropy

MI Mutual information
NED North east down

PNOA Plan nacional de ortofotografía aérea

MDP Markov decision process

RF Radio frequency

RL Reinforcement learning

ROC Receiver operating characteristics

ROS Robot operating system
ROV Robot operated vehicles

RRT Rapidly-exploring random trees

RS Remote sensing
RTK Real time kinematic
SBES Single beam echosounder

SDE Standard deviation of the differential entropy

SGP Sparse Gaussian process

SGPMC Sparse variational GP using MCMC
 SGPR Sparse variational GP regression
 SP Spearman correlation index

Side scan sonar

SVGP Sparse variational GP

TD Temporal difference

UAS Unmanned aerial system

UAV Unmanned aerial vehicle

UCB Upper confidence bound

USBL Ultra short baseline

Contents

A	know	wledgements	i					
Al	ostrac	act	xi					
Re	esum	a a	xii					
Re	esume	nen	xiv					
1	Intr	croduction	1					
	1.1	Objectives	1					
	1.2	Motivation	2					
		1.2.1 Change of paradigm: from inspection to exploration	3					
		1.2.2 Towards automating underwater exploration	5					
	1.3	Context	5					
		1.3.1 Seagrass ecosystems	5					
		1.3.2 Local interest	7					
	1.4	Document overview	7					
	1.5	Related publications	8					
2	Related work							
	2.1	Benthic data acquisition	9					
		2.1.1 Remote sensing	9					
		2.1.2 <i>In situ</i> sampling	13					
	2.2	Data-driven exploration with AUVs	16					
		2.2.1 Information Gathering	17					
		2.2.2 Perception	17					
		2.2.3 Decision-making	17					
3	Sem	mantic habitat modelling	20					
	3.1	Sensing	21					
		3.1.1 Image gathering	21					
		3.1.2 Semantic segmentation	22					
		3.1.3 Image projection	24					
	3.2	Spatial modelling	25					
		3.2.1 Training and prediction	25					
	3.3	Experiments	28					
		3.3.1 Robot configurations	28					

Contents x

		3.3.2	Field-tests	29
		3.3.3	Resolution parameters	31
		3.3.4	Model selection criteria	31
		3.3.5	Validation criteria	33
	3.4	Results	5	34
		3.4.1	Model selection	34
		3.4.2	Validation	37
	3.5	Discuss	sion	10
4	D			4-1
4	4.1			11 12
	4.1			12
			1 /	13
			1 0	ьэ 14
	4.0		1 0	
	4.2			15
		4.2.1		15
				16
				17
	4.3	-		17
		4.3.1		17
				18
	4.4	•		0
				51
				52
				64
	4.5	Conclu	sions	61
5	Auto	onomous	s navigation 6	64
	5.1	Data u	ncertainty	55
	5.2	Naviga	tion framework	57
		5.2.1	AUV Turbot	i 8
		5.2.2	Localization filters	72
	5.3	Experi	ments	4
		5.3.1	Missions	75
		5.3.2	Absolute localization groundtruth	6
	5.4	Results	5	7
		5.4.1	General navigation	7
		5.4.2	Absolute localization	31
		5.4.3	Data quality assessment	33
	5.5	Discuss	sion	90
6	Data	a-driven	exploration 9	92
	6.1		•	93
		_		93
		6.1.2	•)4
	6.2	Simula		96
			-	96

Contents xi

			Configurations	
			Results	
	6.3	Field 6	experiments	106
		6.3.1	Colonia de Sant Jordi (CSJ)	107
		6.3.2	Port d'Andratx (PA)	113
	6.4	Discus	ssion	122
7	Con	clusions	s	126
	7.1	Summ	nary	126
	7.2	Contri	ibutions	127
	7.3	Future	e Work	128
D:	1. 12	raphy		130

Abstract

Most of the land terrain on Earth is being constantly mapped using sub-meter resolution spaceborne images. However, about eighty percent of the oceans' seafloor remains unexplored and most of the benthic habitat and geological structure distribution continues unknown. The main problem is that the methods that work on land are not applicable underwater. Remote sensing (RS) methods using spaceborne or airborne images can be used to map shallow water environments up to 10m depth and acoustic RS methods using multibeam echosounders are now providing increased resolutions on the backscatter data that can be used for benthic habitat mapping (BHM). Nonetheless, the feature richness of data collected with these types of methods is very limited. Most of the methods used for BHM require the use of in situ (IS) data in order to validate or even train their mapping algorithms.

This Thesis aims to facilitate the acquisition of IS data by pushing the boundaries of autonomous underwater vehicles (AUVs). Nowadays, AUVs are increasingly used to acquire ocean data, bridging the gap between the use of remote operated vehicles (ROVs) and shipborne acoustic RS. However, their autonomy is usually limited by the condition of following preprogrammed paths, their behavior is blind with respect to benthic data collected and are autonomous only in the sense that they are not tethered and are able to estimate their location and control their motion. This Thesis presents three novel methods to (1) provide an online semantic perception of the environment by processing the online data flow and building a probabilistic model of the benthic environment, (2) enlarge the decision-making autonomy of AUVs providing an adaptive capacity to replan mission paths based on a semantic understanding of the physical variable under study that depends on the objective of the campaign, (3) improve the autonomous navigation for in situ image recording of AUVs.

All in all this thesis build a data-driven exploration architecture that automates the *in situ* sampling process on unknown environments while maximizing data informativeness.

The algorithms described in this Thesis have been extensively validated in field using an AUV equipped with a stereo camera rig used to gather images of the seabed partially covered by *Posidonia oceanica* seagrass meadows.

Resum

La major part del terreny terrestre es continuament cartografiant mitjançant l'ús d'imatges satelitals de resolució submètrica. Tot i això, un vuitanta per cent del fons marí dels oceans encara no ha estat explorat. El problema principal és que els mètodes que funcionen a terra no són aplicables sota l'aigua. Els mètodes basats en tècniques de *Remote Sensing* (RS) que utilitzen imatges satel·litàries o aèries poden utilitzar-se per cartografiar entorns d'aigües poc profundes, i els mètodes de RS que utilitzen ecosondes acústiques multifeix proporcionen una resolució suficient per cartografiar hàbitats bentònics (BHM) a majors profundidats. Tot i això, la riquesa de les dades recollides amb aquests tipus de mètodes és molt limitada. La majoria dels mètodes utilitzats per a BHM requereixen l'ús de dades *in situ* (IS) per validar o fins i tot entrenar els algorismes.

L'objectiu d'aquesta tesi és facilitar l'adquisició de dades IS millorant les característiques que ofereixen els vehicles submarins autònoms (AUV) actuals. Avui dia, els AUVs s'utilitzen cada cop més per adquirir dades oceàniques, salvant la distància entre l'ús de vehicles operats a distància (ROVs) i mètodes de RS acústics. No obstant això, la seva autonomia sol estar limitada per la necessitat de seguir trajectòries preprogramades, el seu comportament no té en compte les dades bentòniques recollides durant la navegació. Són autònoms tan sols en el sentit que no requereixen connexió cablejada i són capaços d'estimar la seva ubicació i controlar el seu moviment. Aquesta Tesi presenta tres nous desenvolupaments per a (1) proporcionar una percepció semàntica de l'entorn mitjançant el processament de dades recollides a temps real i la construcció d'un model probabilístic de l'entorn bentònic, (2) ampliar l'autonomia en la presa de decisions dels AUVs proporcionant una capacitat d'adaptació per replanificar les rutes de la missió basant-se en una comprensió semàntica de l'entorn, i (3) millorar la navegació autònoma per al gravat dímatges in situ dels AUVs. Finalment, els desenvolupaments esmentats s'utilitzen per construir una arquitectura d'exploració adaptativa que automatitza el procés de mostreig in situ en entorns desconeguts alhora que maximitza la informació de les dades recollides.

Els mètodes descrits en aquesta Tesi han estat àmpliament validats mitjançant assajos de camp utilitzant un AUV equipat amb una càmera estereoscòpica utilitzada per recollir imatges del fons marí parcialment cobert per praderies de *Posidonia oceanica*.

Resumen

La mayor parte del terreno terrestre es continuamente cartografiado mediante el uso de imágenes satelitales de resolución submétrica. Sin embargo, un ochenta por ciento del fondo marino de los océanos aún no ha sido explorado. El principal problema es que los métodos que funcionan en tierra no son aplicables bajo el agua. Los métodos basados en técnicas de *Remote Sensing* (RS) que utilizan imágenes satelitales o aéreas pueden utilizarse para cartografiar entornos de aguas poco profundas, y los métodos de RS que utilizan ecosondas acústicas multihaz proporcionan una resolución suficiente para cartografiar habitats bentónicos (BHM) a mayores profundidades. Sin embargo, la riqueza de los datos recogidos con este tipo de métodos es muy limitada. La mayoría de los métodos utilizados para BHM requieren el uso de datos *in situ* (IS) para validar o incluso entrenar los algoritmos.

El objetivo de esta tesis es facilitar la adquisición de datos IS mejorando las características que ofrecen los vehículos submarinos autónomos (AUV) actuales. Hoy en día, los AUVs se utilizan cada vez más para adquirir datos oceánicos, salvando la distancia entre el uso de vehículos operados a distancia (ROVs) y métodos de RS acústicos. Sin embargo, su autonomía suele estar limitada por la necesidad de seguir trayectorias preprogramadas, su comportamiento no tiene en cuenta los datos bentónicos recogidos durante la navegación. Son autónomos tan solo en el sentido de que no requieren conexión cableada y son capaces de estimar su ubicación y controlar su movimiento. Esta Tesis presenta tres novedosos desarrollos para (1) proporcionar una percepción semántica del entorno mediante el procesamiento del datos recojidos a tiempo real y la construcción de un modelo probabilístico del entorno bentónico, (2) ampliar la autonomía en la toma de decisiones de los AUVs proporcionando una capacidad de adaptación para re-planificar las rutas de la misión basándose en una comprensión semántica del entorno, y (3) mejorar la navegación autónoma para el grabado de imágenes in situ de los AUVs. Por último, los desarrollos mencionados se utilizan para construir una arquitectura de exploración adaptativa que automatiza el proceso de muestreo in situ en entornos desconocidos al tiempo que maximiza la información de los datos recogidos.

Los métodos descritos en esta Tesis han sido ampliamente validados mediante ensayos de campo utilizando un AUV equipado con una cámara estereoscópica utilizada para recoger imágenes del fondo marino parcialmente cubierto por praderas de *Posidonia oceanica*.

Chapter 1

Introduction

1.1 Objectives

This Thesis has the main objective of developing a set of methods to enable data-driven exploration of benthic environments with AUVs. The rationale for this research is the need to improve the current methods used for recording high-resolution underwater data.

AUVs have proved to be a very effective platform for underwater inspection. However, actual limitations on underwater communications slow down underwater exploration, which usually require a human on the loop for data post-processing and mission planning.

The main goals of the data-driven exploration framework proposed in this Thesis are:

- Eliminate, or at least reduce, human intervention in underwater exploration campaigns.
- Maximizing the time that an AUV spends to data recording during field campaigns.
- Maximizing the quality of the recorded data by reducing their uncertainty and increasing their informativeness.

The methods described in this Thesis aim to improve the quality and extension of the recorded data by proposing (a) an enhanced perception capacity for modelling the environment, (b) a decision-making capacity to enable adaptive mission replanning, (c) an improved untethered navigation method to increase the vehicle autonomy, and (d) a data-driven exploration framework that joins (a), (b) and (c).

This Thesis targets the *in situ* data gathering and processing of geo-referenced images in order to generate spatial distribution data for benthic habitat mapping (BHM), with the focus set on the identification of *P. oceanica* seagrass meadows.

Although the testbed of the research has been the *P. oceanica*, this Thesis aims to provide a modular framework that can be adapted to gather data of particular processes with different spatial distributions, using disctint sensors and strategies. The common goal of the methods proposed is the generation of high-resolution datasets that try to get an informative representation of a process in an accurate and time efficient manner. Targeting long-lasting unsupervised monitoring, with communication limitations and no (or uncertain) prior environmental knowledge.

1.2 Motivation

Most of the Earth's terrestrial terrain is constantly mapped with sub-meter resolution imagery collected by Worldview and PlanetScope spaceborne systems, the Moon surface has been recently mapped with a 20-60m/px resolution using multiple sources of data [1], and the HiRISE has been providing 0.3m/px resolution images from Mars surface for the last 15 years. However, about an eighty percent of the oceans' seafloor remains unexplored, and only a 10%-15% has been mapped considering spatial resolutions up to 100m/px. There are emerging coordinated efforts to map the world oceans' depth for 2030 [2]. However, such efforts only consider bathymetric mapping using acoustic remote sensing, with expected spatial resolutions ranging from 100 to 800m/px depending on the sampling depth. Figure 1.1 provides an illustration of the loss of information when gathering data as such spatial resolutions. The figure shows the downscaling of a raw image of Mars recorded from the HiRISE satellite and a raw image of the Earth recorded from the WorldView-3; from left to right the original image and the downsampling to 100m/px and 800m/px. Notice the loss of very relevant detail.

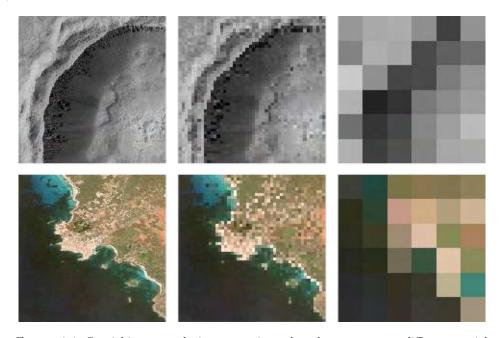


FIGURE 1.1: Spatial image resolution comparison; the columns represent different spatial resolutions (original, 100m/px and 800m/px) and the rows represent different images. The top row represents an image from Mars surface taken from the HiRISE satellite, the bottom row represents an image from the Earth gathered with the WorldView-3 satellite. Notice how the loss of resolution affects the image interpretation. Only a 10%-15% of the oceans have been mapped with spatial resolutions lower than 100m/px.

Higher resolution data is required to map the seafloor, not only for bathymetric studies but also for benthic mapping. The main difficulty that we face for mapping the oceans seabed is the rapid attenuation of electromagnetic waves in the underwater medium. Some remote sensing techniques used on land can still be used on shallow waters, but they become less accurate as the depth increases. While acoustic devices for remote sensing have become more precise and less expensive during the last decades, reducing the spatial data resolution is usually a matter of getting the sensing devices closer to the seabed. *In situ* data gathering systems, in contrast to remote sensing ones, are required to record high-resolution data and enable precise mapping of the seafloor.

In situ underwater data gathering has largely relied on scuba divers, towed systems and ROVs. However, last technological advances on AUVs are filling the gap between the use of the aforementioned systems

and the use of RS methods, improving the spatial and temporal resolutions of the recorded data [3]. AUVs are extensively used today to perform nested surveys because of their ability to follow preprogrammed inspection paths independently of a support vessel. In such nested surveys, for instance, acoustic RS from ships, AUVs and ROVs work in a complementary way for increasing details in specific target areas [4].

1.2.1 Change of paradigm: from inspection to exploration

Autonomous navigation is the basic nature of AUVs, being able to follow a mission path given several sensor measurements. In its classical conception, the robot is insensitive to the recorded data, that is stored for post-processing after recovery. Such behavior involves the execution of a control loop for path following; using the measurements from the navigation sensors, the AUV pose is estimated by a *localization* process and compared to the reference one in the *control* given a desired mission path, thus generating the thruster set points to minimize the errors. Figure 1.2 illustrates such autonomous navigation framework.

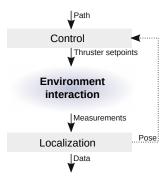


FIGURE 1.2: Autonomous navigation framework used for inspection applications: the control process commands the motor setpoints given a desired mission path and the estimated pose generated by the localization filters.

A typical autonomous navigation mission would require to follow a preprogrammed path, for instance a lawn mower pattern. This is a very useful method for inspection purposes, to be able to choose precisely where to sample and leave the robot executing autonomously the recording. However, this framework is not effective for dealing with large unknown regions and uncertain data.

In the case of dealing with an unknown environment, it can be difficult to preprogram a mission path; several decisions have to be made in terms of mission extension, resolution and location. Such decisions can lead, for instance, to over-record low informative areas and under-record hotspot regions. We solve such difficulties evolving from inspection to an exploration framework. In this context, exploration refers to the ability to retrieve and interpret data from an unknown environment that can be used to modify the initial mission plan during its execution.

In the case of having permanent and high bandwidth communication, a remote operator could act as the exploration agent, adapting the sampling paths according to the mission requirements. However, benthic data gathering operations with AUVs usually involve a low communication bandwidth, which is not sufficient to transmit the sampled data to a remote operator. Hence, the data transmission with a remote operator can only take place between robot dives.

Figure 1.3 illustrates the change of paradigm. When using an inspection framework for exploration of an unknown target area, the pipeline is usually based in the following steps: (i) an operator preprograms the sampling paths using the information extracted from prior existing data, (ii) the robot is deployed to

navigate autonomously along the programmed paths while recording data, (iii) the robot is surfaced or recovered to retrieve the recorded data, (iv) the recorded data is post-processed in order to plan further missions and iterate over the same processes.

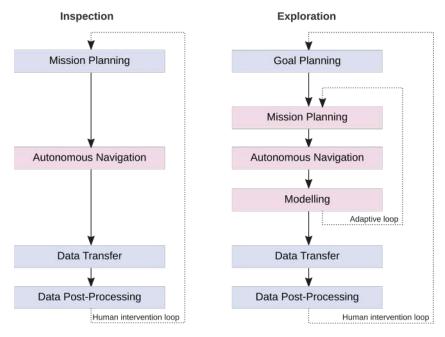


FIGURE 1.3: Data gathering process of an unknown environment following either an inspection or an exploration framework. In red the processes in which the robot is gathering data.

Following an inspection framework based uniquely on autonomous navigation to explore target regions has several limitations. This means to stop the robot from sampling, recover or surface it to download the recorded data, process such data, replan further missions and deploy it again. In contrast, the exploration framework includes two extra processes that enable the addition of an *adaptive loop* composed by two extra autonomous processes.

Notice that in the inspection framework the operator defines preprogrammed missions using the information extracted from data collected in prior dives. In contrast, in the exploration framework the operator would define a set the sampling goals (such as the target region, or physical processes of interest) and the mission planning would be automated and conditioned to such mission goals and to the online processed data. The main objective of such change is to increase (1) the time that the AUV can spend sampling and (2) the informativeness of the gathered data.

First, increasing the sampling time is a matter of taking the operator out of the sampling loop, or at least reducing its presence. Considering that tasks related to human intervention require stopping the sampling process, a higher degree of autonomy or adaptability is required to enlarge the time that the robot is able to keep in the sampling process. Second, increasing data informativeness is a question of adapting the sampling process to the knowledge about the environment. The addition of an adaptive loop will provide a faster response to drive the robot to high informative regions.

The change of paradigm involves the automation of the decision about where to sample; this is the automation of the exploration process.

1.2.2 Towards automating underwater exploration

AUVs have many challenges to overcome in terms of mission endurance [5] and increased autonomy [3]. Nevertheless, last improvements on edge computing are enabling the execution of adaptive sampling methods on board of AUVs and automate the exploration process. According to *Huvenne et al.* [4], the improved resolution of unmanned vehicles together with their rapid cost reduction is expected to increase their use on the future, being the coordination of multiple vehicles together with autonomous adaptive sampling for ocean exploration the most promising development of next future.

For the automation of the exploration process a robot must be able to assess the environment in terms of data informativeness, and to balance the exploration vs. exploitation trade off. This means that the exploring agent has to (a) integrate the capacity of perceiving and modelling the variable under study, and (b) leverage the exploration of the environment in order to improve the model, and the exploitation of the model to sample the most useful data. In any case, automating underwater exploration involves the integration of enhanced perception and decision-making capabilities.

This Thesis aims advance the boundaries of marine exploration by developing new methods for enhanced perception, decision-making and autonomous navigation to enable data-driven exploration. This will facilitate *in situ* data collection, reducing data uncertainty and increasing AUV mission performance and autonomy.

1.3 Context

The drive towards automating underwater exploration is the urgency on developing new methods to record high resolution data on seagrass ecosystems in order to ease the management of such environments. The introduction of new technologies based on AUVs is promising, and allow the substitution of tedious, high risk and repetitive tasks usually performed by divers.

1.3.1 Seagrass ecosystems

Seagrass ecosystems generate huge ecological value [6, 7, 8]. According to Olyarnik et al. [9], they provide key ecological benefits to coastal marine environments, including organic carbon production and export, nutrient cycling, sediment stabilization, enhanced biodiversity, and trophic transfers to adjacent habitats. Considering that changes in seagrass distribution directly affect the health of an ecosystem, seagrass meadows are also valued as biological sentinels: reduced seagrass density or spread points to major problems in ecosystems, generating losses in the ecological services provided. In this sense, they can be used as an environmental descriptor.

Posidonia oceanica is an endemic seagrass species to the Mediterranean Sea. According to Telesca et al. [10], P. oceanica has seen an estimated 35% aggregate reduction in extent over the last 50 years. Seagrass has a long evolutionary history. Nevertheless, its adaptation is now threatened by rapid environmental changes of anthropogenic origin. A central concern is its rapid regression, that creates a feedback loop that hampers restoration. In order to defend and encourage scientific research on P. oceanica meadows, they have been included amongst priority habitats in several European directives: the EU Habitat directive 92/43/CEE and, more recently, in the Marine Strategy Framework Directive 2008/56/EC. Seagrass meadows restoration is key to rebuild the marine life [11] and reducing CO_2 levels to meet the targets of the Paris Agreement of 2016 on climate change.

Knowledge about the environment is essential for seagrass management. Whereas the interest of scientific research into seagrass has heightened over recent decades, a consensus on quantifying the causes underlying P. oceanica regression has not yet emerged: some authors identify climate change as a major threat [12], whilst others focus on sewage spills or mechanical damage caused by anchoring [13], or trawling [14]. Nonetheless, there is general agreement that the main threats are anthropogenic [15, 16, 17, 18, 19]. The lack of definitive answers explaining the recession of P. oceanica is broadly linked to the lack of precise data, which are key for proper ecological management and conservation [20]. More specifically [21] highlight a lack of data on spatial distribution and a need to introduce modern data acquisition techniques.

The spatial distribution of seagrass meadows is usually described in terms of shoot density, upper/lower limits and sea bottom coverage [22, 20], with the most common sampling methods based on mapping, discrete quadrat sampling and transects [21]. Figures 1.4a and 1.4b show two raw images recorded from a 4m and 3m altitude, and Figure 1.5 represent a mosaic build from the images recorded with an AUV following a preprogrammed mission path.

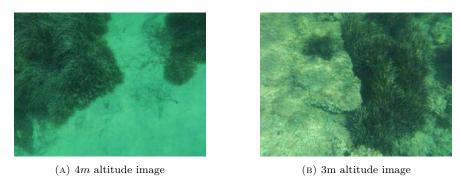


FIGURE 1.4: Images from two benthic environments covered by P. oceanica seagrass.



FIGURE 1.5: Image mosaic from a benthic environment covered by P. oceanica seagrass. Recorded with an AUV following a preprogrammed lawn mower pattern of 40x15m.

This work attempts to answer Abadie's call for the need to introduce modern techniques to generate data on the spatial distribution of P. oceanica [21]. It aims to fill the gap between academic research and end users by proposing a functional solution for spatial data-gathering, using cutting-edge technologies in lightweight marine autonomous vehicles and the latest developments in image processing and machine learning.

1.3.2 Local interest

The work presented in this Thesis has been supported by the Government of the Balearic Islands (CAIB) under Grant FPI/2031/2017, and developed in the Systems, Robotics & Vision research group of the University of the Balearic Islands. The objectives set for this Thesis were driven by the opportunity of working in the line of research that the group was approaching about benthic habitat mapping of P. oceanica meadows, and by the goals defined in the Science, Technology, Innovation and Entrepreneurship Plan (PCTIE) of the Balearic Islands 2013–2017 relative to Area of Marine Science and Technology.

At that moment, the research group started providing proof of the advantage of using automatic methods for seagrass image classification and habitat mapping [23, 24], and aroused the interest of the *Mediter-ranean Institute for Advanced Studies (IMEDEA)* to begin a series of collaborative works. The development of this Thesis has been simultaneous to the execution of the DETECPOS project 2018 - 2022 for the development of new technologies for the automatic and periodic assessment of changes in *P. oceanica* meadows due to anthropogenic causes.

The aim of improving the methods for automatic monitoring *P. oceanica* meadows has lead to the renovation of two marine vehicles, the AUV Turbot and the ASV Xiroi, which are equipped with stereo camera rigs and a complete sensor payload to allow autonomous navigation in challenging environments, maintaining a short recording distance from the seabed, and a high level behavior based in an enhanced perception system and decision-making capacity resulting from this Thesis.

1.4 Document overview

This document is structured in the following chapters:

- Chapter 2 provides a background on benthic data acquisition methods, describes and enumerates different remote sensing and *in situ* methods used for benthic habitat mapping, and reviews recent works related to data driven exploration.
- Chapter 3 describes a novel method for online benthic habitat modelling. This method processes the data recorded by an AUV and trains a probabilistic model used to generate habitat predictions, which are key to enable autonomous exploration.
- Chapter 4 describes a novel decision-time adaptive replanning framework. The presented framework integrates a variation of the Monte Carlo tree search algorithm to solve the information path planning problem.
- Chapter 5 describes a set of navigation methods implemented on a hovering AUV to enable sampling stereo images from the seabed with a high spatial resolution and low localization error.
- Chapter 6 describes a novel data-driven exploration framework. This framework integrates the methods described in the previous chapters for modelling, replanning and navigation in a unified framework developed for adaptive visual information gathering.
- Chapter 7 concludes the work by providing a quick summary and discussion of the results obtained during the development of this work, drawing the main conclusions, contributions and future lines of research.

1.5 Related publications

Journal Articles

- E. Guerrero, F. Bonin-Font, G. Oliver, Adaptive Visual Information Gathering for Autonomous Exploration of Underwater Environments, *IEEE Access*, vol. 9, pp. 136487, 2021, DOI 10.1109/ACCESS.2021.3117343.
- M. Bresciani, F. Ruscio, S. Tani, G. Peralta, A. Timperi, Eric Guerrero, F. Bonin-Font, A. Caiti and R. Costanzi, Path Planning for Underwater Information Gathering Based on Genetic Algorithms and Data Stochastic Models. *Journal of Marine Science and Engineering, MDPI*, vol. 9, no. 11, pp. 1183, 2021, DOI 10.3390/jmse9111183.
- E. Guerrero, F. Bonin-Font, M. Martín, Y. González, G. Oliver, Sparse Gaussian Process for Online Seagrass Semantic Mapping, Expert Systems with Applications, Elsevier, vol. 170, pp. 114478, 2021, DOI 10.1016/j.eswa.2020.114478.
- M. Martín, E. Guerrero, F. Bonin-Font, Y. González, Deep Semantic Segmentation in an AUV for Online Posidonia Oceanica Meadows Identification. *IEEE Access*, vol. 6, pp. 60956, 2018, DOI 10.1109/ACCESS.2018.2875412.
- F. Bonin-Font, J. Lalucat, G. Oliver, M. Massot, E. Guerrero, P. L. Negre, Evaluating the Impact of Sewage Discharges on the Marine Environment with a Lightweight AUV, Marine Pollution Bulletin, Elsevier, vol. 135, pp. 714, 2018, DOI 10.1016/j.marpolbul.2018.07.071.

Conference Proceedings and Workshops

- E. Guerrero, F. Bonin-Font, G. Oliver, Decision-time Adaptive Replanning (DAR) behavior for Benthic Environment Monitoring using AUVs, *IEEE ICRA (International Conference on Robotics and Automation)*, 1st Advanced Marine Robotics Tc Workshop: Active Perception, 2021.
- Y. González, F. Bonin-Font, E. Guerrero, A. Martorell, M. Martín, G. Oliver, Autonomous Marine Vehicles and CNN: Tech Tools for Posidonia Meadows Monitoring, IEEE Oceans, Sant Diego/Virtual, 2021.

Chapter 2

Related work

This chapter provides some relevant background to contextualize the methods proposed in this Thesis. First, Section 2.1 focus on the benthic data acquisition problem. Reviewing different type of methodologies; ranging from remote sensing methods such as satellite imaging to *in situ* alternatives such as image recording with ROVs or AUVs. Afterwards, Section 2.2 sets the focus on the *in situ* data generation with AUVs, introducing a taxonomy of relevant works regarding data-driven exploration.

2.1 Benthic data acquisition

The objective of this section is to provide a review of general methods used to acquire benthic data. In particular, we are interested on recording *spatial data* that can be used for benthic habitat mapping (BHM). The term spatial data refers to data that references a specific geographic location. Such data, for instance, could contain bathymetry, bionomy or geological information, and could be used to classify benthic environments or assess their condition. In general, benthic data acquisition methods are used to provide knowledge about certain physical, biological, chemical or geological processes, that may vary depending on the application.

Benthic data acquisition methods are usually classified in two general groups depending on the location of the sensing device: remote sensing (RS) and in situ (IS) methods. RS methods include the acquisition of optical data such as spaceborne and airborne imagery, as well as acoustic data from echo sounders. In contrast, IS methods typically include human inspection, video recording and sediment sampling. RS and IS methods are not mutually exclusive, but complementary. Whereas RS methods provide data with a large footprint that allow large area coverage, IS methods are able to generate high-resolution data. Sections 2.1.1 and 2.1.2 delve deeper in both methods.

2.1.1 Remote sensing

RS methods are very effective to generate long time series due to their repeatability. RS methods have been used for the assessment of coastal resources for long time [25], and they remain the most used method for retrieving long time series of data in shallow waters [26]. Such methods are a great choice for large scale mapping (over $1km^2$) of blue carbon ecosystems [27] and have been used extensively for seagrass monitoring [28, 29]. For example, in the latter case the lower bound and the deep bottom coverage of the seagrass meadow can be determined using acoustic backscattering methods, whilst the upper bound

and the shallow bottom coverage can be determined with airborne or spaceborne images [30]. Here we divide RS into optical and acoustic methods depending on the type of sensor installed.

Optical RS

Optical methods include the use of multispectral and hyperspectral cameras. The main difference between them lie in the number of bands that they provide and in the narrowness of each of the bands. While multispectral cameras provide from 3 to 10 bands, hyperspectral cameras provide from a hundred to more than a thousand bands. The main practical drawback of hyperspectral cameras at the moment is their cost and their image resolution.

Moreover, in the optical group of RS methods we consider passive methods such as spaceborne imaging and active methods such as aircraft and unmanned aerial vehicle (UAV) imaging methods.

In terms of spaceborne RS systems, the launch of *IKONOS* (1999-2015) enabled the acquisition of 0.82 meters resolution panchromatic¹ images and 3.28 meters resolution multispectral images (4 bands). And two years after *QuickBird-2* (2000-2015) started producing high resolution multispectral images with four bands; 0.61 meters resolution panchromatic images and 2.44 meters resolution multispectral images. This pushed forward the research on remote sensing methods for benthic mapping in shallow waters in general [31], and specifically for mapping seagrass [32, 33, 34].

More recently, other commercial spaceborne systems such as WorldView-2 (2009) and WorldView-3 (2014) are providing higher spatial resolutions and more spectral bands than prior RS systems. The data generated from both systems have been extensively used for land [35, 36] semantic analysis. Some recent studies have assessed their performance for marine and estuarine environments [37, 38, 39, 40]. Table 2.1 provide more details about the spaceborne RS systems mentioned above.





- (A) WorldView-3 satellite. Credit: geoimage.
- (B) PlanetScope satellite. Credit: Planet.

Figure 2.1: Example of optical spaceborne systems.

PlanetScope is the most up to date RS system launched for multispectral imaging. It is a new concept of constellation formed by 130 small-size satellites, called *Doves*, with a design life of 2 to 3 years. It provides multispectral images with 4 bands and a very high temporal frequency (almost daily observations) suited for monitoring fast dynamic events on land [41] and on marine environments [42, 43].

Freely available spaceborne datasets from Sentinel-2 (S2) have been also extensively used for seagrass mapping [44, 45, 46]. However, most public RS datasets resolution range from 10 to 60 meters depending on the spectral band and, unfortunately, they are still far away from commercial options.

In contrast with spaceborne, since airborne systems fly at lower altitudes, the spatial resolution of the images can be increased and the atmospheric attenuation of light reduced. Lathrop et al. [47] and

¹A panchromatic image is a single-band grayscale image with a high spatial resolution that contains the information from the visible R, G, and B bands. It yields a single integrated band containing no wavelength-specific information.

RS system	Bands	Spatia	l resolution [m]	Altitude	Launch	Decommission	Design life
		PC	MS	[km]			[year]
IKONOS	4	0.82	3.28	681	1999	2015	7
QuickBird	4	0.61	2.44	450 - 482	2001	2015	5
WorldView-2	8	0.46	1.84	770	2009	-	7.25
WorldView-3	16	0.31	1.24	617	2014	-	7.25
Sentinel-2	13	-	10,20 or 60	786	2015	-	7.25 - 12
PlanetScope	5	0.8	3.00	450 - 580	2016	-	-

Table 2.1: Spaceborne RS systems specifications. PC and MC refer to panchromatic and multispectral bands respectively.

Uhrin et al. [48] use multispectral images for seagrass mapping with a mean spatial resolution after orthorectification of 1.0m and 0.3m, respectively. At airborne altitudes hyperspectral camera images start providing data with a spatial resolution fine enough to map seagrass environments accurately. Clarke et al. [49] describe a study case where the authors map an area of $784km^2$ in coastal waters using airborne hyperspectral images, recorded with a spatial resolution of 2m. They validate the method performing a series of video transects, getting an overall mapping accuracy of 98% for seagrass cover and a 85% for genus classification in waters up to 10m depth.

The use of unmanned aerial systems (UAS) for RS has been increasing during the last years thanks to their increased autonomy and reduced costs. *Colomina et al.* [50] provides a complete review of the use of nano-micro-mini UAS for photogrammetry and RS purposes. The authors also provide a detailed description of the most relevant/used imaging sensors.

 $Taddia\ et\ al.[51]$ presents a study case in which the authors use a 5 band multispectral camera $MicaSense\ RedEdge-M$ mounted on a quad-rotor UAV to monitor seaweed growth, providing a spatial resolution of the recorded images around 8cm and 5cm for a recording altitude of 120m and 70m respectively. $Komarek\ et\ al.\ [52]$ describe a study for the classification of land cover and vegetation type. The authors use a fixed wing UAS equipped with three imaging sensors (RGB, 4 bands multispectral, and thermal cameras) providing 0.03-0.18m spatial resolution data for identification of individual species of plants in land. They obtain very accurate results, and conclude that increasing spectral resolution leads to a significantly better performance.

Moreover, Nahirnick et al. [53] provides research about the environmental conditions in which UAS should be used to obtain high confidence seagrass mapping. The authors conclude that the best conditions for recording are: (1) sun angles below 40° to avoid reflectance, (2) positive theoretical visibility with Secchi depths > 5m, (3) cloud cover conditions of < 10% or > 90% to reduce impact on the radiometric consistency, and (4) wind speeds less than 5km/h.

Considering that each of the above-mentioned systems have their own advantages, *Gray et al.* [38] describes the use of UAS RS data for training spaceborne RS data classification algorithms. The authors use UAS RS data with 3cm spatial resolution and three spectral band images as a groundtruth to train spaceborne RS data from *WorldView-3* and achieve a seagrass coverage classification accuracy of 93%, which is very high considering the exhaustive field groundtruth used for validation.

Acoustic RS

Acoustics have been largely used for benthic mapping [54, 55]. By retrieving bathymetry and backscattering it is possible to infer different benthic areas. In this section we focus on the description and uses

of the most common systems used for benthic habitat mapping; single beam echosounders (SBES), side scan sonars (SSS), and multibeam echosounders (MBES).

Bathymetry information is directly obtained by measuring the time of flight of the acoustic echos to return to the receiver. However, backscatter information intrinsic to the seafloor is more difficult to obtain. It is not only determined by the consistency and the roughness of the seabed but also by the range distance and the angle of incidence of the acoustic wave. The retrieved backscatter information also depends on the frequency of the acoustic wave used, low frequencies < 100kHz are typically used for substrate identification, whilst high frequencies > 400kHz are used for benthic habitat classification purposes.

Acoustic profiles obtained from SBES have been largely used for characterization of benthic environments [56, 57]. However, due to their narrow beam width they are not used for coverage mapping on their own, but complementary with other RS approaches such as spaceborne images [58, 33].

In contrast, backscatter data obtained from an SSS provide a very wide field of view (around 120°) that have been extensively used for coverage mapping of benthic habitats during the last few decades. For instance, SSS data has been used to determine sediment variations [59] and for seagrass monitoring [60, 61]. The intrinsic issue of SSS RS is that the backscatter data can not be localized relative to the acoustic beams. The SSS does not provide bathymetric data, it only allows to get a temporal response for each ping. In order to plot the acoustic response on a map, several works mount a SBES to measure the central altitude and project the temporal data assuming a roughly flat environment or a pre-existent bathymetry of the area.

The first approach for mapping seagrass meadows using MBES is described by *Komatsu et al.* [62]. The authors use a MBES operating frequency of 455kHz with 60 beams and an angular resolution of 1.5° for building a three-dimensional spatial distribution map of seagrass meadows.

The last decade has supposed a big leap for MBES development. According to Lucieer et al. [63] the improvements in the backscatter data quality measured using MBES come from (a) the precise co-registration of backscatter with the MBES bathymetry dataset, (b) the optimal signal-to-noise ratio imposed from bathymetry measurements, and (c) the increased resolution of the measurements. MBES is now, by far, the most used device for acoustic benthic habitat mapping. The co-registration of bathymetric and backscatter data enables more accurate benthic classification since the range distance and incidence angle of backscatter data is known [64].

Prampolini et al. [65] describes the use MBES for substrate classification surveying at depths up to 250m with an operating frequency of 70 - 100kHz and obtaining a maximum spatial resolution of 1m. Advances in MBES technology have made possible substrate classification on shallow waters up to 30m depth with a sub-meter resolutions, and operating frequency of 360kHz [66, 67]. Ierodiaconou et al. [68] and Montereale et al. [69] used a Kongsberg Maritime EM2040 MBES equipped with 800 beams with an angular resolution of 0.28° for mapping in very shallow waters up to 22 and 13m depth and achieving spatial resolutions of 25 and 5-20cm respectively. With such resolutions the authors where able to distinguish different types of biota, including seagrass, macroalgae and sponges using an operation frequency of 300 and 360kHz.

Furthermore, since the intensity response changes depending on the sampling frequency used, recent work have focused on the use of multiple frequency backscatter data for benthic mapping. Using backscatter data recorded with different operating frequencies has lead to improve seabed classification accuracy using data recorded during different surveys [70, 71] and data generated using multiple-frequency MBES to record backscatter data at different frequencies simultaneously [72, 73]. Costa et al. [73] suggests that

multiple-frequency backscatter maybe most useful for soft-bottom habitats, whereas single-frequency backscatter may be adequate for hard bottom applications.

2.1.2 In situ sampling

IS sampling methods are extensively used in tandem with RS methods. In fact, most of the RS based methods for BHM require the acquisition of IS data, either for validation or for training classification algorithms. Hence, whilst IS sampling provides poor coverage and might be labor intensive, the high resolution and feature richness of the resulting data makes a good fit to be used as groundtruth for RS BHM methods.

There are many IS sampling methods depending on the physical variable under study. The most common group of methods used for BHM are: (1) sediment sampling, (2) direct observation, and (3) underwater photography. Sediment sampling includes the use of hand samplers by divers and remotely operated vehicles (ROVs), as well as grab samplers such as the Van Veen sampler used from a boat by *Trzcinska et al.* [70]. *Tuit et al.* [74] provides an extensive review of current methods for marine sediment sampling.

Direct observation is the most traditional way to retrieve spatial data from benthic environments. Either from surface by boat or snorkeling in very shallow waters with good visibility or by scuba divers. Whereas samples obtained from surface can be geo-referenced using GPS positioning, geo-referencing scuba diver observations is more complicated. In order to be methodical, scuba divers usually sample following transects or using quadrants (see Figure 2.2 for a representation of typical sampling strategies). Although these methods can vary significantly depending on the research study, they set the basis to reference samples. The main idea behind the transect method consists on mounting a metric line and annotate the benthic type changes along the line, providing an approximate resolution of 10cm. The quadrant sampling consists on annotating the benthic type and conditions contained on a quadrat of maximum 50x50cm. Both transects and quadrants can be placed randomly or in specific locations. However, in order to track temporal changes it is recommended to take the measurements in the same locations, since the habitat patchiness could have a major effect on the measurements than the temporal changes [21].

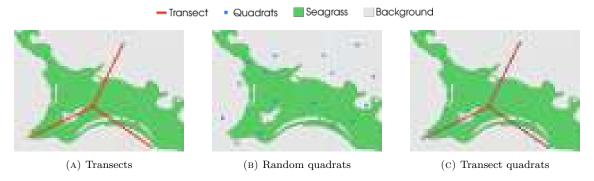


FIGURE 2.2: Different examples for visual inspection of benthic environments by scuba divers: (a) using three 100m long transects, (b) sixteen 30x30cm random samples or (c) sixteen 30x30cm samples localized along the transects.

Quadrant sampling is usually performed annotating the type of environment found in a quadrat placed in a random or specific location, content and coverage. Figure 2.2b shows a random sampling example, and Figure 2.2c shows a sampling example where the quadrants have been placed in specific locations along the transects. The advantage of guiding the sampling using transects is that if the central spike is localized using a GPS from surface and the orientations of the transects are measured using a compass, the sampled data can be geolocalized.

Underwater photography enables to store high quality seafloor images containing color and texture, including the use of monocular, stereo, and hyperspectral cameras. Such images have been recorded typically by scuba divers, by tethered systems such as drop-down cameras, towed cameras and ROVs, and more recently with autonomous underwater vehicles.

Scuba divers

In general, data acquisition by scuba divers has many drawbacks in terms of health, autonomy, control and localization. First, the risks associated with scuba diving. Second, the duration and extension of the sampling is limited by the scuba tanks. Third, some environments might be confusing and lead to undesired recording paths with bad coverage or uninformative data, Figure 2.3 represents the recording path followed by a scuba diver: some areas are unrecorded (lack of overlap between neighboring images in the orange circles) while others are over recorded. Finally, for the images to be geolocalized, scuba divers have to be equipped with acoustic transponders to enable localization [75], or follow the transect or quadrants strategies at known locations using a GPS on surface. If the divers are not geolocalized, the techniques used are prone to inaccuracies, e.g. drift during the immersion after taking the GPS positioning, or drift on the metric tape. Johnson et al. [76] provides a relevant study and Pizarro et al. [77] describes a relevant method to generate full image coverage for building small-scale photogrammetry-based seabed reconstructions.

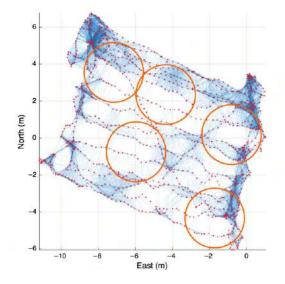


FIGURE 2.3: Lawn mower path followed for image recording with scuba divers. In red the camera keyframes, in blue the lines that join the overlapping images. The orange circles point out some unrecorded spots. Credit: *Pizarro et al.* [77]

Tethered imaging systems

Drop-down, towed and ROV image recording have a common characteristic of enabling a remotely operated in situ image recording, avoiding the human risks associated with scuba diving at expenses of a wired connection to a host ship. Whereas the sensing features of each specific device can vary depending on the target application, the main difference between them is the type of movement they are designed for. Drop-down systems are usually dived and left underactuated for a period of time, they are typically used to record images in particular spots, such as quadrant sampling. In contrast, towed systems are equipped with fins in order to record steady image transects when dragged from a host ship. Most drop-down and towed camera systems are designed to be lightweight and low cost, and rely on the host ship GPS

for data localization [56, 78, 58]. However, more complex systems are equipped with three-axis compass for measuring orientation—i.e. roll, pitch and heading—[79] and acoustic positioning systems [80], which improves the sampling accuracy and repeatability.

Figure 2.4 shows some examples of a particular ROV, towed camera and drop camera for particular applications.







(B) Towed camera system. Credit: Rende et al. [78]



(c) Drop camera system. Credit: National Oceanic and Atmospheric Administration, US.

FIGURE 2.4: Example of tethered image systems.

The difference with ROVs and drop or towed camera systems is that ROVs have several degrees of maneuverability, and can be actuated from an operator on board of the host ship, allowing a more accurate control of the sampling [70]. *Macreadie et al.* [81] provides an extensive review of ROV data collection practices, and *Robert et al.* [82] uses recorded images from a ROV to build photomosaics.

Autonomous underwater vehicles

The benthic data acquisition process can be automated by taking advantage of the latest developments in autonomous underwater vehicles (AUVs). According to Wynn et al. [5] AUVs are unmanned, untethered and self-propelled vehicles that can operate independently of the deploying ship. In contrast with ROVs, AUV have an enhanced mobility that enables surveying larger areas at higher speeds. However, the untethered condition means that AUVs have to be self-sufficient at least in terms of energy and navigation. The first is obvious, since there is not a wired, connection the AUV has to be equipped with a power source, thus energy efficiency becomes critical in order to increase mission endurance. The second is related to the limited communication bandwidth (if exists), and the lack of GPS.

Underwater environments are particularly challenging due to the rapid attenuation of electromagnetic waves. Acoustic signals are well suited for this medium, they are extensively used for wireless underwater communication. However, acoustic communications suffer from small bandwidth, low data rate, high latencies, multipath and variable speed that hinder remote operation. AUV localization usually relies on acoustic positioning systems [83] as the unique source of absolute positioning. It is important to notice

that for AUVs having an accurate localization is not important only for data geo-referencing but also for the vehicle control.

Whilst AUVs are considered IS sampling platforms, they are increasingly used as an intermediate solution between ship borne acoustic sampling and ROV inspection. Depending on the sensor payload, the variable under study and the resolution required for the recorded data, they can be configured to sample at different altitudes from the seabed. Thus, bridging the gap between RS and pre-existent IS methods.

BHM AUVs are usually equipped with stereo cameras or MBES for high-resolution data acquisition. For instance, Johnson et al. [76] uses the small torpedo-class Iver2~AUV equipped with a Prosilica stereo camera and achieve a very high resolution of 2mm/px, and Robert~et~al. [82] achieve a 0.5m/px resolution using a MBES mounted on the Autosub~6000~AUV to map vertical walls. Other relevant works include the use of the Sirius~AUV for stereo images recording coregistered with multibeam data [84, 85], and with hyperspectral imagery [86].

Figure 2.5 shows to examples of AUVs: the REMUS-600, that is a 3.6m long AUV able to reach a 600m depth, and the Autosub 6000, a 5.5m long AUV rated to navigate at a 6000m depth.





(A) REMUS-600 AUV. Credit: Sheri White, Woods Hole (B) Autosub 6000 AUV. Credit: National Oceanography Oceanographic Institution, US

Centre, UK

Figure 2.5

2.2 Data-driven exploration with AUVs

This section introduces a taxonomy of the most relevant works about data-driven exploration with AUVs regarding our particular application. The key idea behind data-driven systems is to use the data gathered during mission time to guide the robot navigation and increase its autonomy. Therefore, the objective is evolving from a framework in which the trajectory of the robot is preprogrammed and insensitive to the environment, to a framework in which the decision of the locations to visit depends on the data recorded so far.

Data-driven systems prove a great option in aiding robot navigation in order to maximize the information gain during data gathering operations. For instance, active SLAM approaches [87, 88], which are a remarkable application of data-driven systems, are used to determine when and where to look in order to find image loop closures to improve vehicle localization using saliency maps.

Another remarkable application of data-driven systems is for environmental monitoring. Some approaches propose the use of data-driven strategies to map environment variables, find trajectories to maximize information gain [89, 90], and reduce uncertainty around a threshold value [91, 92]. This thesis focuses on this application.

2.2.1 Information Gathering

Environmental monitoring and many other applications in robotics share the common objective of exploring an unknown environment for recording data to represent it. *Information Gathering* (IG) algorithms involve a particular type of data-driven methods that use an information metric generated from recorded data to guide the exploration. Such information metric used by IG algorithms represents the informativeness of the environment variable under study in particular locations. This metric is used to drive the data recording process towards the more informative spots whilst minimizing a cost, such as the number of measurements, the navigation distance or the mission time. IG algorithms have been used for different types of exploration, for instance; (i) goal-based, where the objective is traveling from an initial location to a goal location with a given cost budget [93, 94, 95], (ii) front-based, for traversing a given threshold area [91, 96], (iii) frontier-based, usually for indoor environment mapping [97, 98, 99, 100], (iv) multimodal, using different data sources [101, 102], (v) multirobot, using multiple robots [103, 104, 105], (vi) hotspot-based, to find environmental variable hotspots [92, 89], and (vii) coverage-based, for environmental variables dense estimation [106, 107], which is, in fact, the focus of this work.

The aforementioned methods are differentiated mainly by the method they use to perceive the physical process under study, and to decide the sampling strategy depending on the perceived information. The perception and the decision methods are key for the design of an IG strategy.

2.2.2 Perception

The perception method can be further divided in terms of the sensor used to measure the physical process under study, and the method used for modelling its spatial distribution. The type of sensor used heavily depends on the physical process under study. The most common in the literature is the use of punctual measurement like salinity sensors [106], cyanobacteria [107] or plankton [90] that are geolocalized with the estimated robot position. Furthermore, *Jadidi et al.* [100] use range sensors to model spatial occupation and *Popovic et al.* [108] use cameras for terrain monitoring.

In general, the most used method for spatial distribution modelling is using a Gaussian Process (GP) fed with the data collected by the robot [109]. GPs are a powerful nonparametric set of techniques that can handle a large variety of problems, and have the ability to learn spatial correlation with stochastic noisy measured data [110]. The key feature of GP for IG algorithms is their ability to handle both, data uncertainty and data incompleteness, effectively, which allows performing dense estimation of environmental variables for coverage-based exploration purposes. The problem of GPs is that they do not scale well, having a complexity of $\mathcal{O}(n^3)$. Some works overcome this issue by proposing the use of a sparse GP [106, 111] or local maps fusion using Bayesian Committee Machine (BCM) [100] to decrease complexity and to enable the online execution. Figure 2.6 provide a representation of a bathymetric estimation performed by $Ma\ et\ al.\ [106]$.

2.2.3 Decision-making

The decision-making method used can be further differentiated according to (a) the information function, (b) the *Informative Path Planning* (IPP) strategy, and (c) the adaptive strategy for replanning.

Information function

The information function is used to drive the robot navigation, it is related to the environmental variable under study, and it is usually computed considering the modelling uncertainty. *Hollinger et al.* [112]

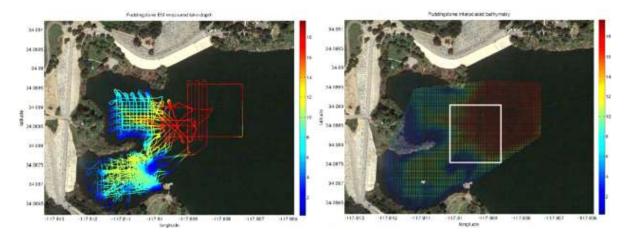


FIGURE 2.6: Resulting bathymetry estimation via GP from $Ma\ et\ al.$ [106]. The authors use bathymetric data from an acoustic depth sensor to train a GP and predict the environment bathymetry in a given target region. The color scale of the data points of the figure represent the lake depth in meters. Credit: $Ma\ et\ al.$ [106]

propose the use of an interpolated version of the GP model predicted variance, but most of the methods use either *Differential Entropy* (DE) or *Mutual Information* (MI) as information function. Using DE results in lower computation times and higher uncertainty reduction than using MI, when computed from a GP prediction in a stationary setup [113]. Despite the submodularity property of MI [114], the computational time of such information metric results prohibitive in many applications. Moreover, the use of an *Upper Confidence Bound* (UCB) information function has provided good results in applications that, in addition to exploration, require exploitation of the model [90].

IPP strategy

The IPP strategy has the objective of defining the most informative trajectories for IG, applying constraints such as the planning time or the travelling cost. The IPP strategies used for IG can be classified in two exclusive groups; myopic and non-myopic.

Myopic strategies are short-sighted, they only plan one step forward, and usually work following greedy heuristics [115, 116]. Thomson et al. [117] proposed a complete pipeline for autonomous planetary exploration, introducing the concept of science on the fly to adapt the robots' exploration to the collected instrument data, using a GP for environmental modelling and a greedy method for IPP.

In contrast, non-myopic strategies look several steps ahead, providing paths that may result better on the long run. Such strategies are usually graph-based, sampling-based or evolutionary-based. Whilst graph-based solutions are usually time expensive, they have been used for small state-space scenarios [93, 106] and in non-adaptive frameworks [102] where the mission path is processed offline. In contrast, sampling and evolutionary-based strategies, have been used for adaptive frameworks, and have yield high performance.

Most sampling-based methods [118, 89, 113, 119, 120] base their strategies in modified versions of standard sampling-based strategies [121, 122], by adding the information gain. In particular, *Hollinger et al.* [89] presented a rapidly-exploring IG tree (RIG-tree), that was the base of a further developed two-step planning process presented by *Viseras et al.* [113]. They proposed a solution based on *Rapidly-exploring Random Trees* (RRT) and RRT* [123], using a GP for environment modelling and a DE as information function, all to find a goal position providing a high information path under a budget constraint. Figure 2.7 provides further detail the RIG-tree framework designed by *Hollinger et al.* [89].

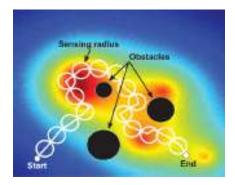


FIGURE 2.7: RIG-tree framework from *Hollinger et al.* [89]. The authors propose a series of rapidly exploring information gathering algorithms able to plan information paths in areas containing obstacles. The color map represents the map informativeness; the red areas contain more information than the blue areas. Credit: *Hollinger et al.* [89]

Evolutionary-based strategies solve the IPP problem by parametrization and optimization of a path. *Hitz* et al. [107] and *Popovic et al.* [108] propose the use of a GP to model the environmental variable and a CMA-ES algorithm to optimize the path to be followed by the robot, with a fixed length and predefined initial and ending locations.

Adaptive strategy

In terms of adaptive strategies for replanning the literature is very scarce. The usual is to provide a linear execution pipeline where the vehicle is stopped for mapping and planning [113, 98]. Nonetheless, some authors propose to perform the process of environmental modelling in parallel to the planning process in order to stop only for replanning [124], and some others avoid stopping by proposing to use as the origin of the new planning a location contained in the current mission path that the robot has not reached yet [107]. In any case, most relevant strategies, either sampling-based [113] or evolutionary-based [107] do not transfer planning knowledge between consecutive planning iterations.

Some of the key points that differentiate the framework that we propose in this Thesis from previous works are (1) the method used to generate spatial data samples from stereo images, (2) the representation of the benthic environment using a specific sparse GP configuration, and (3) the use of a computational efficient method based on reinforcement learning for IPP. The adaptive replanning strategy proposed in the next chapters will allow continuous navigation without stopping the robot neither for mapping nor planning, will transfer planning knowledge between consecutive replanning iterations, and will be computationally efficient in order to be deployed in a lightweight AUV.

Chapter 3

Semantic habitat modelling

This chapter focuses the perception part of the IG strategy. Providing a framework to enable the construction of a probabilistic model of the environment using the data provided by an AUV equipped with a stereo camera. The semantic perception of the environment will be the basis of further adaptive planning and exploration of the next chapters.

The objective of the proposed framework is to semantically model the spatial distribution of benthic environments. The challenge here is threefold. First, the semantic modelling has to be computation efficient and time bounded to enable online execution onboard of an AUV, we are assuming low bandwidth communications with an untethered AUV that impede remote computing. Second, the semantic model has to be probabilistic in order to capture data uncertainty, such uncertainty will be required for exploration purposes. Finally, the last challenge is that the semantic model has to integrate data from a stereo camera.

In order to handle that challenges, the proposed framework has been based on the use of an encoder-decoder convolutional neural network (CNN) to segment the images recorded by the stereo camera and a sparse Gaussian process (SGP) to model the spatial distribution uncertainty of the environment. The configuration of such CNN and SGP models is determined using field-test data and extensively validated during field experiments.

To the extent of the author's knowledge, no similar approaches aimed at improving visual data-gathering operations for seagrass have been published in the literature so far. The most relevant *in situ* optical mapping techniques for seagrass do not generate an online data flow to assist in strategic decisions during data-recording campaigns [75, 125, 23]. Where the goal of *Bonin et al.* [23] is to improve data post-processing, and not to ease data gathering operations. None of them provide a measure of the uncertainty bonded to the recorded data.

The proposed framework is based on two layers, one for sensing and another for spatial modelling, described in Sections 3.1 and 3.2 respectively. Section 3.3 provides a description of various experiments performed for model selection and Section 3.4 enumerates the results obtained. Finally, Section 3.5 outlines the main conclusions.

3.1 Sensing

The sensing layer captures stereo image pairs (left and right images) and uses a pre-trained deep neural network to produce an output flow of *segmented point clouds*, one per image pair, with a pre-defined frequency. The following processes are involved:

- Image Gathering: The acquisition of stereo images, including the setting of the navigation altitude, vehicle speed and maximum exposure time.
- Stereo Image Processing: This includes the rectification and decimation of the left and right images, as well as the computation of the disparity image between frames to estimate the image depth.
- Image Segmentation: This inputs rectified and decimated left images into the CNN and outputs a greyscale image that represents the probability of *P. oceanica* being present.
- Image Projection: This uses the disparity image to project a set of segmented image pixels into the 3D space (relative to the camera pose or keyframe). The resulting segmented point clouds are used for spatial modelling.

Sections 3.1.1 to 3.1.3 provide a more in-depth description of the image gathering, image segmentation and image projection processes.

3.1.1 Image gathering

Depending on the mission requirements and the lighting conditions, several navigation settings must be configured beforehand in order to obtain high-quality images.

In terms of mission requirements, a configuration for fast sparse modelling requires, for instance, high vehicle altitude to retrieve images with a large footprint and high speed; in contrast, high resolution mapping requires lower altitudes and lower speeds in order to achieve small pixel size and low blurring. Maximum vehicle speed and altitude play an important role and are determined as follows.

Vehicle altitude conditions image footprint $fp_{w,h}$ and pixel size $ps_{w,h}$ (both in metric units), where w and h express the width and height of the image. The image footprint is obtained using the camera field of view $FOV_{w,h}$ and the aforementioned vehicle altitude a, such that

$$fp_{w,h} = 2 \cdot a \cdot \tan\left(\frac{FOV_{w,h}}{2}\right).$$
 (3.1)

Pixel size $ps_{w,h}$ is obtained by using the image resolution $r_{w,h}$, given in pixels, for both height or width, such that

$$ps_{w,h} = \frac{fp_{w,h}}{r_{w,h}}. (3.2)$$

Both the footprint and pixel size are proportional to the altitude.

Low lighting conditions may require a higher maximum exposure time E_{max} . Since,

$$E_{max} = \frac{ps_{w,h} \cdot blur_{max}}{s},\tag{3.3}$$

decreasing the speed s or increasing the navigation altitude (pixel size) results in an increased exposure time margin. The maximum blurring admitted $blur_{max}$ is usually taken as 1 pixel in order to be insignificant.

The frame rate fps conditions the image overlap between consecutive images. For non-holonomic motion it is expressed as,

$$d = \frac{s}{fps} \tag{3.4}$$

$$d = \frac{s}{fps}$$

$$overlap_x = \max\left(0, \frac{fp_x - d}{fp_x}\right),$$
(3.4)

where d represents the Euclidean distance between consecutive keyframes, and fp_x the footprint in the advancing direction.

Our proposed method uses a lower frame rate and resolution for processing than for recording. Firstly, since the method does not look for correspondences between successive images, image overlap is not crucial. Secondly, although using higher resolution images would increase network performance in terms of accuracy, the image segmentation and the stereo disparity computation loads would also increase. Thus, the gathered images are downsampled to 480×360 pixels resolution, which is the input size of the CNN used, and the processing period is left as a resolution parameter: the *image segmentation period*.

Chapter 5 provides a further revision of the navigation settings, providing a set of expressions used to assess the impact of the navigation conditions, such as altitude, light and vehicle speeds to the recorded data accuracy.

3.1.2 Semantic segmentation

The semantic segmentation uses the encoder-decoder based CNN architecture VGG16-FCN8 [126] represented in Figure 3.1. It uses the model described in Martin et al. [127], pre-trained to discriminate the P. oceanica from the background in underwater images. Martin et al. compares the performance of VGG16-FCN8 with other state-of-the-art semantic segmentation architectures such as the U-Net [128] and the SegNet [129], resulting in a superior performance of the VGG16-FCN8 in terms of accuracy, precision, recall and fall out for the three tested datasets.

Use a pre-trained VGG16-FCN8

The architecture takes the images gathered by an AUV as input and outputs greyscale images where the grey level represents the probability of P. oceanica presence. This architecture is fully convolutional and can efficiently learn how to make dense predictions for per-pixel image tasks such as semantic segmentation. The architecture can be divided into two main blocks: the encoder and the decoder.

The purpose of the encoder is to extract features and spatial information from the original images by making use of a series of convolutional layers. This information is passed along the layers as feature maps, starting from low-level to deeper high-level information as more convolutional layers are applied. Furthermore, encoders implement a series of max pooling layers to reduce the size of feature maps, offering better computational performance as the number of parameters is concentrated.

The selected architecture makes use of the VGG16 encoder [130], removing the last classification layer and converting the last two fully connected layers into convolutional layers used to preserve the spatial information in the decoder and obtain an initial low-resolution segmentation. In turn, dropout layers are added to help reduce overfitting.

The purpose of the decoder is to take the low-resolution segmentation output of the encoder and upsample it to the input image size, obtaining high-resolution segmentation. The decoder makes use of transposed convolutional layers that apply inverse convolution on the input, up-sampling it. The decoder

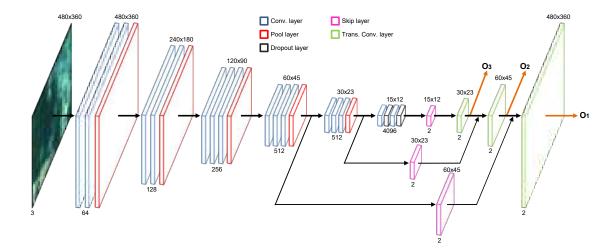


FIGURE 3.1: Scheme of the encoder-decoder based CNN architecture VGG16-FCN8 used for online image segmentation. Encoder: convolutional (blue), pooling (red) and dropout (black) layers. Decoder: score (purple) and transposed convolutional (green) layers from the decoder. The numbers under and above the layers indicate the number of feature maps and their size, respectively. The orange arrows represent the featured output points O_1, O_2 and O_3 .

also contains skip layers [131] used to combine high-level features from the encoder with the coarse information of the transposed convolutional layers. Lastly, an activation layer is applied in order to obtain the final semantic segmentation.

The architecture makes use of the FCN8 decoder structure [126] that contains three of the aforementioned transposed convolutional layers and three interleaved skip layers. The network outputs a resolution grey scale image depicting the P. oceanica presence probability.

Optimize the network performance

The CNN has been integrated in the ROS-based node using the open source library TensorFlow [132] to enable online segmentation onboard of an AUV running a software architecture based on ROS. In order to save resources, the network is fed with raw images at a predefined rate. Furthermore, the images fed to the CNN are decimated to the input resolution of the CNN, which is 480×360 pixels.

Moreover, in the same direction of reducing computing loads, three different models have been looked at from the original decoder.

- M1 is obtained by applying the softmax layer at the end of the architecture, after the third transposed convolution—i.e. 480×360 pixels resolution output, O1 in Figure 3.1—, offering a full-size prediction containing high-level detailed information.
- M2 is obtained by changing the softmax layer position after the second transposed convolution—i.e. 60 × 45 resolution output, O2 in Figure 3.1—, where two skip layers have already incorporated some high-level information.
- M3 is obtained by applying a softmax layer right after the first transposed convolutional layer, obtaining a low-resolution prediction output—i.e. 30 × 23 pixels resolution output, O3 in Figure 3.1—containing solely coarse information.

The main idea is to set the output of the network before of some of the last transponded convolutional layers, producing a lower resolution output. These models have been extracted from the original best

performing model presented in [127], where it was trained and validated to detect P. oceanica areas. Sections 3.3.4 and 3.4.1 provide a specific validation of the presented models in order to select one best performing model to be used in the modelling framework.

3.1.3 Image projection

The process makes use of stereo image disparity in order to project a *subset* of pixels from the segmented image into 3D coordinates. The resolution of this *subset* is set as a configuration parameter referred to as *segmented point cloud resolution*. Every projected subset, the so-called *segmented point cloud*, is obtained by resizing the disparity image and the segmented image linearly to the given segmented point cloud resolution. Then, by using the disparity image and the intrinsic camera parameters, we are able to perform the transformation from image coordinates (u,v) to 3D positions (x,y,z) referenced to the camera keyframe. Figure 3.2 illustrates the projection process and Figure 3.3 represents eight examples of resulting point clouds.

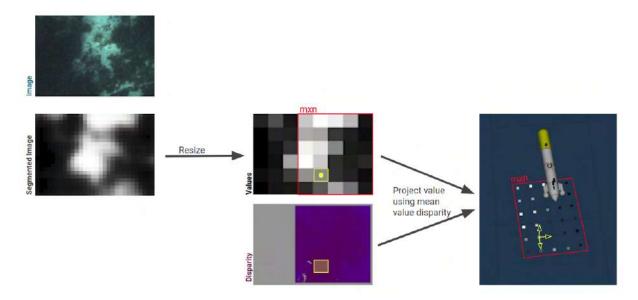


FIGURE 3.2: Online image processing to generate semantic point cloud data from stereo images recorded by an AUV.

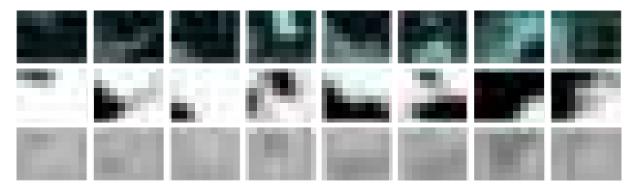


FIGURE 3.3: Example of 8 images gathered during field-tests together with the corresponding segmented images and semantic point clouds.

The final step to transform such semantic point clouds to global coordinates is to obtain the pose (referenced to the global reference frame) of the left camera. The localization of the AUV is used to compute such pose. As successive point clouds are processed they are stored for spatial modelling.

3.2 Spatial modelling

The spatial modelling layer is executed under query. It stores the segmented point clouds provided by the image processing module and uses the estimated robot trajectory to produce a series of map representations of a given target region: (i) the raw data density map represents the density of the gathered data; (ii) the grid map represents the samples used for training the GP; and (iii) the coverage and (iv) uncertainty maps represent the probabilistic spatial distribution of P. oceanica. The spatial modelling layer is composed by the following processes:

- Transformation: This transforms the segmented point clouds to global coordinates using the estimated robot trajectory. Such transformed data, raw data hereon in, comprise data points described by their location (latitude, longitude, depth) and a label value. This label represents the probability of P. oceanica being present in a determined location.
- Sampling: This performs a grid sampling of the raw data contained in the target region. The resulting samples are referred to as grid samples.
- Training: This optimizes a GP using the grid samples as training samples to model the spatial distribution of *P. oceanica* in the target region, generating a GP model.
- Prediction: Using the trained GP model, this process provides estimations of P. oceanica distribution in the target region; a coverage map represents its mean value and an uncertainty map provides the associated variance.

The estimated robot trajectory used in the *Transformation* process can be obtained by using an extended Kalman filtering approach [133]. However, it can also be obtained using an extended Kalman smoothing approach [134], reducing the position error between successive image keyframes. Smoothing the estimated trajectory is useful, for instance, when using acoustic beacon positioning systems. Such systems usually provide low frequency measurements, which produce bounces in the estimated trajectory.

The grid filter used by the Sampling process uses the median cell value. By using a grid with a given grid sampling resolution, each grid cell retains the data point with the median label value, considering all the raw data contained in a threshold distance from each cell center. The distance threshold is set to the same value as the grid sampling resolution. Only the label value is considered for the median calculation and the grid cells that do not contain data are left empty. In this process, we also compute a raw data density map, which is build by counting the number of raw data samples used to obtain the median sample value in each cell.

Figure 3.4 shows a representation of the resulting grid samples projected in 3D, as well as the AUV model and the EKF trajectory.

3.2.1 Training and prediction

The training stage has the objective of training an SGP using the grid filtered semantic point clouds. The outcome of this process is an SGP model able to compute predictions about the type of habitat in a particular location. Such predictions are composed of a mean and a variance value. The mean prediction value is contained in the interval [0,1], the same interval as the labels of the data points used for training.

The next sections provide more detail regarding the spatial modelling using GPs and the implementation.

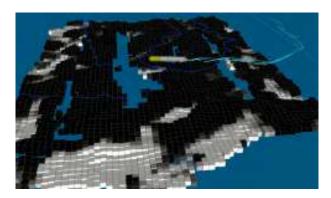


FIGURE 3.4: Representation of the *grid samples* projected in the 3D space (cube markers), the AUV model and the EKF trajectory. White cubes represent grid samples with low probability of having *P. oceanica*, dark cubes represent higher probabilities.

Background on Gaussian Process

Let us introduce the basis of the Gaussian processes used for model training. A GP is defined as a collection of random variables which have a joint Gaussian distribution [135]. The random variables represent the value of the non-observable function value $f(\mathbf{x})$ at location \mathbf{x} . Such function is specified by its mean function $m(\mathbf{x})$ and covariance function $k(\mathbf{x}, \mathbf{x}')$ values such that,

$$f(\mathbf{x}) \sim \mathcal{GP}\Big(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\Big)$$
 (3.6)

Assuming a linear regression model, $f(\mathbf{x}) = \phi(\mathbf{x})^{\top} \mathbf{w}$, where $\phi(\mathbf{x})$ represents a basis function vector, or kernel, and the weights \mathbf{w} follow a zero mean normal distribution $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}_{\mathbf{p}})$. Since $\mathbb{E}[\mathbf{w}] = 0$, $f(\mathbf{x})$ results in a zero mean:

$$m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})] = \phi(\mathbf{x})^{\top} \mathbb{E}[\mathbf{w}] = 0$$
 (3.7)

In addition, the covariance of $f(\mathbf{x})$ is derived from using the zero mean weights assumption such that:

$$k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x})) (f(\mathbf{x}') - m(\mathbf{x}'))]$$

$$= \mathbb{E}[f(\mathbf{x}) f(\mathbf{x}')]$$

$$= \phi(\mathbf{x})^{\top} \mathbb{E}[\mathbf{w}\mathbf{w}^{\top}] \phi(\mathbf{x}')$$

$$= \phi(\mathbf{x})^{\top} \mathbf{\Sigma}_{\mathbf{p}} \phi(\mathbf{x}')$$
(3.8)

It results in that the GP covariance directly depends on the basis function vector, the query locations and the variance in the weights.

Furthermore, since there is no direct access to the function values themselves—i.e. $f(\mathbf{x})$ —, and only noisy observations, we assume additive white noise ϵ with variance σ_n^2 on a latent variable modelled as $y = f(\mathbf{x}) + \epsilon$. Then according to Seeger et al. [135] the joint distribution of the observations \mathbf{y} and the latent values \mathbf{f}_* according to the prior is defined as,

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}^* \end{bmatrix} \sim \mathcal{N} \left(\mathbf{0}, \begin{bmatrix} K(X,X) + \sigma_n^2 I & K(X,X_*) \\ K(X_*,X) & K(X_*,X_*) \end{bmatrix} \right), \tag{3.9}$$

being $X = \{\mathbf{x_i} \mid i = 1,...,n\}$ the train locations, $X_* = \{\mathbf{x_i^*} \mid i = 1,...,n_*\}$ the test locations and $K(X,X_*)$ the matrix of covariances evaluated at the referenced locations.

Implementation details

The developed method for spatial modelling makes use of the open source library *GPflow* [136] for GP training and prediction. This library provides several models, kernels, and likelihood functions to build GPs.

Kernel functions

GP predictors are a probabilistic kernel machine. The learning process is based on the optimization of the parameters for the covariance kernel K(X,X), which characterize the spatial correlation of the function values.

These covariance kernels are usually based on smooth functions. This work focuses on using squared exponential ϕ_{SE} and Matérn $\phi_{M_{\nu}}$ basis functions. These functions have shown to represent the latent function spatial distribution with a good precision and using a minimum number of function parameters. They are described as:

$$\phi_{SE}(r) = exp\left(-\frac{r^2}{2l^2}\right) \tag{3.10}$$

$$\phi_{M_{\nu}}(r) = \frac{2^{1-\nu}}{l} \left(\sqrt{2\nu} \frac{r}{l}\right)^{\nu} K_{\nu} \left(\sqrt{2\nu} \frac{r}{l}\right), \tag{3.11}$$

where r represents a kernel distance metric and l the kernel length scale. The Matérn functions also have the scaling factor ν , which is typically set to $^{1}/_{2}$, $^{3}/_{2}$, $^{5}/_{2}$, and K_{ν} , a modified Bessel function value for a particular ν .

Models

This work proposes the use of SGP models. This kind of model approximates the training data by using a small group of *inducing points*. These have been shown to provide good approximations of the studied variable whilst improving the computational performance when using large datasets.

Three SGP models are looked on this work:

- Sparse Variational GP Regression (SGPR) [137].
- Sparse Variational GP using MCMC (SGPMC) [138].
- Sparse Variational GP (SVGP) [139].

The SGPR model introduces a variational formulation that jointly infers the inducing inputs and kernel hyperparameters. Whereas the SGPMC and SVGP models allow for non-Gaussian approximations over the function values, using non-Gaussian likelihood functions.

Likelihood functions

The likelihood function should represent the semantic label distribution of the data. Taking into account that (a) the semantic data used for training the SGP is contained in the interval [0,1], and that (b) the pixel labels are biased towards the extreme values (the CNN is confident about the predictions), we propose the use of a beta function to model the likelihood distribution of the latent function for models that allow non-Gaussian approximations (SVGP and SGPMC). The beta function is defined by a mean

sample value, given by the mean label value of the training set, and a scaling factor. We set the scaling factor to s = 1 in order to give higher probabilities to the extreme values [0,1].

3.3 Experiments

A total of three field-tests (FT1, FT2 and FT3) were performed using either the AUV Turbot or the ASV Xiroi, near the Bay of Palma (Balearic Islands), providing data from partially covered *P. oceanica* environments with different meadow spatial distributions.

This section introduces the robot configurations used for these experiments, the field-tests performed for data collection and the criteria followed for model selection and validation of the method.

3.3.1 Robot configurations

The software architecture of both robots is based on ROS [140]. Table 3.1 briefly describes the main specifications of the robots in terms of navigation payload, stereo camera and computer. All the processes related to image processing and environment modelling have been executed on CPU.

Table 3.1: Specifications of	the robots used for	r field-testing th	he semantic habitat	mod-
	elling.			

	AUV Turbot	ASV Xiroi
Navigation Payload	DVL, Pressure, IMU, Compass, GPS, USBL	IMU, Compass, GPS RTK
Stereo Camera	Point Grey CM3-U3-31S4 (2048x1536pixels)	Point Grey BB2-08S2 (1032x776pixels)
Computer	Intel i7 2.5 GHz, 4 cores, 8 GB of RAM and Ubuntu 16.04	Intel i3 2.33 GHz, 4 cores, 8 GB of RAM and Ubuntu 16.04

The AUV Turbot, Figure 3.5a, is a torpedo-shaped AUV based on a Sparus II [141] unit with three degrees of maneuverability: surge, heave and heading. The navigation payload consists of (a) a Doppler Velocity Log (DVL) to obtain linear and angular speeds and altitude; (b) a pressure sensor to obtain high-frequency depth measurements; (c) an Inertial Measurement Unit (IMU) to measure accelerations and angular speeds; (d) a compass for heading; (e) a GPS to be geo-referenced during surface navigation, and (f) an acoustic modem used in combination with an Ultra Short Baseline (USBL) for absolute localization. An Extended Kalman Filter (EKF) [133] with a posterior smoothing stage [134] estimates the robot's trajectory by fusing multiple state estimation from the sensor equipment. The downward facing stereo camera has an FOV in salt water of [0.54,0.40]rad. Moreover, in order to achieve fast data transfer, the AUV is tethered to a buoy equipped with an Ubiquiti Bullet M2-HP antenna.

The ASV Xiroi [142], Figure 3.5b, is an ASV based on a catamaran shape with two degrees of maneuverability: surge and heading. In this instance, it is important to mention that it uses an Emlid Reach Real Time Kinematic (RTK) GPS which, according to producer specifications, provides position measurements with an absolute error of less than 10cm. The ASV uses an EKF [133] filter plus smoothing to fuse the IMU, compass and GPS measurements for trajectory estimation. Moreover, the ASV is equipped with a wide angle stereo camera that has an FOV in salty water of [0.92,0.74]rad. Communication with the control station (CS) is established using a long-range Ubiquiti Bullet BM2HP antenna installed on board.





(A) AUV Turbot

(B) ASV Xiroi

FIGURE 3.5: Robots used for the semantic modelling experiments.

The vehicles were only used for data acquisition, and the CS computer was used for image segmentation and mapping. The CS computer is equipped with an Intel i7 CPU @ 3.40 GHz, 32 GB of RAM and Ubuntu 16.04. All the data gathered by the robots from the three field-tests were stored for model selection and validation.

3.3.2 Field-tests

Table 3.2 briefly describes the main specifications of the three field-tests performed. FT1 was conceived to be a best-case scenario in terms of localization and image segmentation. We used the Xiroi ASV to record highly precise localized data of a seagrass meadow with very sharp edges. Figure 3.6a shows an aerial image of the target region; a sharp meadow bound can be clearly distinguished. Figure 3.6b shows the raw data density map obtained from the surveyed area. This map represents the locations from which data have been gathered. Very large areas from the target region were left unrecorded in order to see how the GP estimation performed.

	FT1	FT2	FT3
Robot	ASV Xiroi	AUV Turbot	AUV Turbot
Location	39.5253N 2.5474E	39.5244N 2.5505E	39.5244N 2.5504E
Target region	$32 \times 18m$	$25 \times 15m$	$15 \times 10m$
Altitude	2.8m	3.0m	3.0m
Speed	1.0m/s	0.25m/s	0.25m/s
Seagrass meadow characteristics	Sharp edges	Many small patches with fuzzy edges	Few big patches with fuzzy edges

Table 3.2: Field-test specifications.

Following the reasoning of Section 3.1.1, the maximum exposure time to avoid blurring was found to be 2.72ms, sufficient for the lighting conditions at such depth. Figure 3.7a shows an example of the kind of images recorded: the light conditions are good, and the meadow has sharp edges.

In the case of FT2 and FT3, in order to use the AUV Turbot with precise USBL positioning measurements, the target region had to be deeper than 6m. However, such a target region also had to be as shallow as possible to be able to partially distinguish seagrass meadow bounds from aerial images. Thus, we surveyed two regions approximately 6.5m in depth following a lawn mower survey pattern.

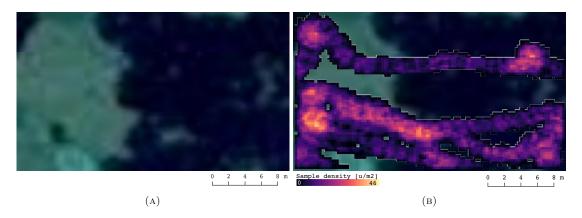


Figure 3.6: Cartographies from FT1 region: (a) $Google\ Maps$ cartography, superimposed in (b) with $raw\ data$ density map

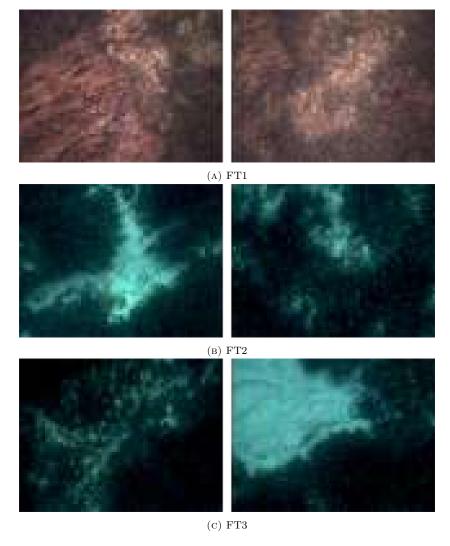


Figure 3.7: Sample images from FT1, FT2 and FT3. Seabed covered by sand, rocks and $P.\ oceanica.$

From the recorded images (Figure 3.7), the seabed in the target region in FT2 seems to have many small patches of background typologies (sand, rock and dead matte). By contrast, the target region in FT3 had fewer and bigger background areas. Figure 3.8a shows the aerial image of both target regions along with the inferred hand-labelled meadow bounds. This illustration also highlights small patches with fuzzy edges in the case of FT2 and fewer and bigger patches in FT3.

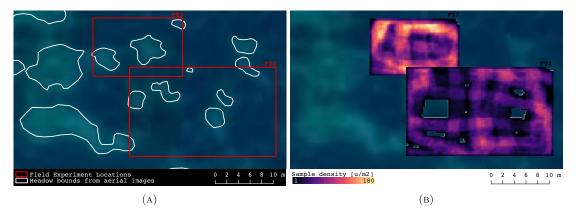


FIGURE 3.8: Cartographies from FT2 and FT3 regions: (a) Google Maps cartography, superimposed in (b) with raw data density map. The meadow bounds inferred from the aerial image are also represented in white.

Moreover, Figure 3.8b represents the raw data density maps. The mission for FT2 was programmed to have a small image overlap in order to leave several small regions unrecorded. By contrast, the mission for FT3 was programmed to provide more overlap between the transects and produce a high density of raw samples.

3.3.3 Resolution parameters

The processes described require several resolution parameters to be set beforehand: (a) the *image seg-mentation period*, (b) the *segmented point cloud resolution*, (c) the *grid samples resolution* and, (d) the *inducing points density*. They depend on the aims of the data-gathering campaign. Table 3.3 details three alternative configurations to test in this work; low, medium and high resolution (LR, MR and HR).

Image Processing	$\mathbf{L}\mathbf{R}$	\mathbf{MR}	$\mathbf{H}\mathbf{R}$
Image Segmentation period [s]	2.5	2.5	1.5
Segmented Point Cloud	[4 9]	[12,9]	[19.0]
Resolution $[p_u, p_v]$	[4,3]	[12,9]	[12,9]

Table 3.3: Configurations used for low, medium and high resolution mapping.

Mapping	LR	MR	$\mathbf{H}\mathbf{R}$
Grid Sampling Resolution [m]	0.5	0.3	0.25
Inducing Points Density [units/m]	0.1	0.3	0.4

3.3.4 Model selection criteria

CNN model selection

The three encoder-decoder CNN models described in Section 3.1.2 (M1, M2 and M3) have been evaluated in terms of class separability and execution time, using the original image test set proposed by *Martin*

et al. [127]. The image set consists of 117 images representing different *P. oceanica* meadow conditions such as meadow density, rhizome proportion, matte compactness and coloration, and different recording conditions such as water illumination, depth and turbidity.

The area under the curve (AUC) of the receiver operating characteristic (ROC) curve is proposed as a measure of class separability to evaluate the performance of each model. The ROC curve is computed using the number of *True Positives*, *False Positives*, *True Negatives* and *False Negatives* obtained when comparing the prediction output of the CNN with the hand-labelled groundtruth (GT) image pixel by pixel; the predicted image is binarized by using several thresholds contained in the pixel value interval [0,1]. The AUC of the dataset is computed as the mean AUC obtained from the ROC curve for each image.

In addition, the average time that it takes a model to segment an image has also been used to evaluate the performance of each model. The execution time of the dataset is computed as the mean execution time for each image. The best model configuration is selected as the one offering the best trade-off between both metrics: AUC and execution time.

SGP model and kernel selection

The SGP models and kernel functions described in Section 3.2.1 are evaluated in terms of class separability, uncertainty representation and execution time. A total of twelve SGP configurations (three models and four kernel functions) have been evaluated experimentally using three field-tests, using the SGPR, SGPMC and SVGP models and the ϕ_{SE} , $\phi_{M_{12}}$, $\phi_{M_{32}}$, $\phi_{M_{52}}$ kernel functions. Furthermore, each configuration has been tested using the three resolution configurations in Table 3.3 of Section 3.3.3.

The GT used to obtain the evaluation metrics comprises a small set of raw data (transformation process output) which, instead of being used for the sampling process to produce the grid samples used to train the GP, is pulled out of the pipeline and kept for testing. We use a k-fold cross-validation (CV) scheme and, for every test, the input dataset (raw data) is divided into k groups of samples uniformly distributed along the dataset. We perform k iterations per test, using one group for testing and (k-1) for sampling and training. In this case, a value of k=10 has been considered sufficiently high in order to lower the impact of reducing the training set. In order to obtain the evaluation metrics, we compare the prediction of each trained model with the set of raw data kept as GT. The resulting metrics for each test are obtained as the mean metric values of the k=10 CV iterations.

Class separability of the predicted samples is evaluated using the AUC of the ROC. The higher the AUC, the more separable the predicted classes. The ROC is obtained comparing the prediction output of the GP, sample by sample, with the GT set of raw data kept for testing. The SGP is queried for predictions at the same locations where GT data has been kept, and such predictions are binarized using several thresholds contained in the label value interval [0,1].

Uncertainty representation is studied as the correlation between regression error and resulting SGP variance, with the regression error being the sum of the absolute differences between GT and predicted values. In order to be useful, the resulting variance of the SGP has to be higher where the error in the prediction given by the trained model is higher. Such correlation has been captured using the absolute value of the Spearman correlation index (SP). An SP value near zero means no correlation between regression error and SGP variance, whilst a value of one implies an exact monotonic relationship; the higher the SP the better the uncertainty representation. Additionally, the execution time is measured as the time that it takes a training process to be executed.

3.3.5 Validation criteria

This section provides a detailed description of the criteria followed to validate the methodology proposed in this chapter. The validation is based on two steps: one to validate all the processes until the raw data is obtained, and one to validate spatial modelling. Both validation steps use data from the three field-tests described in Section 3.3.2, and the three resolution configurations proposed in Table 3.3 of Section 3.3.3.

In terms of raw data validation, we verify the quality of the segmented point clouds; in terms of spatial modelling, we evaluate the GP regression performance and how well the GP is learning from the recorded data.

Raw data validation

The first step in the validation process is planned to provide the source error of the raw data used by the spatial modelling. Such data is obtained through a sequential process of image gathering, segmentation, projection and transformation, and have two sources of error: one associated with misclassification and one with localization noise. Whilst the former can be quantified using pixel discrimination error metrics, the latter is more complex to assess.

Raw data misclassification is evaluated as follows: for each stereo pair of a data set (1) compute the segmented point cloud, (2) get a GT image by hand-labelling the rectified left image, (3) project the GT image in order to get a GT point cloud (using the same projection resolution as for the segmented point cloud), and (4) calculate the AUC of the segmented point cloud using the GT point cloud. The resulting AUC of the dataset is the mean AUC obtained from all the images contained in this dataset.

The localization error of the raw data can be associated with the image pixels projection to 3D and the robot pose estimation. The former is related to the disparity image computation and the intrinsic camera parameters; this validation assumes a precise stereo camera calibration and leaves this source of localization error out of the discussion. In contrast, we deem that the robot pose estimation may be prone to produce relevant localization errors on the absolute camera frame position and orientation estimations.

Pose estimation errors depend on the sensor equipment installed on the robot, as well as on the localization filter used. In the case of the robots used in the tests, as described in Section 3.3.1, we assume a position error of < 0.1m for the ASV and < 1m for the AUV, given the RTK GPS of the former and the multisensor payload of the latter [143]. In any event, the use of an EKS strategy is expected to produce improved cohesion for the mapping process since it tightens successive absolute position corrections together.

Whereas it has been impossible for the author to provide a map GT to validate the gathered data, the localization consistency of the raw data has been verified by using a set of geo-referenced aerial images obtained from *Google Maps*. Although such aerial images do not provide enough resolution to distinguish which areas are covered with *P. oceanica* in order to act as a GT, they have been used to check morphological coherence between both data sources. Therefore, these aerial images have been used to compare the location of the most evident seagrass meadow bounds with the gathered data.

GP prediction validation

The resulting maps from the GP regression have been validated using the AUC of the ROC to measure class separability, the Spearman (SP) correlation index to assess uncertainty estimation, and the execution time.

In order to obtain statistical metrics, we propose the use of the same strategy used for model selection; the same GT and CV analysis. However, in this instance we only test the selected GP configuration and perform k = 40 CV iterations in order to decrease the impact of keeping part of the raw data for testing. We analyze the performance of the selected model using the recorded raw data from each individual field-test.

3.4 Results

This section shows the results obtained after running the model selection and validation tests in line with the criteria proposed in sections 3.3.4 and 3.3.5. Section 3.4.1 reviews the model selection of the CNN and Section 3.4.1 reviews the model and kernel selection of the SGP.

3.4.1 Model selection

CNN model selection

Table 3.4 shows the results obtained for the three image segmentation models proposed in Section 3.1.2 and Figure 3.9 show some examples of segmented images using the different models.

Table 3.4: Results obtained to select the most suitable CNN model selection.

CN	$CNN\ Model$		ics
Name	Resolution	Time [s]	AUC
M1	480x360	0.133	0.994
M2	60x46	0.116	0.992
M3	30x23	0.115	0.963

The three models have AUC values over 0.96 which, following the criteria established by *Powers et al.* [144], represent excellent classifiers. Whereas M2 has an AUC that is 0.2% smaller than M1, both models have AUC values that are 3% higher than those of M3. The execution time drops a 13.5% and a 12.8% for models M3 and M2 respectively in comparison to M1.

Moreover, considering that the image recording altitude was 3.5m, and considering the Equation 5.7 provided in Section 5.1, the pixel size for the three output images is 0.8cm, 6.5cm and 13.1cm, respectively for M1, M2 and M3. Choosing a CNN model will depend on the required segmentation frequency and computing capacity of the AUV.

SGP model and kernel selection

Table 3.5 details the results obtained for the different tested SGP configurations.

The first important result is that the SGPR model configurations result in a negative SP index. This negative correlation indicates a low uncertainty representation: the regression error is not monotonically correlated with the resulting SGP variance. Thus, this GP model will be discarded from the discussion.

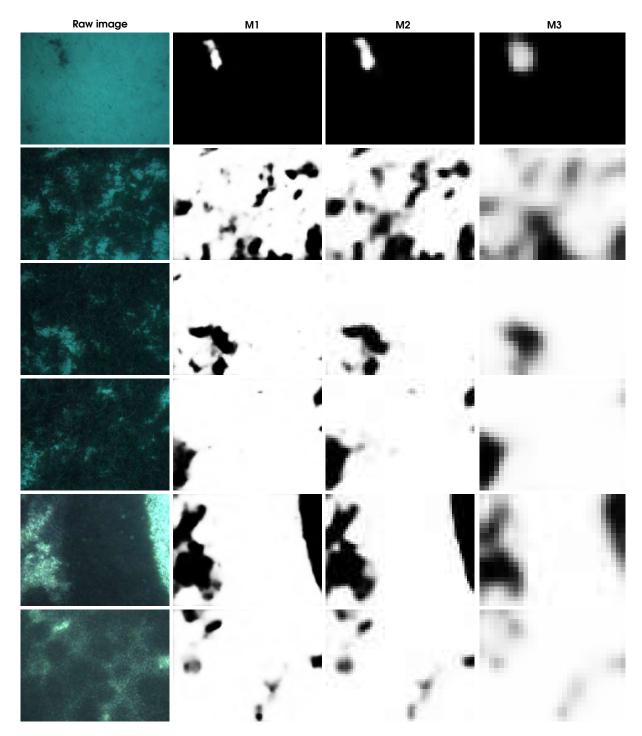


FIGURE 3.9: Resulting image segmentation for the three segmentation models considered. From left to right: the raw image used as input for the models (1920x1440), and the segmented output images from the models, M1 (480x360), M2 (60x45) and M3 (430x23).

SGP Con	figuration		Metrics	
\mathcal{M}	\mathcal{K}	Time	AUC	\mathbf{SP}
	$\phi_{M_{12}}$	4.196	0.883	0.817
SGPMC	$\phi_{M_{32}}$	8.622	0.889	0.828
SGI MC	$\phi_{M_{52}}$	9.466	0.887	0.829
	ϕ_{SE}	10.188	0.879	0.832
	$\phi_{M_{12}}$	4.060	0.881	- 0.171
SGPR.	$\phi_{M_{32}}$	3.019	0.877	- 0.167
SGIII	$\phi_{M_{52}}$	3.466	0.876	- 0.165
	ϕ_{SE}	4.740	0.873	- 0.229
	$\phi_{M_{12}}$	7.167	0.870	0.810
SVGP	$\phi_{M_{32}}$	9.316	0.876	0.820
SVGF	$\phi_{M_{52}}$	9.941	0.874	0.821
	ϕ_{SE}	9.712	0.857	0.818

Table 3.5: Mean results of the 40-fold cross-validation tests for the SGP model selection; execution time, the AUC and the Spearman (SP) correlation index. \mathcal{M} stands for model and \mathcal{K} for kernel.

In contrast, the SVGP and SGPMC models present a very high SP index. Which means that the regions with higher variance coincide with the higher regression error. Such an improved prediction of the posterior variance in the SVGP and SGPMC model configurations is associated with the use of a more suitable likelihood function. As described in Section 3.2.1, both models have been configured to use a beta function with an s=1 scalar factor. While the SGPR model assumes, by definition, a Gaussian likelihood distribution.

The uncertainty representation of the SGP depends on the capacity of the likelihood function to adapt to the input data distribution. The use of a beta function as a likelihood function of the SGP is key to capture the distribution probability of our input data, even when the samples in a dataset are biased towards one label value.

Taking into consideration SGP configurations that use a beta likelihood function, the SGPMC models with $\phi_{M_{12}}$ or $\phi_{M_{32}}$ kernels stand out from the rest. They provide low execution times whilst maintaining high AUC and SP metrics. Figure 3.10 provides more details about these two SGP configurations.

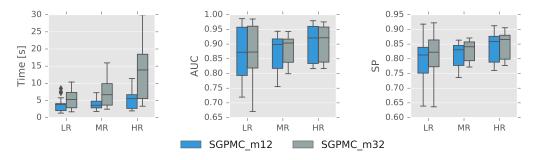


FIGURE 3.10: Comparison of the SGPMC models with $\phi_{M_{12}}$ or $\phi_{M_{32}}$ kernels. Execution time, AUC and SP results considering all three field experiments and distinguishing the results obtained for each resolution configuration (LR, MR and HR).

The figure distinguishes between the resolution configurations used and shows of the results' distribution. Using a $\phi_{M_{12}}$ kernel results in faster execution times than using a $\phi_{M_{32}}$ for all resolution configurations analyzed. The $\phi_{M_{12}}$ results in slightly lower AUC and SP values than the $\phi_{M_{32}}$ for the same resolution configuration. However, for the same execution time budget, the SGP model with a $\phi_{M_{12}}$ kernel can be computed with a higher resolution than $\phi_{M_{32}}$, providing improved performance. For instance, the

SGMPC $\phi_{M_{12}}$ configuration, trained in MR, results in lower execution times and higher AUC and SP than the SGMPC $\phi_{M_{32}}$ configuration trained in LR.

Thus, the SGPMC model with likelihood function beta s = 1 and kernel $\phi_{M_{12}}$ is selected as the default SGP configuration for the proposed methodology.

3.4.2 Validation

Raw data validation

Table 3.6 presents the results of the classification performance for each dataset and tested resolution configuration.

Table 3.6: Resulting AUC values of the raw data obtained from field-tests FT1-3 and resolution configurations LR, MR and HR

AUC	FT1	FT2	FT3
LR	0.983	0.911	0.906
MR- HR	0.984	0.908	0.912

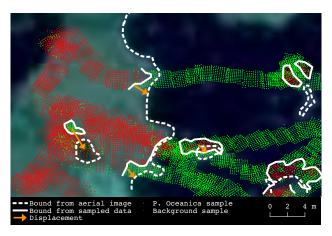


FIGURE 3.11: Meadow bounds representation to validate the consistency of the raw data obtained from FT1. The circular dots represent the data samples, the color indicates the sample value. The white lines represent the meadow bounds inferred from the aerial images and from the samples. The orange arrows represent the distance between both inferred bounds.

From these results, no correlation has been found between the resolution configuration used and the AUC obtained. In other words, a high resolution of the segmented point cloud does not condition the resulting AUC. One explanation for this is that higher resolutions preserve more boundary information, whilst lower ones take into account wider areas, weakening boundary errors.

With regard to localization consistency, we compare the location of the meadow bounds that can be inferred from the raw data of a dataset and the bounds inferred from an aerial image. Figure 3.11 shows the raw data obtained from FT1 overlaid on top of a geo-localized aerial image.

The meadow bounds inferred from the aerial images and from the collected data have an offset of approximately 2 meters. Considering that data was recorded using an ASV with a trajectory estimation based on RTK GPS with an accuracy given by the manufacturer of < 0.1m, the localization mismatch cannot be categorically linked to the trajectory estimation, but also to errors induced by the aerial image projections to the WGS 84 global reference system.

GP prediction validation

Figures 3.13, 3.14 and 3.15 show the grid maps and uncertainty maps obtained for the three datasets using an MR resolution configuration. The grid samples shown in the grid map represent the probability of *P.oceanica* presence in a determined location; the uncertainty map represents the variance obtained from the GP regression after training the SGPMC $\phi_{M_{12}}$ model using the aforementioned grid samples.

The results obtained from CV analysis are represented in the box plots of Figure 3.12. They show important differences depending on the test. In the case of execution time, this difference mostly comes from the fact that the computation complexity of an SGPMC increases as the number of samples increases since $\mathcal{O}(mn^2)$ [138], where m represents the quantity of inducing points and n the quantity of training samples; such quantities are proportional to the area of the extension covered, which are $262m^2$, $282m^2$ and $117m^2$ for FT1, FT2 and FT3, respectively. In terms of the AUC and SP metrics, the results for each environment depend on the spatial distribution complexity of the seagrass, i.e. background patch size and meadow edge sharpness.

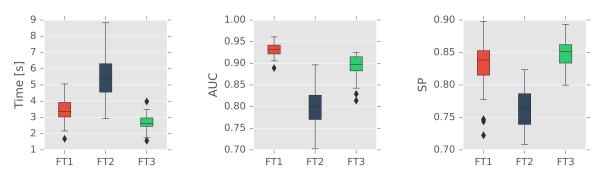


FIGURE 3.12: Results obtained for the validation of the SGPMC-beta model with kernel m12 using an MR resolution configuration. Each plot represents the mean and dispersion of a given metric (time, AUC and SP) after running 40-fold cross-validation iterations for the field-tests FT1, FT2 and FT3.

As expected, FT1 presents a best-case scenario resulting in low execution times and high AUC and SP. These results are due to good image segmentation performance (proven in Section 3.4.2), accurate localization and a seagrass meadow containing sharp edges. The uncertainty map provides a proper illustration of the high uncertainty areas, Figure 3.13. The GP is able to model uncertainty in information rich areas, such as heterogeneous data regions which contain small paths and sharp bounds, as well as in unrecorded spots close to the heterogeneous data areas. Notice that Figure 3.13b shows high GP variance near the meadow bound, and in the nearby non-sampled areas. In turn, non-sampled areas surrounded by the same sample labels result in low variance.

FT2 shows the worst performance due to (i) a low image segmentation performance caused by hard differentiable background spots, (ii) small localization errors due to the low-frequency underwater positioning system and (iii) low sampling redundancy. Figure 3.14 shows the resulting sample and uncertainty maps. Even in such a worst-case scenario, the uncertainty map highlights information-rich areas of the surveyed seabed.

FT3 contains the same localization inaccuracies as FT2. However, since background areas in the seagrass meadow are bigger, and more overlapping data has been recorded, the GP predictions are more accurate. The uncertainty map perfectly captures the information-rich areas of the target region.

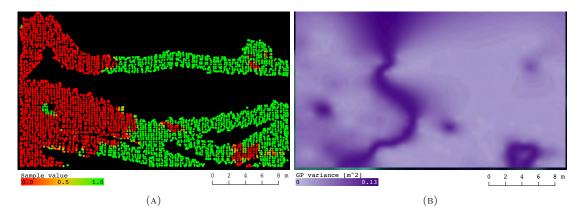


FIGURE 3.13: Resulting maps of FT1 experiment. The top image (a) shows the grid map. It represents the samples used for training the GP; the sample color represents the sample label—i.e. the probability of having $P.\ oceanica$ —. The bottom image (b) shows the uncertainty map; darker areas represent higher uncertainty. High uncertainty areas correspond to heterogeneous data areas and unrecorded spots close to the heterogeneous data.

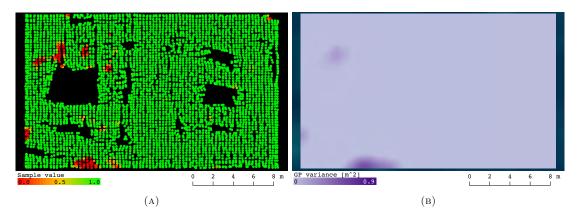


FIGURE 3.14: Resulting maps of FT2 experiment. The top image (a) shows the grid map. It represents the samples used for training the GP; the sample color represents the sample label—i.e. the probability of having *P. oceanica*—. The bottom image (b) shows the uncertainty map; darker areas represent higher uncertainty. The uncertainty map highlights information-rich areas containing heterogeneous samples.

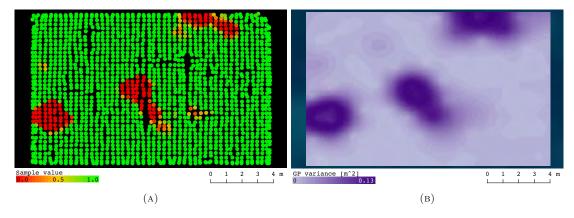


FIGURE 3.15: Resulting maps of FT3 experiment. The top image (a) shows the grid map. It represents the samples used for training the GP; the sample color represents the sample label—i.e. the probability of having *P. oceanica*—. The bottom image (b) shows the uncertainty map; darker areas represent higher uncertainty. The uncertainty map highlights information-rich areas far from the mean label value.

In short, all the tests showed that high GP variance values are found in information-rich areas. Areas containing similar sample values obtained low variance, and meadow bounds or patches resulted in high variance. In this sense, critical spots from which more data are required are highlighted.

3.5 Discussion

This chapter provides a method to ease data-gathering campaigns for *P. oceanica* meadows by performing (a) an online image segmentation using an encoder-decoder CNN, (b) a projection of such processed data to world coordinates using stereo image processing and the estimated robot trajectory and (c) an estimation of the collected data uncertainty using an SGP.

A series of field-tests were carried out in order to provide enough data to select the most suitable CNN and SGP configuration, as well as to validate the method in diverse *P. oceanica* environments.

In these tests, the encoder-decoder CNN showed that removing the last transponded convolutional layers of the VGG16-FCN8 network has a small impact on the segmented point cloud classification error and a substantial reduction in the execution time. It resulted in an AUC reduction of 0.2% in favor of a lower execution time of 12.8%.

With regard to the selection of a sparse SGP configuration, the experiments showed that the SGPMC model with kernel $\phi_{M_{12}}$ provides the best overall performance. Such a model configuration provided shorter training times and excellent uncertainty representation.

Furthermore, our application has shown that using a likelihood beta function with scaling factor s=1 in the SGP model is fundamental in order to model the uncertainty of our prediction model. Validation tests using such a likelihood function have shown that (1) the surveyed areas which are considered to be more salient show higher SGP variance than areas containing similar sample values and (2) the estimation shows good prediction properties in non-sampled areas too, inferring, for instance, meadow bound continuations.

One of the main benefits of this new methodology is the possibility to ease, speed up and reinforce the robustness and precision of geo-referenced data gathering processes. The method also permits easing and increase the frequency of the data gathering campaigns in repeated and precisely geo-referenced sites, which is key to inspect periodically the same ecosystem to observe changes and get temporal series of geo-referenced biological data in marine sites of special ecological interests, something very costly in time and resources using classical methods based on divers.

Producing spatial distribution predictions of seagrass whilst the AUV records images from a target region provides an increased perception of the environment, which results in improved supervision during data-gathering campaigns. This increased perception aids operational decision-making, for instance by performing a quick exploration of a target region, identifying bad data quality or modifying the robot trajectory in order to visit strategic spots. This methodology sets the basis for further research on mission replanning; such enhanced perception about the environment is key to perform autonomous decision-making and be able to develop robust data-driven exploration for environmental monitoring.

Chapter 4

Decision-time adaptive replanning

This chapter tackles the decision-making capacity required to drive an AUV adaptively replanning mission paths around a target area. The key idea behind the replanning method described in this chapter is the use of a semantic model of the environment generated online to adapt the mission paths and enable exploration.

This chapter describes a novel decision-time adaptive replanning (DAR) method for adaptive mission replanning which has been designed to execute successive informative paths, without stopping, considering the newest information obtained from an environment model.

Figure 4.1 provides an overview of the replanning method. It first builds a network of possible sampling locations inside a target area (build node network), establishing connections between neighbor nodes. Then, with a given fixed frequency it iteratively executes (1) an update node network process to query the modelling method for the recorded data density and SGP predictions at the sampling locations and update the network of sampling locations. Then, considering the actual mission status, (2) a get next initial state provides the starting state for the next IPP execution. Finally, (3) a planning process sequentially updates a search tree and retrieves the most informative path. Such path is stored to be launched as the robot finishes the current section maneuver.

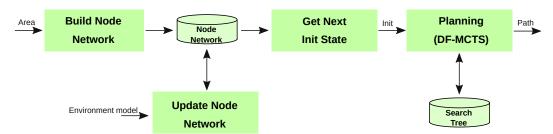


Figure 4.1: Decision-time adaptive replanning (DAR) method.

We solve the *information path planning* (IPP) problem by means of a novel *depth-first* (DF) version of the *Monte Carlo tree search* (MCTS). The DF-MCTS method has been designed to explore the state-space in a depth-first fashion, provide solution paths of a given length in an *anytime* and *non-myopic* manner, and reward smooth paths for field realization with non-holonomic robots.

The DAR method joins the advantages of graph-based and sampling-based methods, devoting some time at the initialization to produce a network of neighbor nodes (a graph) that is used for sampling paths and grow a decision tree. The consecutive planning iterations update such decision tree and select a

candidate path to execute in an anytime manner. Having a precomputed network of nodes boosts the sampling computations. This point, together with the fact of maintaining a decision tree alive between consecutive planning iterations are two key features to reduce the planning time and improve the overall performance.

The remainder of the chapter is organized as follows: Section 4.1 introduces some background about RL algorithms. Sections 4.2 and 4.3 describe the developed methods—i.e. DAR method and DF-MCTS strategy, respectively—. Section 4.4 describes the tests performed in simulation and illustrate the results. Finally, Section 4.5 summarizes the main conclusions and draws future research objectives.

4.1 Background on Reinforcement Learning (RL)

This section introduces the basis of the reinforcement learning formalization used to solve the IPP problem with the DF-MCTS strategy described in Section 4.3. The basic idea behind RL is to learn from interaction. The learning is based on trial and error; in real or simulated environments. In this work we will use RL in order to explore the space of possible paths to be executed by the AUV during its adaptive mission by interacting with the environment model. The selection of the best path to follow is considered a sequential decision process. Next sections provide the basis of the problem formalization in terms of a *Markov decision process* and solutions for such problem.

4.1.1 Markov decision processes (MDP)

MDPs are a classical representation of sequential decision processes [145], and are characterized by the Markov assumption: the decisions taken only depend on the current state. Moreover, such processes are called Finite MDP (FMDP) when the states and actions are finite and Partially Observable Finite MDP (POFMDP) when the state can not be completely observed.

MDPs are suited for RL problems, being useful to map situations to actions by maximizing a numerical reward signal; an agent interacts with the environment and gets a reward signal and a change of state. They are characterized by four main subelements; a policy, the reward signal, the value function and optionally, a model of the environment:

- The policy determines the action to be taken from a particular state. It can be (1) stochastic, where a probability is specified for each action following, for instance, a Gaussian or a uniform distribution; (2) deterministic, for instance taking the action with the highest expected reward in a greedy fashion; or even (3) ϵ -greedy, where a stochastic policy is used with a probability given by $\epsilon \in [0,1]$ or greedy otherwise.
- The reward signal comprises the goal of the RL problem. The objective of the RL problem is to maximize the total expected reward along an executed path.
- The *state-value function* estimates how good is for the agent being in a given state. It considers the total amount of reward that the agent can expect to accumulate over the future. Whilst rewards are given directly from the environment, values are obtained from trial experience. This value function takes into account the future rewards of the states that are likely to follow the actual state.
- The *model* describes the behavior of the environment. It predicts the transitions between states given an action, which is, the next state and reward from the current state and action. Whilst *model-free* methods learn by trial and error, *model-based* methods are able to consider future possible situations before they are experienced.

4.1.2 Iterative policy evaluation

In this work the IPP strategy described in Section 4.3 is approached as a POFMDP, in which the set of states, actions and rewards (S, A and R respectively) have a finite number of elements, and where the state can not be fully observed. The POFMDP is solved using a value function method based on iterative policy evaluation. In particular, we are using Monte-Carlo (MC) for the expansion and Temporal difference (TD) for the value backpropagation, methods detailed below, in Equations 4.11 and 4.12. In this case the Markov assumption can be translated into a probability of resulting a particular state s' and reward r, at a particular time t, given a preceding state s and action a:

$$p(s',r|s,a) \doteq \Pr\{S_t = s', R_t = r|S_{t-1} = s, A_{t-1} = a\},\tag{4.1}$$

for all $s,s' \in \mathcal{S}$ and $r \in \mathcal{R}$, $a \in \mathcal{A}(s)$. Such probability determines the dynamics of the MDP and is characterized by $\mathcal{S} \times \mathcal{R} \times \mathcal{S} \times \mathcal{A}(s) \to [0,1]$, where

$$\sum_{s'} \sum_{r} p(s', r|s, a) = 1, \text{ for all } s \in \mathcal{S}, a \in \mathcal{A}(s).$$

$$(4.2)$$

The state-value function $v_{\pi}(s)$ for an arbitrary policy function $\pi(s)$ is defined as,

$$v_{\pi}(s) \doteq \mathbb{E}_{\pi}\{G_t|S_t = S\} \tag{4.3}$$

This is the expected return on the long run (several time steps ahead). Where G_t is defined as the discounted sum of expected rewards such as,

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \dots \tag{4.4}$$

$$= R_{t+1} + \gamma \left(R_{t+2} + \gamma R_{t+3} + \gamma^2 R_{t+4} + \dots \right) \tag{4.5}$$

$$= R_{t+1} + \gamma G_{t+1} \tag{4.6}$$

where $\gamma \in [0,1]$ is the discount rate. With a $\gamma = 0$ the agent would be myopic, considering only immediate rewards. Whereas with a higher γ value the agent would have a stronger consideration on further rewards. From the last expression we can start revealing the recurrent nature of RL problems.

The state-value function can be further expanded using (4.3) and (4.6), and it can be deduced that,

$$v_{\pi}(s) \doteq \mathbb{E}_{\pi} \{ R_{t+1} + \gamma G_{t+1} | S_t = S \}$$
 (4.7)

$$= \mathbb{E}_{\pi} \{ R_{t+1} + \gamma v_{\pi}(S_{t+1}) | S_t = S \}$$
(4.8)

$$= \sum_{a} \pi(a|s) \sum_{s',r} p(s',r|s,a) \Big[r + \gamma v_{\pi}(s') \Big], \tag{4.9}$$

where the policy $\pi(a|s)$ provides the probability of choosing a particular action a from a state s. The Bellman equation (4.10) provides an iterative expression to find an optimal solution $v_*(s) \doteq \max_{\pi} (v_{\pi}(s))$ to the RL problem as $k \to \infty$. Dynamic programming (DP), Monte-Carlo (MC) and Temporal difference (TD) methods can be used to perform such iterative policy evaluation.

$$v_{k+1}(s) \doteq \sum_{a} \pi(a|s) \sum_{s',r} p(s',r|s,a) \left[r + \gamma v_k(s') \right]$$
 (4.10)

DP can be used to incrementally compute optimal policies when a perfect model of the environment is available. However, they require a fully observable environment, they can not be used to solve partially observable MDP. The value update for MC and TD methods is performed as follows:

$$MC: v_{k+1}(s) \doteq v_k(s) + \lambda \Big(G_t - v_k(s)\Big)$$

$$\tag{4.11}$$

$$TD: v_{k+1}(s) \doteq v_k(s) + \lambda \Big(r + \gamma v_k(s') - v_k(s) \Big)$$
 (4.12)

where the λ parameter defines the learning rate.

The basic difference between them is that whilst MC learns after trial, TD guesses from guesses, that is MC uses the expected reward value G_t and TD uses the discounted value of the next step $v_k(s')$, it bootstraps. Using either MC or TD, the value of one state can be updated from the rewards obtained on successive steps following a default policy.

4.1.3 Decision-time planning

Decision-time planning methods are used to plan a series of decisions from a root state [145]. These methods use simulated experience from a model to improve a policy or a value function. Instead of solving the whole MDP, they focus on solving a sub-MDP. They plan using the MDP model to look ahead from the current state. Here, the approximate value function is obtained on leaf nodes of the current state and then propagated backwards to the current state.

Rollout algorithms [145] are a type of decision-time planning methods, based on Monte Carlo control. They sample multiple trajectories in a depth-first fashion from a root state. Each trajectory (or rollout) consists in taking successive decisions according to a given default policy until a terminal state is reached. They are used to obtaining near-optimal decisions by taking random samples in a decision space and building a search tree according to the results. They are useful for artificial intelligence applications with large states and actions spaces. Moreover, this kind of methods are a good fit for our applications since they are anytime, always provide a solution, where more computing power leads to better performance.

Monte Carlo tree search (MCTS) is a particular rollout method improved to bias the growing of a decision tree towards the highest valued regions. A tree is started from a root node, and grows iteratively following the next iteration steps: (1) Selection of the highest valued non-exhausted and non-ending node according to some tree policy. (2) Expansion of the tree. Randomly select a non-taken action to connect a new node to the selected one. (3) Simulation of a playout. Propagate the new node by selecting sequential actions according to a default policy until a given budget is exhausted, and compute the new node value. (4) Backpropagation of the node value upwards through the tree. Once the search is interrupted an action of the root node is selected according to some predefined criteria. For instance select the action that conduces to the highest valued child, the most visited child, or a weighted function between value and visits. The key parts of the MCTS algorithm are the tree policy, the default policy and the value function; which can be shaped based on the nature of the problem domain.

The limitation of MCTS for our application lies in the fact that, whereas MCTS explores the environment in a depth first manner, the decision tree tends to grow exhaustively, resulting in a shallow tree if a short planning time is given or a high discount factor is used. For this reason we propose a novel variation, the DF-MCTS, described in Section 4.3.

4.2 Decision-time adaptive replanning (DAR)

This section describes the developed strategy for decision-time adaptive replanning (DAR). This DAR method has been designed to be executed online while the AUV is moving; the planner iteratively command new mission paths considering the newest data available as the robot navigates and the GP-based environment model is updated. It consists in four key ideas:

- Building and updating a network of nodes N that results in a pre-initialization of part of the content included in the states and actions considered during decision-time planning.
- Having the AUV in constant motion, while being flexible to execute updated missions. The robot is neither stopped for planning nor forced to complete the commanded mission paths.
- Growing a search tree in a depth-first fashion, following a novel DF-MCTS strategy for decision-time planning.
- Recycling part of the last search tree for successive planning executions.

The main structure of the method is illustrated in Algorithm 1. It inputs: a target area \mathcal{A} and an obstacle region \mathcal{O} defined as polygon shapes in global coordinates, a model of the environment \mathcal{M} containing the SGP model and the k-d tree used for computing the raw data density, an initial state s_0 defined by the initial AUV pose, the sampling nodes density ρ configured, and the nearest neighbor distances d_1 and d_2 set. The path to execute \mathcal{P} and the remaining path \mathcal{P}_{s_0} are initialized empty. The algorithm starts generating a set of sampling nodes \mathcal{N} , and then iteratively: (1) updates the nodes utility using the newest environment model \mathcal{M} , (2) gets the next initial state s_0 if a path \mathcal{P} has been previously generated, (3) gets a path \mathcal{P}_{s_0,s_n} using the DF-MCTS strategy, and (4) saves the path.

Algorithm 1: DAR()

```
Input: \mathcal{A}, \mathcal{O}, \mathcal{M}, s_0, \rho, d_1, d_2
\mathcal{P} \leftarrow \varnothing; \mathcal{P}_{s_0} \leftarrow \varnothing;
\mathcal{N} \leftarrow buildNodes(\mathcal{A}, \mathcal{O}, d, d_1, d_2)
while \neg stopCondition() do
| \mathcal{N} \leftarrow updateNodes(\mathcal{N}, \mathcal{M}) 
if \mathcal{P} \neq \varnothing then
| s_0 \leftarrow getNextInitState(\mathcal{P}) 
end
| \mathcal{P}_{s_0, s_n} \leftarrow getPath(s_0) 
\mathcal{P}_{s_0} \leftarrow getRemainingPath(\mathcal{P}, s_0) 
\mathcal{P} \leftarrow \mathcal{P}_{s_0} + \mathcal{P}_{s_0, s_n} 
savePath(\mathcal{P}) 
end
```

4.2.1 Node network

The Algorithm 2 describes the generation of the node network \mathcal{N} (of Algorithm 1). It randomly generates a set of nodes \mathcal{N} inside a given target area \mathcal{A} and outside of given obstacle areas \mathcal{O} . The number of nodes n to build is determined by a desired node density ρ and the area of \mathcal{A} . Then, a k-d tree is build using the locations of the nodes contained in \mathcal{N} for quick nearest-neighbor lookup. The node network is build by querying the k-d tree for two sets of neighbors for each node; a first set of nodes \mathcal{N}_1 located at a distance $d < d_1$, and a second set of nodes \mathcal{N}_2 located at a distance $d > d_1$ and $d < d_2$. Being the distance thresholds $d_1 < d_2$.

Algorithm 2: buildNodes()

```
Input: \mathcal{A}, \mathcal{O}, \mathcal{M}, \rho, d_1, d_2
      // Get sampling locations:
      a \leftarrow Area(\mathcal{A})
      n \leftarrow a \cdot \rho
      while k < n do
      p \leftarrow getRandomLocation(A)
      if (p \notin \mathcal{O}) then
           \mathcal{N} \leftarrow append(\mathcal{N}, p)
           k \leftarrow k + 1
      \mathbf{end}
      end
      // Set neighbors:
      kdt \leftarrow KDTree(\mathcal{N})
      for N \in \mathcal{N} do
      \mathcal{N}_1 \leftarrow getNearNodes(kdt, N, d_1)
      \mathcal{N}_2 \leftarrow getNearNodes(kdt, N, d_2) - \mathcal{N}_1
      setNodeNeighbors(N, \mathcal{N}_1, \mathcal{N}_2)
      end
```

The Algorithm 3 describes the update of the node network \mathcal{N} (of Algorithm 1). The objective is updating the sampling utility at the \mathcal{N} locations. The model \mathcal{M} is queried to get the data density and SGP prediction in the \mathcal{N} locations. Then, the information gain I and the utility value U of the nodes in \mathcal{N} are computed, that will be used to guide the IPP.

```
Algorithm 3: updateNodes()
```

```
Input: \mathcal{N}, \mathcal{M}

D \leftarrow getDensity(\mathcal{M}, \mathcal{N})

\mu, \sigma^2 \leftarrow getPrediction(\mathcal{M}, \mathcal{N})

I \leftarrow computeInformation(\mu, \sigma^2)

U \leftarrow computeUtility(D, I)

\mathcal{N} \leftarrow D, \mu, \sigma^2, I, U
```

4.2.2 Information gain and node utility

The information gain I is computed from the prediction obtained with the SGP model at the node location. We have considered two options to compute such information, either using the differential entropy (DE) I_{DE} or the upper confidence bound (UCB) I_{UCB} functions, Equations (4.13) and (4.14), respectively.

$$I_{DE} = \frac{1}{2} \ln \left(2 \cdot \pi \cdot e \cdot \sigma^2 \right) \tag{4.13}$$

$$I_{UCB} = \mu + 1.96 \cdot \sqrt{\sigma^2} \tag{4.14}$$

Whilst I_{DE} uses the variance σ^2 provided by the SGP in the query locations, I_{UCB} also uses the mean semantic label μ provided by the GP. I_{UCB} provides higher values to higher mean label locations, which increases exploitation (coverage) in such locations at the expenses of reducing exploration in lower mean label locations. Moreover, instead of using the I directly for planning we build a utility value U to leverage the I with the neighboring data density D such as:

$$U = I' \cdot (1 - D')^{\alpha} \tag{4.15}$$

The objective of using such U is to attenuate the interest of visiting areas that do not present a reduction on the predicted information I, even though they have been repeatedly recorded (high density). This situation may happen in areas with heterogeneous data, such as meadow boundaries. Moreover, since the resolution of the SGP model is bounded, recording more data on a high information location does not imply improving the fitness of the SGP model. I' and D' represent the normalized information and density values for all the nodes to the range [0,1] using a min-max normalization. The α parameter works as a weight on the density factor, the higher the α the bigger is the impact of the density D.

4.2.3 Replanning

In order to allow a continuous navigation and a flexible path execution, we take into account the time expend in planning in order to set the initial state used for next planning iteration. The distance between the actual position and the next initial state s_0 has to be larger than a minimum distance $d_{min} = v_{max} \cdot t_{plan}$, where v_{max} is the maximum speed of the robot and t_{plan} is the planning time. The next initial state s_0 used for next planning is selected taking into account the distance covered by the robot while the planning process is executed. As the robot finishes a section maneuver, it executes the latest planned path.

4.3 Depth-first Monte Carlo tree search (DF-MCTS)

This section describes the IPP strategy developed to provide mission paths to the DAR method. Considering the high number of possible states and actions available in our field robotics application and the necessity of an online realization, a decision-time planning method has been developed to solve the IPP problem. DF-MCTS is different from MCTS in one key aspect: it keeps all the states traversed during the rollout in the search tree. This provides a faster growth of the tree, and guarantees a solution path of a given length. The limitation of MCTS for our application is that whereas MCTS explores the environment in a depth first manner, the decision tree tends to grow exhaustively resulting in a shallow tree if a short planning time is given or a high discount factor is used.

4.3.1 Problem formalization

We propose to solve the IPP problem using a RL-based algorithm using a finite number of non-fully observable states. For that end, a POFMDP represents the sequential decision problem.

States

A state S_k is defined by an associated node N, a parent state S_{k-1} , a set of action candidates $\mathcal{A}(S_k)$, a state value V, a distance cost c from the initial state, and an orientation θ ; $S_k \leftarrow \{N, S_{k-1}, \mathcal{A}(S_k), V, c, \theta\}$.

Nodes

The nodes are used to provide a discrete representation of the possible sampling locations in the target area. A node N is defined by a north-east position x, a raw data density value d, an information gain i, a utility value u, and two sets of neighbor nodes \mathcal{N}_1^N and \mathcal{N}_2^N ; $N \leftarrow \{x, d, i, u, \mathcal{N}_1^N, \mathcal{N}_2^N\}$. The nodes are initialized at the initialization of the DAR method and updated before each IPP execution as described in Section 4.2.1.

Actions

The actions are directly associated to the transition to a given neighbor node. So, the selection of an action directly means the selection of a neighbor node to be visited next. Taking a particular action from a given state S_k will result on the transition to the neighbor node considered in such action, and in the creation of a new state S_{k+1} . The set of action candidates $\mathcal{A}(S_k)$ of a given state S_k will be composed of a subset of the neighbor nodes \mathcal{N}_2^N of its associated node N such as $\mathcal{A}(S_k) \in \mathcal{N}_2^N$

4.3.2 Method

The structure of the proposed method is represented in Figure 4.2 and includes 3 sequential steps: selection, expansion and backpropagation. It starts building a tree \mathcal{T} from a root state s_0 (initial state from DAR of Section 4.2). Then, it iteratively selects a high valued node, expands the tree with multiple rollouts, and backpropagates the state-values; selection, expansion and backpropagation.

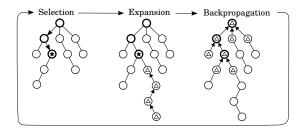


FIGURE 4.2: One iteration of the proposed DF-MCTS algorithm. The stared state represents the selected state for expansion, the states with a triangle correspond to the states used for update the tree values for the expansion and the backpropagation steps.

The darker circles represent the states whose values are updated.

Remark 1 (Keep search tree). For successive plannings, the tree structure that hangs from the next initial state s_0 , from now on called \mathcal{T}' , is conserved. And prior to the next IPP execution \mathcal{T}' is traversed to update the distance cost, extend the leave states and update the state values to the newest map estimation.

Selection

Select a state from the tree \mathcal{T}_{ne} to expand, where \mathcal{T}_{ne} includes the non-exhausted states of \mathcal{T}' -i.e. state containing non-tested actions—. The state selection follows a tree policy based on an ϵ – greedy method (with a parametrizable epsilon), which consists in getting a random number r and evaluating if r is bigger or lower than epsilon. If $r > \epsilon$, the highest valued state s^* from \mathcal{T}_{ne} is selected. Otherwise, get a random state. Being,

$$s^* = \operatorname*{argmax}_{s \in \mathcal{T}} v(s) \tag{4.16}$$

Expansion

Expand the tree following a given default policy. This step is different from the expansion step in MCTS algorithms in the following: DF-MCTS (a) performs multiple rollouts from the same selected state, and (b) the states traversed are kept in the tree and are considered for further tree expansions.

An MC rollout is build by iteratively (1) searching for candidate actions $\mathcal{A}(s)$ to the current state s, (2) selecting an action $a \in \mathcal{A}(s)$ according to a given default policy $\pi(a|s)$, (3) performing action a and (4) getting next state s' and reward r until a distance budget \mathcal{B} is exhausted.

We define the default policy $\pi(a|s)$ with a uniform distribution (4.17) and deterministic action-state transitions (4.18). We assume that the execution of action a being in state s results in a new state s' located on a neighboring node position pointed by action a.

$$\pi(a|s) = \frac{1}{\|\mathcal{A}(s)\|}, \text{ for all } a \in \mathcal{A}(s)$$
(4.17)

$$p(s'|s,a) = 1$$
, for all $s \in \mathcal{S}, a \in \mathcal{A}(s)$ (4.18)

The set of action candidates $\mathcal{A}(s)$ of a given state s is built when a state is visited for the first time. These actions include the nodes that can be reached one step ahead. The set of actions of a given state $\mathcal{A}(s)$ are build checking the node network built in the DAR initialization, see Section 4.2.1. In order to get smoother rollout trajectories, two sets of action candidates are used: \mathcal{A}_1 and \mathcal{A}_2 , represented in Figure 4.3.

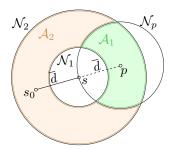


FIGURE 4.3: Two sets of action candidates for a given state s: $\mathcal{A}_1 = \mathcal{N}_2 \cap \mathcal{N}_p$ and $\mathcal{A}_2 = \mathcal{N}_2 - \mathcal{N}_p$. Where \mathcal{N}_p includes the nodes located at a distance d from p and $\overline{s} \ \overline{p} = \overline{s_{parent}} \ \overline{s} = \overline{d}$.

The set $\mathcal{A}_1(s)$ includes the priority actions. The default policy $\pi(a|s)$ will use the set $\mathcal{A}_1(s)$ if it is not empty, otherwise it will use the set $\mathcal{A}_2(s)$. Besides, afterwards an action candidate a is used, it is removed from its action candidate set. As a result from this, as the tree grows, some states get exhausted of action candidates. Such states are considered exhausted and are removed from the \mathcal{T}_{ne} used for selection.

The values v(s) of the states traversed during rollout are updated using the state-value function (4.3). That, considering (4.1) and (4.17), can be rewrote as:

$$v(s) = r(s) + \gamma v(s'), \tag{4.19}$$

where r(s) represents the reward generated in state s.

In order to generate smooth trajectories the relative turns between consecutive states are considered in the reward function r(s). Defined as

$$r(s) = u(s) \cdot \cos^{\omega} \frac{\theta_r(s)}{2},\tag{4.20}$$

where $u(s) \in U$ is the utility value u of the node belonging to state s obtained from the update of the node network using Equation 4.15, and $\theta_r(s)$ is the relative angle between the orientations of s and its parent state. The second term has been designed to penalize sharp turns, rewarding small variations of $\theta_r(s)$, and is contained in the interval [0,1]; the ω acts as a parameterizable weight. Since $u \in [0,1]$ (Section 4.2.1) the reward will be unitary $r \in [0,1]$. Computing the value of the rollouts is a mater of sequentially computing the discounted sum of rewards. Furthermore, as the adaptive replanning advances and more data is recorded, the map estimation gets more accurate, which impacts in a more accurate utility value

and reward signal. Hence, the more accurate the SGP regression becomes the more accurate the reward signal becomes.

Backpropagation

Back-propagate the rollout values to the preceding states. We use for this a value function based on the TD expression (4.12) presented in Section 4.1.2. However instead of updating the state value using only a child state value, we use the mean value of all child states $s_c \in \mathcal{S}_c(s)$ for this value update.

$$v_{k+1}(s) = v_k(s) + \lambda \left[r + \gamma \left\| \sum_{s_c \in \mathcal{S}_c} v_k(s_c) \right\| - v_k(s) \right]$$

$$(4.21)$$

Assuming that there will not be child outlier values is necessary and strongly supported in the fact that the child's state value accuracy is directly determined by the rewards signal accuracy, which directly depends on informativeness accuracy computed from the SGP estimation, elements already discussed previously. Although the SGP estimation might be very inaccurate at the beginning of the exploration due to the limited data, as they progress, GPs with the configurations used have shown to provide very smooth and valued predictions that inhibit the possibility of outliers in the child state values.

4.4 Experiments and Results

This section addresses the evaluation of the DF-MCTS strategy and the DAR method by executing a series of tests in batch. Here, the image acquisition process is simulated by generating a series of image keyframes on top of the execution paths. Then, a background groundtruth image representing the simulation environment is used to clip the simulated images considering their keyframe positions, a given field of view, orientation and localization noise.

For this evaluation, we consider two variations of the DF-MCTS strategy that differ in the planning constraints applied. Table 4.1 enumerates the common parameters for both configurations. And Table 4.2 shows the specific parameters for both configurations, constrained (C1) and unconstrained (C2). The constrained is more conservative with respect to real robot conditions: it (i) penalizes high density areas in the utility function (non-zero alpha value), (ii) penalizes turns in the reward function (non-zero omega value), and (iii) considers a larger minimum distance between successive states easing robot's navigation (larger neighbor distance d_1).

Table 4.1: Default parameter configurations for the DF-MCTS experiment.

DF-MCTS	Value
Planning time $[s]$	15
Information function	DE
Discount factor	0.8
Learning rate	0.9
Rollouts number	32
Epsilon	0.01
Number of nodes	20000

In addition, Table 4.3 summarizes the configuration parameters of the SGP model used for spatial modelling.

Section 4.4.1 starts introducing the metrics used for the assessment. Then, Section 4.4.2 describes the tests and results obtained to evaluate the information richness of the DF-MCTS strategy. And Section

Table 4.2: Specific parameter configurations for the DF-MCTS experiment.

Configuration Parameter	C1	C2
Alpha (Utility)	2.0	0
Omega (Reward)	1.5	0
Minimum neighbor distance d_1	2.0	1.0

Table 4.3: SGP configurations for the DAR experiment.

GP	
Model	SGPMC
Kernel	Matérn 32
Kernel length scale $[m]$	30
Likelihood	Beta
Likelihood scale	0.5
Optimizer	Scipy
Iterations	2500
Number of SGP training samples	1600
Number of SGP inducing points	200

4.4.3 describes a set of tests performed to evaluate the DAR method and compares the use of (i) the DAR method integrating a novel DF-MCTS strategy, (ii) the DAR method integrating a sampling-based strategy, and (iii) the execution of preprogrammed mission paths.

4.4.1 Evaluation Metrics

The results are presented using the following evaluation metrics, which are obtained after each iteration of the environment modelling layer.

• Mean of the Differential Entropy (MDE): Provides the mean map information, and is computed using the variance value of the environment model prediction σ^2 at n locations inside the target area.

$$MDE = \frac{1}{n} \sum_{i=1}^{n-1} DE(i)$$
 (4.22)

$$DE(i) = \frac{1}{2} \ln \left(2 \cdot \pi \cdot e \cdot \sigma^2(i) \right)$$
 (4.23)

• Standard deviation of the Differential Entropy (SDE): Provides the standard deviation of the pixelwise map information, also computed using the variance value of the environment model prediction σ^2 at n locations inside the target area.

$$SDE = \sqrt{\frac{1}{n} \sum_{i=1}^{n-1} [DE(i) - MDE]^2}$$
 (4.24)

• Area Under the Curve of the Receiver Operating Characteristics (AUC ROC): This metric is used only on the simulation tests, using the simulation groundtruth GT represented in Fig. 6.5a. Such GT is a precise monochrome representation of the simulation environment since, in fact, is the source for building the seabed textures distribution containing P. oceanica and sand. Given such mono-chrome image GT and a mean prediction (obtained from the last SGP model) with the same shape and resolution, the ROC diagram is obtained by plotting the pixel-wise true positive

rate (TPR) with respect to the pixel-wise false positive rate (FPR) using a varying discretization threshold for the mean SGP prediction. The AUC ROC provides the area under the ROC curve.

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{TN + FP}$$
(4.25)

$$FPR = \frac{FP}{TN + FP} \tag{4.26}$$

Where TP, FP, TN, FN represent the number of true positives, false positives, true negatives and false negatives for a given threshold value in the interval [0,1].

• Coverage percent (CP): Provides a percentage measure of the recorded data extension. It is computed as the recorded area A_r divided the total area A_t of the target region, where the recorded area A_r is obtained as the total number of samples n_c contained in the target region multiplied by the squared sample resolution R.

$$CP = 100 \cdot \frac{A_t}{A_r} = 100 \cdot \frac{A_t}{n_c * R^2}$$
 (4.27)

4.4.2 **DF-MCTS strategy for IPP**

This section addresses the evaluation of the DF-MCTS strategy. The objective is to evaluate the information richness of the mission path provided separately from the replanning. Every test executed here is composed only by a tree search and a best path selection. For these tests, we use a pre-trained SGP model of the environment. Figure 4.4 shows the prediction given by the pre-trained SGP model in the target area. The yellow areas represent the higher uncertainty regions-i.e. a large variance on the SGP prediction—, the gray areas represent lower uncertainty regions.

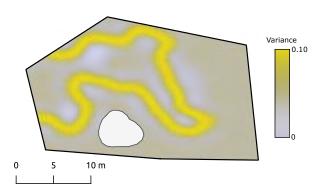


FIGURE 4.4: Prediction of the pre-trained SGP model used for testing the DF-MCTS strategy; the variance of the SGP prediction is represented with a color map ranging from yellow to gray for areas with high variance to areas with lower variance. The white background shape inside the region represents an obstacle region.

The resulting paths quality is evaluated using the information richness of the paths. Such information richness is obtained by computing the integral of the DE along each path, that is, (1) the resulting paths are split into $\delta \lambda = 0.01m$ steps, (2) the variance of the environment model is queried at each step, (3) the DE is computed using equation (4.13).

Figure 4.5 shows the resulting tree search and best path selection for a particular execution using configurations C1 and C2. Both configurations grow a search tree precisely over areas with high uncertainty-i.e. high values of the SGP prediction variance. However, they provide opposite features; whilst C1 shows a smoother tree search and selected path, the C2 grows a search tree higher fitting over the high uncertainty area. The resulting DE for both paths is $PI_{C1} = 0.35$ and $PI_{C2} = 0.32$.

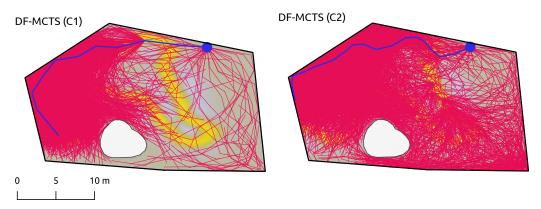


FIGURE 4.5: Results obtained for the two DF-MCTS variations; C1 constrained and C2 unconstrained. (a) shows the search tree (red) and the selected path (blue) on top of the predicted variance of the SGP model used (yellow for higher variance). C1 results in a smoother path, and C2 in a higher information path; $PI_{C1} = 0.35$ and $PI_{C2} = 0.32$.

After performing 20 executions with the same conditions for each configuration we obtained the results represented in Figure 4.6; each blue line represents the selected path of each iteration test. Using C1 clearly results in smoother paths than using C2 at expenses of a lower accuracy fitting to the high variance areas.

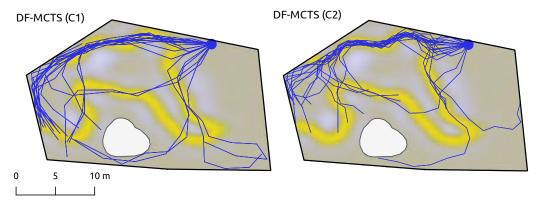


FIGURE 4.6: Paths obtained after performing 20 DF-MCTS executions for the same environment model and initial position, using configurations C1 and C2.

Finally, Figure 4.7 shows a comparison of the DE obtained for the benchmarking sampling strategy (green) and DF-MCTS using configurations C1 and C2 respectively (blue and orange). Each method has been executed 20 times for the same environment model and 5 different initial positions, 100 tests per method.

These results show a better performance of the DF-MCTS when compared with the sampling-based strategy. The sampling-based strategy returns very short paths, while DF-MCTS results in longer paths which are near to the configured path distance and are more informative.

The resulting difference between the use of the configuration C1 and C2 on DF-MCTS is that C1 tends to result in longer paths while C2 tends to result in slightly higher information paths. In general, the use of the DF-MCTS with configuration C2 provides smoother trajectories at expenses of a slightly lower informativeness. The smoothness on the executed path is expected to impact the sampling speed as well as

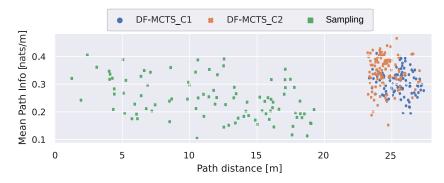


FIGURE 4.7: Representation of the mean path information and path distance obtained for 20 planning executions in 5 different initial positions for the same environment model and the two DF-MCTS configurations C1 and C1 described in Table 4.2.

the resolution of the recorded data. Hence, a further analysis of the trade of between path informativeness and path smoothness would be beneficial to generate a tailored setting of the DF-MCTS configuration. This would depend on the recording footprint of data gathering sensor used and the specific navigation settings. The reductions on the recorded data resolution caused by sharp turns is studied in Chapter 5.

Hence, in this section we have analyzed the capacity of the IPP strategy to provide a mission path in terms of path information and smoothness. It has been shown that there exists a trade-off between such values. The higher the path informativeness obtained the higher the exploitation of the knowledge about the environment, but what about exploration? In this analysis we have considered one-step planning with a fixed environment model, next section provide further analysis for sequential replanning.

4.4.3 DAR method for replanning

This section goes one step forward in the analysis. Instead of testing the IPP strategy with a fixed environment model, now the test consists in the execution of successive replanned paths using the DAR method. The main idea behind this simulation pipeline is to be able to execute replanning tests prior to the integration on an AUV software architecture for field-testing, and compare the resulting performance with the execution of a different sampling-based strategy and with the execution of predefined lawn mower patterns.

Each test starts executing an initial path, followed by a sequential execution of the DAR methodology. When a mission plan is enabled, the simulation process (1) gets the starting position and orientation, (2) retrieves an image from the groundtruth map with added Gaussian noise $\mathcal{N}(0,0.45m)$ in the position, (3) advances 1m the position along the trajectory, and iteratively repeats steps (2) and (3) until termination. The mission plans are not fully executed, only a proportion of the full path distance is executed. Such partial execution simulates the distance that the AUV would navigate for a given replanning time, until a new path is commanded.

Then, every iteration of the sequential replanning simulates the image recording of a part of the last commanded path, models the environment by processing the simulated images, trains an SGP, computes a mission path following an IPP strategy and commands the execution of the resulting path.

A total of four target areas of different shapes and sizes have been used for testing. Figure 4.8 show the hand-labeled groundtruth image used for image clipping, which is based on the seagrass coverage that can be inferred from aerial images, in white the presence of *P. oceanica* and in black the absence of such seagrass. The four target areas used for testing have been drawn on top.

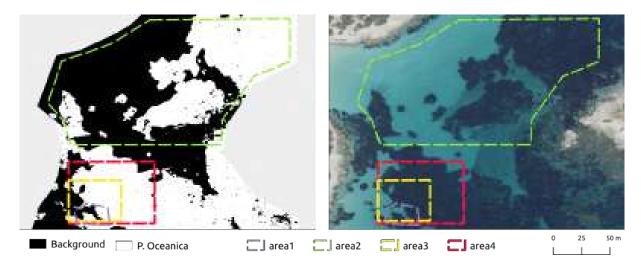


FIGURE 4.8: Aerial image and groundtruth used for testing the DAR method. The four target areas used for the simulation experiments are represented in different colors. The aerial image used to generate the groundtruth has been obtained from the PNOA 2015 campaign of the $Instituto\ Geodésico\ Nacional\ (39.534372288N,2.590594628E)$, located in the south-west coast of Mallorca Island

Moreover, Table 4.4 provides the total area of each target area, and some specific configuration parameters that depend on the target area scale such as the path budget, the maximum neighbor distance d_2 , the grid-filter resolution and the induction point density. In order to generalize to different target areas for these tests, the path budget \mathcal{B} has been set to $\mathcal{B} = 1.2 \cdot \sqrt{area}$, and the neighbor distance d_2 to a fifth of the path budget. The total number of training points for the SGP n_{SGP} has been set to 1600 points, which determines the grid-filter resolution of $r = \sqrt{\frac{area}{n_{SGP}}}$. And the total number of inducing points n_{IP} to 200 points, fixing an inducing point density $d_{IP} = \frac{n_{IP}}{area}$. Table 4.3 summarizes the configuration parameters of the SGP model used for spatial modelling.

Table 4.4: Scale parameters used for the target areas A1-A4 of the 2D simulation experiments.

Scale Parameter	A1	$\mathbf{A2}$	$\mathbf{A3}$	$\mathbf{A4}$
Area $[m^2]$	472	17480	1655	4105
Path Budget $[m]$	26.0	158.7	48.8	76.9
Maximum neighbor distance d_2 [m]	5.2	31.7	9.8	15.4
Grid-filter resolution $[m]$	0.54	3.31	1.02	1.60
Inducing points density $[m^2]$	0.43	0.01	0.12	0.05

Figure 4.9 shows an overview of the results obtained after replanning in the area A1, using a partial constraint configuration ($\alpha = 1.0$, $\omega = 0.5$, $d_1 = 1.0$). For this simulation we define a path execution distance of 8.0m; this means that when a path is proposed for execution, 8.0m of the total path is simulated and the rest of the path is discarded. This parameter is used only to simulate the sequential replanning process where paths are overwritten before their complete execution. Furthermore, for this simulation in the area A1 we set a stop criterion for the simulation of 300m traveled.

The replanning process starts by executing a short initial path of $\sim 10m$, from this execution we simulate the acquisition of a series of images from the synthetic groundtruth, and produce a raw data set for grid-filtering and spatial modelling using an SGP model and obtain a stochastic prediction of the seagrass distribution (mean and variance) across the target area. It can be noticed that as the robot advances and collects more data, the prediction improves; the prediction represents better the GT, and the high variance areas are located on the meadow boundaries. The first part of the replanning process seems to

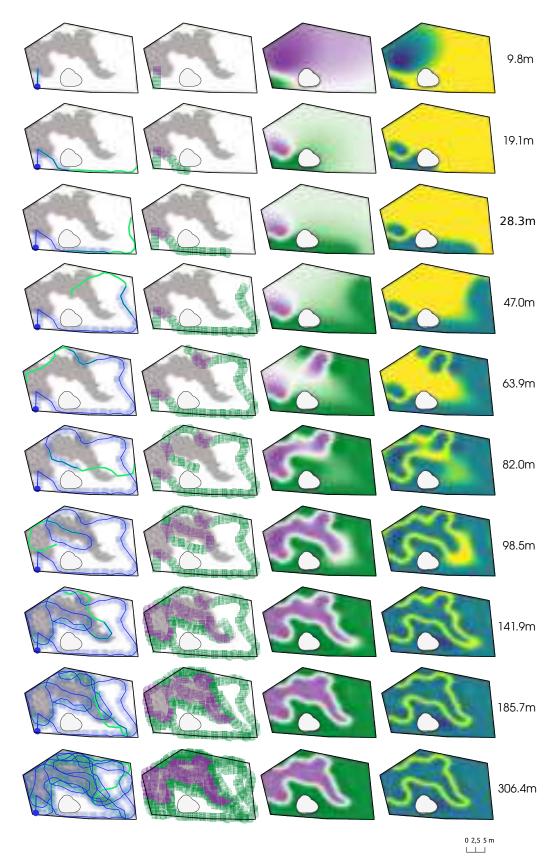


FIGURE 4.9: Results obtained for an adaptive replanning in area A1 $(472m^2)$. From left to right we represent: (1) the path planned (green), the executed path (blue) and the raw data density shaded in blue/green, the simulation groundtruth (white represents positive label), (2) the raw data samples recorded, where the colored dots represent P. oceanica detection, green states for positive and purple for negative, (3) the mean value of the SGP prediction, green for positive predictions and purple for negative, and (4) the variance value of the SGP prediction, where the yellow area represents high values.

be focused on exploring broadly the environment, whilst the last part focuses on visiting the meadow boundaries, exploiting the prediction by collecting more data in high variance that are more susceptible of mapping errors.

Next sections compare the performance of the two DF-MCTS versions previously defined C1 and C2. First with a sampling based strategy proposed by *Viseras et al.* [113] in two target areas A1 and A2, and secondly, with two types of predefined lawn mower paths in two target areas A3 and A4.

Comparison with a sampling-based strategy

In this section we compare the results obtained for the two configurations proposed of the DF-MCTS algorithm with the sampling-based algorithm proposed by *Viseras et al.* [113]. This sampling-based algorithm is based on two steps; (1) a *Search Station* step to localize the highest reachable information spot nearby growing an RRT, and (2) an *Informative Path Planner* step to propose near-optimal path to reach the station spot growing an RRT*, by means of maximizing the path information. Table 4.5 shows the parameters used for these experiments, which are a scale of the original ones, adapted to our application scale.

TABLE 4.5: Configuration parameters for the RRT and RRT* algorithms of the sampling-based strategy used for comparison; given for target areas A1 and A2.

Sampling-based Configuration	A1	A2
Expand distance [m]	0.5	3.0
Near distance $[m]$	0.5	3.0
Budget $[m]$	20	120

Figure 4.10 shows the resulting path and the recorded data density, after performing an adaptive replanning in simulation, in the target areas A1 and A2, for the DF-MCTS strategy using configurations C1 and C2, and for the sampling based strategy.

Moreover, Figure 4.11 shows of the resulting MDE, SDE, AUC ROC and CP obtained during the assessment as the mean value of 12 tests.

Let us analyze the obtained results:

- The constrained configuration C1 of the DF-MCTS results in smoother paths and the highest coverage rate for the two areas under study. And results in a high exploration of the large target area A2, providing the highest MDE reduction.
- In contrast, since the unconstrained configuration C2 of the DF-MCTS do not penalize turns, it is able to focus the navigation in recording high information heterogeneous regions (meadow bounds). This configuration provides the highest DE reduction in A1.
- The sampling-based strategy provides good coverage on boundary regions at expenses of providing non-smooth paths and the lowest MDE reduction.

Comparison with predefined lawn mower pattern

Missions programmed using predefined lawn mower (LM) patterns are widely used to record data from the seabed in a homogeneous way. As stated before, the main limitation of such technique, is that the process is passive in the sense that the gathered data is not used to drive the robot.

Here we compare the use of two types of predefined LM patterns, LM1 and LM2, with the two configurations C1 and C2 of the DF-MCTS strategy. The predefined path type LM1 consists in a sparse LM

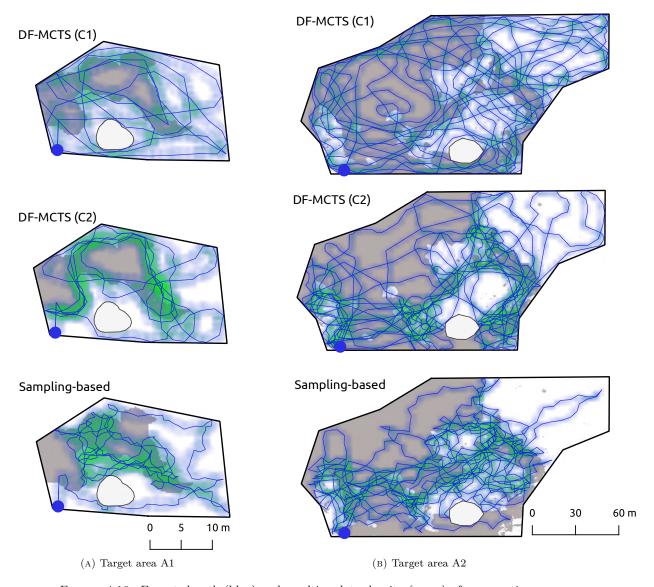


Figure 4.10: Executed path (blue) and resulting data density (green) after executing an adaptive replanning 2D simulation over the target area A1 and A2; using DF-MCTS with configurations C1 and C2, and a sampling-based method.

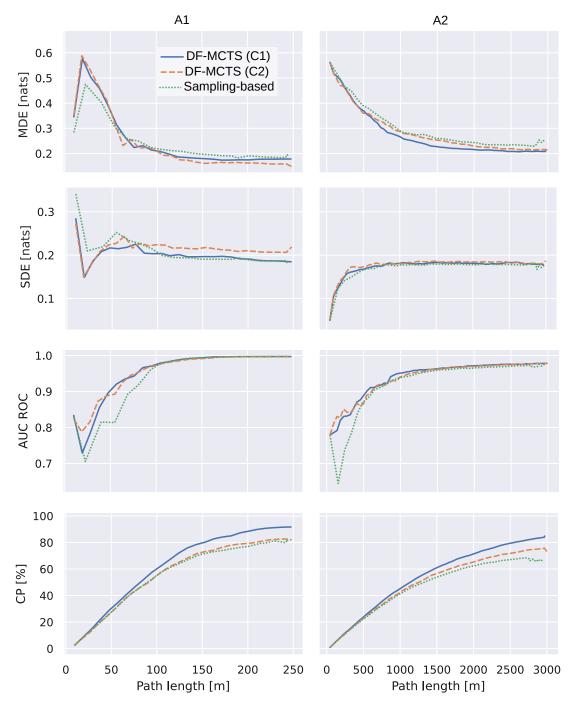


FIGURE 4.11: Results obtained in 2D simulation for the DF-MCTS with configurations C1 and C2 compared with a sampling based strategy in the target areas A1 and A2.

pattern with a lateral distance between transects of 6m, and LM2 consists in a LM pattern with a lateral distance between transects of 2m. LM1 is usually used to get a fast representation of the environment, and LM2 is used to get high data coverage.

Figures 4.12 and 4.13 show the executed paths and the recorded data density obtained in the target areas A3 and A4 respectively. The DF-MCTS algorithm focuses the data recording in the meadow bounds, in locations that present high entropy during the mission execution whilst the data acquisition using LM1 and LM2 is homogeneous.

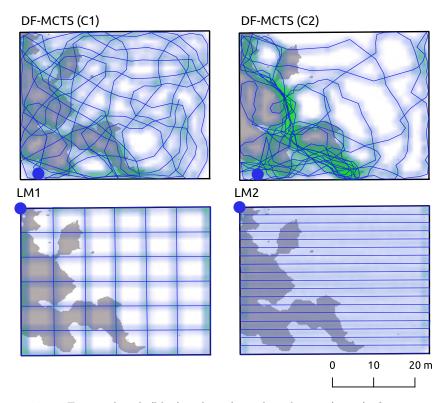


FIGURE 4.12: Executed path (blue) and resulting data density (green) after executing a 2D simulation on target area A3; using DF-MCTS with configurations C1 and C2, and two different preprogrammed LM types; LM1 and LM2.

The mean resulting metrics obtained along the path for 12 tests using the same strategies and configurations are represented in Figure 4.14:

- The highest reduction rate of the MDE is provided by the DF-MCTS tests, and the largest peak on the SDE results in the LM tests, which means that the DF-MCTS methods target precisely the regions to visit in order to quickly reduce the map uncertainty.
- In terms of AUC ROC, the predefined path LM1 provides a slightly faster increase, but converges to a value smaller than the value of LM2 and C1, on both target areas.
- Regarding the coverage, LM2 has a constant growth due to the fixed image overlap that only varies due to the added localization noise. LM1 shows two constant patterns, the first has the highest slope due to that it keeps recording new visited areas, and the second has the lowest slope because it performs the cross track transects of the trajectory, recording data in previously visited spots. In the case of the DF-MCTS strategy, the coverage increases fast at the beginning, while the algorithm focuses on exploring non-visited areas, and reduces the slope slowly depending on the configuration used; C1 results in a high coverage while C2 focuses on recording the heterogeneous data areas.

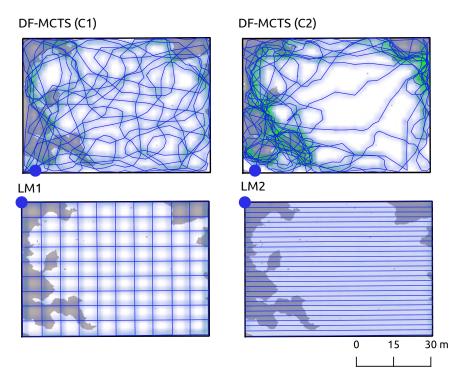


FIGURE 4.13: Executed path (blue) and resulting data density (green) after executing a 2D simulation on target area A4; using DF-MCTS with configurations C1 and C2, and two different preprogrammed LM types; LM1 and LM2.

4.5 Conclusions

This chapter presents a novel design of a decision-time adaptive replanning (DAR) method. Such DAR behavior is coupled with a DF-MCTS strategy for IPP, and joins two advantages of graph-based and sampling-based methods: (a) initializes a node network to set neighbor relations between sampling locations (which reduces computation during online execution), and (b) samples paths in the node network through tree search following a decision-time strategy (that provides near-optimal solutions in an anytime manner).

The proposed methods have been tested and compared with the very relevant strategy proposed by *Viseras et al.* [113] and with a predefined lawn mower pattern mission. The results show that the variable distance between sampling nodes and the penalization of sharp turns in the reward function provide smooth paths that are secure to be followed by the LOSCTE controller. And that the DAR method with DF-MCTS provided the fastest MDE reduction.

The adaptive planning of informative paths has two major contributions to the field of in situ data gathering with AUVs. First, due to the intrinsic mechanism of in situ mapping, there are many sources of uncertainty that can not be easily reduced and affect particular image samples. Having a system that is able to model the uncertainty, to enable resampling of such regions reduces the overall data uncertainty. Second, a big limitation of in situ visual data gathering with AUVs is that the small coverage obtained per image recorded sets a hard bound on the extension that an AUV is able to record for a given time budget. Hence, for large regions that can not be fully covered it is particularly useful to let the robot architecture drive the navigation, targeting the most interesting/informative spots.

This kind of methods have many implications that have not fully covered in this work, but will be considered for further developments. For instance, the DAR method could integrate the execution of

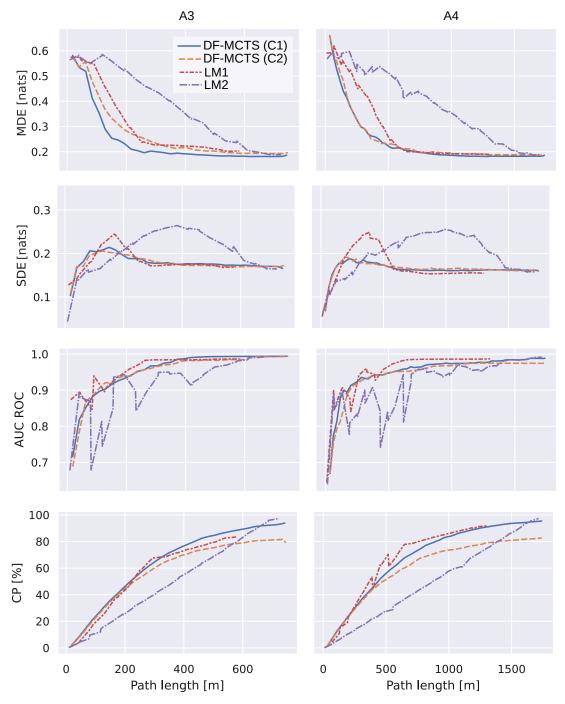


Figure 4.14: Results obtained in 2D simulation for the DF-MCTS planner using configurations C1 and C2 compared with two predefined lawn mower patterns LM1 and LM2 in the target areas A3 and A4.

template paths in high uncertainty regions, such as long sections or small lawn mower surveys. The different physical processes and applications will condition the type of sampling required.

Chapter 5

Autonomous navigation

The evolution from an inspection framework to the data-driven exploration framework proposed in this Thesis involves some important challenges in terms of autonomous navigation. Remember that two of the main objectives regarding the exploration framework, enumerated in Chapter 1, are to maximize (i) the time devoted to data gathering, and (ii) the quality of the recorded data. In terms of navigation, these objectives conduce to the following challenges:

- Long-lasting navigation: A first challenge with regard to the maximization of the time devoted to data gathering is maintaining an accurate localization of the AUV. As data gathering take longer time without surfacing, the localization drift induced by odometric sensors increase. In the particular case of sampling benthic environments, the action of surfacing to receive GPS corrections becomes expensive, since it prevents the AUV of gathering benthic data. Hence, in order to increase the recording time without reducing the quality of the recorded data, the localization payload of the AUV has to integrate a source of absolute position when navigates underwater.
- Absolute positioning during mission time: A second challenge arising from the novel exploration framework is that the absolute position measurements can not be only stored for posterior data geo-localization, in this case the absolute measurements have to be used by the robot architecture to estimate its position during mission time; for data localization and for control. The main motivation behind this specification is that (a) the environment modelling process requires accurate geolocation of the data processed to generate a proper representation, and that (b) the adaptive replanning process requires accurate control to visit certain coordinates.

Given the aforementioned challenges, this chapter focuses on the *in situ* sampling process, introducing the main sources of data uncertainty and describing and assessing an autonomous navigation framework based on the use of an ultra short baseline (USBL) system to enable absolute localization during mission time.

Section 5.1 introduce the main sources of uncertainty caused by the robot navigation during the process of gathering *in situ* images and provides a series of equations that will be used for the assessment of the recorded data. Then, Section 5.2 describes a set of novel features integrated in an autonomous navigation framework based on the use of an USBL system to enable absolute localization of the AUV and attain the requirements imposed by the environment modelling and the adaptive replanning processes. Moreover, Sections 5.3 and 5.4 showcase a set of experiments to assess the quality of the generated data recorded

with the AUV Turbot and Section 5.5 concludes the chapter, providing the main conclusions and the future work directions.

5.1 Data uncertainty

This section introduces the main sources of uncertainty caused by the robot navigation that condition the quality of the *in situ* recorded images. Here we consider the use of down-facing cameras for recording geo-localized images of the seabed, where each geo-localized image is composed by an array of pixels (the image) and its localization (or pose). The main sources of uncertainty can be independently associated with both the accuracy of the image pose and the quality of the recorded images themselves.

The focus of this section is to derive a set of equations to enable the propagation of the robot pose uncertainties into geo-localized image pose uncertainties and robot movement into image pixel blurring. These two elements will define the quality of the recorded data.

Pose and scale

The pose of an image refers to a tuple composed by position coordinates and orientation and the scale refers to the image footprint or extension. Figure 5.1 represents an example of geo-localized image; defined by its horizontal position (x,y) of the top-left corner, footprint (fp_u,fp_v) and orientation ϕ ; with respect to a given coordinate reference frame crs. The inaccuracies related to the pixel locations will be originated by uncertainties in the geo-localized image definition, inexactness on the scale and on the position and the orientation.

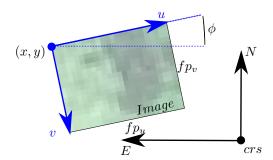


FIGURE 5.1: Geo-localized image, given by its footprint in the u and v directions, its position (x,y) and orientation ϕ with respect a coordinate reference frame crs.

We will consider that the localization uncertainty ϵ of a given image pixel (u,v) is the result of the superposition of the inaccuracies related to the absolute position $\epsilon^P = [\epsilon_u^P, \epsilon_v^P]$, scale $\epsilon^S = [\epsilon_u^S, \epsilon_v^S]$, and orientation $\epsilon^O = [\epsilon_u^O, \epsilon_v^O]$ as expressed by following equations:

$$\epsilon = \|[\epsilon_u, \epsilon_v]\| \tag{5.1}$$

$$\epsilon_u = \epsilon_u^P + \epsilon_u^S + \epsilon_u^O \tag{5.2}$$

$$\epsilon_v = \epsilon_v^P + \epsilon_v^S + \epsilon_v^O \tag{5.3}$$

The absolute position uncertainty ϵ^P can be directly derived from the absolute localization error, and will be constant for all the pixels in the image.

Conversely, scale and orientation uncertainties will vary with pixel location, with pixels further away from the center of the image suffering greater uncertainty. In particular, the scale uncertainty e^{S} depends on the altitude uncertainty $\epsilon^{altitude}$ with the relation given by the following expressions:

$$\epsilon_u^S = d \cdot \cos(\alpha) \cdot \frac{\epsilon^{altitude}}{a} \cdot \frac{fp_u}{2}$$

$$\epsilon_v^S = d \cdot \sin(\alpha) \cdot \frac{\epsilon^{altitude}}{a} \cdot \frac{fp_v}{2}$$
(5.4)

$$\epsilon_v^S = d \cdot \sin(\alpha) \cdot \frac{\epsilon^{altitude}}{a} \cdot \frac{fp_v}{2}$$
(5.5)

Where ϵ_u^S and ϵ_v^S are the perpendicular components of the image plane, d and alpha indicate the polar coordinates of a pixel (u,v) in the image, a represent the navigation altitude, fp_u and fp_v represent respectively the image footprint in the horizontal and vertical components. The image footprint fp = $[fp_u, fp_v]$ is obtained from the navigation altitude a and the camera field of view $fov = [fov_u, fov_v]$ as:

$$fp = 2 \cdot a \cdot \tan(\frac{1}{2} \cdot fov) \tag{5.6}$$

And the polar coordinates d and alpha of each image pixel are computed as follows;

$$d_{(u,v)} = ||u - w/2, v - h/2|| \cdot ps$$

$$\alpha_{(u,v)} = \arctan \frac{v - h/2}{u - w/2}$$

$$\forall (u,v) \in [w,h]$$

, where r = [w,h] are the horizontal and vertical components of the image resolution and $ps = [fp_u, fp_v]$ is the pixel size, given by Equation 5.7.

$$ps = \frac{fp}{r} \tag{5.7}$$

Finally, orientation uncertainty ϵ^{O} depends on the independent imprecision resulting from each orientation component e^{roll} , e^{pitch} and e^{yaw} . We will assume that each orientation component contributes to each horizontal component of the orientation uncertainty as follows:

$$\epsilon_u^O = \epsilon_u^{O_{roll}} + \epsilon_u^{O_{puch}} + \epsilon_u^{O_{yaw}} = a \cdot \epsilon^{roll} + d \cdot \cos(\epsilon^{yaw})$$
 (5.8)

$$\epsilon_v^O = \epsilon_v^{O_{pot}} + \epsilon_v^{O_{pitch}} + \epsilon_v^{O_{yaw}} = a \cdot \epsilon^{pitch} + d \cdot \sin(\epsilon^{yaw})$$
 (5.9)

where the contribution of the roll and pitch are zero respectively for the vertical y and horizontal x uncertainty components.

Hence, the uncertainties related to the pixel-wise locations rely on the accuracies related to the localization, altitude and orientation estimations of the AUV. The sensor payload and the filtering techniques applied will determine the localization performance of the recorded data.

In general, AUVs that use absolute positioning systems based on acoustic transponders such as USBL or long baseline (LBL) set a bound on the positioning error [143]. However, the low rate and low precision of such systems introduce positioning inaccuracies. In contrast, systems that relay their positioning to dead-reckoning may accumulate a large drifting error over time that deteriorates the accuracy of the recorded data. In terms of orientation, magnetometer measurements are prone to suffer biases on the heading estimation depending on the position of the device with respect to the moving robot, and to the environment. The uncertainty on the orientation depends on the quality of the compass used for navigation and in the calibration procedure followed.

Spatial resolution

Let us now discuss the relation between robot movement to image blurring. Such blurring depends on the lighting conditions and the recording speed, and can be expressed with the proportional relation given by the following equation.

$$B = s \cdot E \tag{5.10}$$

Generating small blurring distance B is a matter of enabling low exposure times E and controlling smoothly the robot motion at low sampling speeds s. This work focuses the analysis on the robot motion, thus the exposure times required for recording are left out from the analysis and considered as a fixed constraint given by the environment and the robot characteristics.

The blurring refers to the displacement of the features recorded in the image during the exposure time required to record it. The next equations have been derived to provide the resulting blurring in the horizontal b_u and vertical b_v directions for each pixel caused by the linear (v_x, v_y) and angular $(\omega_x, \omega_y, \omega_z)$ speeds caused by the robot motion in the camera's reference frame. The pixels are given using their polar coordinates d and α .

$$b_u = (\omega_y \cdot a + v_x + \omega_z \cdot d \cdot \sin(\alpha)) \cdot E \tag{5.11}$$

$$b_v = (\omega_x \cdot a + v_u + \omega_z \cdot d \cdot \cos(\alpha)) \cdot E \tag{5.12}$$

The resulting spatial resolution components $sr = [sr_u, sr_v]$ of an image will be the pixel-wise maximum between the blurring components $[b_u, b_v]$ and the pixel size $ps = [ps_u, ps_v]$. Then, we propose to obtain the spatial resolution as pixel-wise maximum between the spatial resolution components.

Let us define the spatial resolution of an image as the maximum value between the ps, the maximum value in b_u and the maximum value in b_v . This will provide an upper bound for the spatial resolution specification.

The equations derived in this section are used for the data assessment of Section 5.3.

5.2 Navigation framework

Reducing the aforementioned sources of uncertainty require some advanced features in terms of navigation. The AUV has to be able to produce a precise localization estimate, navigate close to the seabed, and to navigate steadily at low speeds.

Hovering AUVs have the common characteristic of being able to maintain low advancing speeds while controlling the altitude using vertical thrusters and producing adequate localization estimates. This feature enables getting very close to the seabed even in rough terrains. Figure 5.2 provide some examples of hovering robots adequate for *in situ* benthic sampling using the methods exposed in this work.







(B) Sirius AUV, image from Schmidt Ocean Institute.

FIGURE 5.2: Hovering AUVs.

In particular, the methods presented in this chapter have been integrated in the AUV Turbot, a lightweight hovering AUV equipped with a stereo camera rig and a localization payload that includes the generation of USBL measurements. The next sections provide a description about the robot (regarding the operation modes, acoustic communications and positioning using an USBL system) and about the localization filters used for pose estimation and USBL outlier filtering.

5.2.1 AUV Turbot

The AUV Turbot is a lightweight torpedo-shaped AUV with hovering capabilities and low draught suited for very shallow waters. The robot is based on a *Sparus II* [141] unit, equipped with three degrees of maneuverability: surge, heave and heading.

The Turbot's navigation and communication payload consists of: a (i) DVL to obtain linear and angular speeds and altitude, an (ii) echo sounder (ES) to generate altitude measurements, a (iii) compass for heading, an (iv) IMU to measure accelerations and angular speeds, a (v) pressure sensor to obtain high-frequency depth measurements, a (vi) GPS to be geo-referenced during surface navigation, an (vii) acoustic modem with USBL for communication and absolute localization, and a (viii) stereo camera rig with a 16cm baseline for image recording. Table 5.1 specifies the device model and output of the sensing and communication devices installed on the AUV.

Device	Model	Output
DVL	Teledyne ExplorerDVL	Linear/angular speeds and altitude
ES	Imagenex 852	Altitude
IMU/Compass	ADIS16488	Orientation, linear accelerations and angular speeds
Vacuum gauge	Honeywell 19	Pressure
GPS	G-top FGPMMOPA6H	Absolute position
Cameras	Allied Vision Manta G-283	BayerRG8 images
Modem	Evologics S2CR 18/34	Acoustic communications and positioning from USBL
RF antenna	Ubiquity Bullet M2	Surface communications

Table 5.1: AUV Turbot payload

The AUV Turbot is equipped with two onboard computers. The main computer is a PC/104, used for serial connection with sensors and to execute the processes related to localization, control and safety. The secondary computer, is an Intel NUC i7 that integrates a dedicated interface to connect to the stereo vision system and executes the high-level processes that are non-essential for autonomous navigation.

Figure 5.3 shows a diagram that represents the two computers and the connection with the navigation and communication devices. The figure shows in blue the elements connected via RS232, in red the

elements connected to the main Ethernet LAN, and in green the secondary Ethernet network dedicated to the stereo camera. When the robot navigates on surface the main LAN can be accessed from the CS computer through the 2.4GHz antenna.

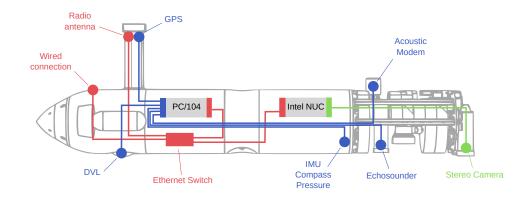


FIGURE 5.3: Navigation and communication payload of the AUV Turbot with two computers onboard. In blue the RS232 connections, in red the Ethernet connections over the primary LAN, and in green a secondary Ethernet LAN for dedicated interface with the stereo cameras.

The stereo camera rig is build using a pair of *Allied Vision Manta G-283* cameras. The cameras are synchronized and configured with the higher native resolution, which is 1936×1458 pixels, and provide raw images in *BayerRG8* format.

A series of processes are executed to generate a visual altitude from the stereo image pairs. (i) Decimation of the raw image pairs to 242×182 pixels, an eighth of the raw images resolution, (ii) rectify both images using the calibration parameters obtained through a process of a previous stereo camera calibration, (iii) compute the stereo pair disparity, (iv) project the image pixels to three-dimensional coordinates, (v) compute the mean image depth.

Finally, for the navigation framework proposed using the AUV Turbot, the supporting control station is equipped with (i) a computer, (ii) a 2.4GHz radio antenna *Ubiquity Bullet M2*, (iii) two GPS *Emlid Reach RS* (a moving module that integrates RTK corrections on top of the USBL mounting support, and a static one located on ground that provides the RTK corrections), and (iv) an USBL *Evologics S2CR 18/34* transceiver with integrated compass. All the devices on the control station are communicated via Ethernet TCP/IP.

Operation modes

Figure 5.4 represents the two operation modes considered: surface and underwater. Both modes allow communication with a supporting vessel, which should be equipped with a control station (CS) computer, a radio frequency antenna compatible with the AUV, a GPS and an USBL. Furthermore, both modes are used automatically depending on the state of the robot, and can switch during mission time without issues.

The CS launches the packages related to the guidance, teleoperation, USBL communications and positioning, which are based on ROS [146]. Both, the AUV and the CS have an active *core* in order to share ROS-based communications in each particular LAN. While the robot is on surface a radio connection is established to join both LANs. The *multimaster FKIE* package [147] is used to share part of the ROS

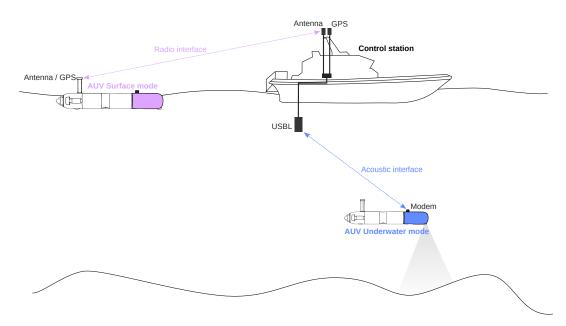


FIGURE 5.4: Operation modes; surface mode with radio interface for communication and GPS positioning, and underwater mode with acoustic interface for communication and absolute positioning.

Table 5.2: Communication and absolute position depending on the operation mode

Operation mode	Communication	Abs. Position
Surface	Radio frequency	GPS
Underwater	Acoustic	USBL

environment between cores. This includes accessing to the images feed and the navigation status from the robot, and executing services and guidance commands from the CS.

While the AUV is underwater the TCP/IP connection between robot and CS lays off. Then an acoustic interface allows a low bandwidth transmission of key information using an $Evologics\ S2CR\ 18-34kHz\ Modem/USBL$. Such transmission includes: (a) sending the AUV navigation status to the CS for monitoring, (b) calling several query services to enable or disable different AUV functionalities and safety actions, and (c) sending the AUV position estimated by the USBL at the CS back to the AUV to use it for pose estimation [143].

In brief, in surface mode the communication is based on RF and the absolute positioning is based on GPS. In contrast, in underwater mode the communication is based on acoustics and the absolute positioning on USBL, as shown in Table 5.2.

Acoustic communications and positioning

This section provides further details on the implemented USBL system for acoustic communication and positioning. Figure 5.5 illustrates the processes related to the USBL system. The figure shows in purple the processes contained in the software architecture based on ROS [146] and in blue the low level processes contained in the physical layer of the acoustic modem and USBL head. Notice that the processes on the left side are executed on the CS and the processes on the right are executed on the AUV. During underwater navigation the acoustic interface (in blue exchanges data between both systems).

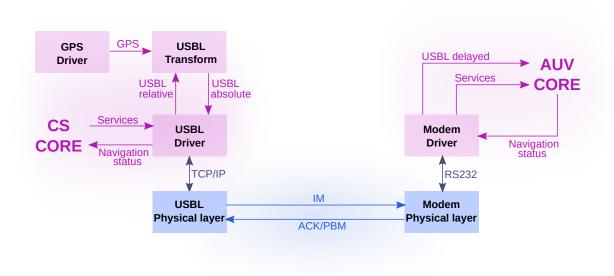


FIGURE 5.5: USBL system for acoustic communication and positioning.

Physical layers

The modem and the USBL physical layers communicate acoustically between them using a S2C (Sweep Spread Carrier) protocol. This protocol is provided by the manufacturer, and permits the transmission of different types of messages between devices. The three type of messages that have been used for enabling acoustic communication between devices are: the instantaneous message (IM), the acknowledgment (ACK) and the piggyback message (PBM). Both are defined on the S2C (Sweep Spread Carrier) protocol implemented on the physical layer of both devices provided by the manufacturer.

- IM message. These messages are used to send short data sentences (max 64bytes) and are transmitted as soon as possible when the transmission command is received by the acoustic device. They can be configured to return a delivery acknowledgment.
- ACK message. These messages are sent automatically by the physical layer of the device when an IM is received with request for acknowledgment.
- **PBM message.** These messages can carry short data sentences as the IM. However, they are transmitted only at the reception of an IM message, they are attached to the IM acknowledgment. A buffer is used to store outgoing PBM messages, the most recent message on the buffer is sent when an IM with request for acknowledgment is received.

Both USBL and modem drivers use a RS232 and TCP/IP protocol respectively to command the transmission of IM messages and to update the PBM buffer. The key idea is that the IMs are used to ping the modem and when the modem acknowledges the reception of the ping by returning an ACK message the USBL is able to compute the position of the modem.

The ping that is sent by the USBL might be empty or may contain a data transmission. Such transmission, from USBL to modem, can be either (a) a query service or (b) an USBL position, and always requests for an acknowledgment. Furthermore, depending on the frequency set to transmit the navigation status, from modem to USBL, the ACK can be sent alone or followed by a PBM.

Acoustic driver

The Algorithms 4 and 5 describe the iterative execution of the protocol in both devices. Such algorithms are implemented in the USBL and modem drivers, which are respectively executed in the CS and AUV cores as represented in Figure 5.5. The USBL iteratively (1) sends an IM to the modem. This message can contain a position measurement and an acoustic service from the CS, (2) waits for an acknowledgment, (3) and publishes the AUV status if received. The modem iteratively (1) reads the serial port for incoming communications, (2) calls the queried services, (3) publishes the received positions, and (4) updates the AUV status on the PBM buffer.

```
Algorithm 4: USBL()
send\_im()
ack \leftarrow wait\_ack()
if status \in ack then
| publish\_status()
end
```

```
Algorithm 5: Modem()

im \leftarrow read\_im()

for srv \in im do

|call\_service()

end

if position \in im then

|publish\_position()

end

update\_status()
```

The implemented driver is accessible in an online repository [148] and is based on the previous work of *De Carolis* [149].

Acoustic positioning

The acoustic positioning measure is generated in the *USBL Physical layer*, it generates (a) a relative orientation measured by trilateration of the incoming acoustic signal using the five transponders contained in the USBL head, and (b) a distance measured between devices using the round trip transmission time from USBL to modem and back to the USBL. The result is a measurement of the modem position relative to the USBL reference frame in North East Down (NED) coordinates. Such relative measure is then transformed to global coordinates in the *USBL transform* process using the GPS position of the USBL.

5.2.2 Localization filters

The goal of the localization filters is to centralize state observations and generate a navigation status information containing a 6-DOF pose and altitude estimations. The first goal is attained by gathering the different state observations coming from sensor drivers, acoustic drivers and image processing outputs such the visual altitude. The second goal is attained by feeding such state observations to a set of localization filters to generate diverse pose and altitude estimations. The processes involved are summarized in Figure 5.6. We use three instances of a third party EKF implementation of *Moore et al.* [150]. One instance

to estimate absolute localization for control, a second instance to estimate motion (odometry) used for USBL outlier detection, and a third instance to estimate altitude.

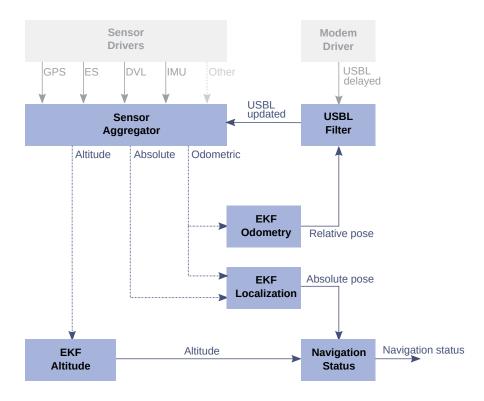


FIGURE 5.6: Localization layer.

- Sensor aggregator. Centralizes and monitors the state observations. It subscribes to the different sensor measurements that provide altitude or pose measurements, such as GPS, ES, DVL, IMU, compass, pressure sensor, USBL measurements, etc. Applies message type transformations and sets the measurement covariances and monitors the performance of the different measurements. This process initializes the EKF Odometry and the EKF Localization estimations only when absolute position measurements from GPS or USBL are available. The filters are initialized using the mean position of the last ten position measurements.
- EKF Odometry. Provides an accurate measure of the robot motion used to filter and update the USBL correction received acoustically from the remote USBL head. This filter inputs all the odometric measurements and the absolute measurements that provide nearly continuous values that prevent the output from jumping. Here we consider all the navigation payload measurements with exception to the GPS and USBL, which are low-frequency and noisy. Thus, the filter includes: (i) the linear speeds provided by the DVL, (ii) the angular speeds provided by the gyro included in the IMU, (iii) the orientations provided by the IMU and compass, and (iv) the depth position calculated from the pressure sensor measurements.
- **EKF Localization.** This filter provides a pose estimation that includes the filtering of all the measurements available on the 6-DOF. This includes the measurements used by the *EKF Odometry* process and the absolute position measurements provided by the GPS and the USBL when the vehicle is on surface or underwater respectively.
- EKF Altitude. The goal of this filter is to provide a continuous altitude estimation by fusing all the altitude measurements in an EKF. The filter includes the use of: (i) the altitude measurement

and the vertical speed provided by a DVL, (ii) the altitudes provided by an echosounder, and (iii) the visual altitude provided by stereo processing. The main reason of using such filter is to integrate sensor redundancy on the altitude estimation and overcome the issues related to low altitude navigation in challenging terrains such as soft seagrass meadows or rough rocky terrains that may complicate bottom tracking.

• USBL filter. This process pursues two important goals: (i) updating the USBL measurements received from the USBL head on the AUV, and (ii) removing outliers. Both goals are carried out using the odometric pose estimation provided by the EKF Odometry process. The update is performed by composing the USBL measurement with the odometric estimated motion between the time that the USBL measurement was generated and the actual time, and is important in order to correct the distance navigated by the robot during the time interval in between the generation of the USBL measurement in the CS and the integration on the localization filter of the AUV. The outlier removal is performed using a simple distance filter that depends on the distance traveled by the AUV during the time elapsed between the time the last USBL measurement was obtained t_{-1} and the time of the current USBL measurement t_{-0} . The expression used is the following:

$$usbl_{t_0}^{t_1} < \alpha + \beta \cdot odom_{t_0}^{t_1} \tag{5.13}$$

, where $usbl_{t_0}^{t_1}$ represents the distance between the USBL measurements taken at t_0 and t_1 , $odom_{t_0}^{t_1}$ represents the distance between the odometric estimations provided by the *EKF Odometry* filter at t_0 and t_1 , α represents a value near the actual USBL position covariance, and β represents a value near the possible odometry drift.

Table 5.3 summarizes the input measures of the EKF Localization filter used by the AUV Turbot. The EKF Odometry filter use the same inputs except of the GPS and USBL measurements.

	P	ositic	n	Ori	entat	tion	Lin	ear v	elocity	An	gular	velocity	Acc	elera	tion
Device	X	Y	Z	α	β	γ	\dot{X}	\dot{Y}	\dot{Z}	$\dot{\alpha}$	$\dot{\beta}$	$\dot{\gamma}$	\ddot{X}	\ddot{Y}	\ddot{Z}
DVL							√	√							
IMU				√	√					√	√	√			
Compass						√									
Depth			√												
GPS	√	√													
USBL	√	√													

Table 5.3: Measurements used for the EKF Localization filter.

The initialization of both EKFs is set by the sensor aggregator and can be initialized using either raw GPS or delayed USBL corrections.

5.3 Experiments

The principal objective of the experiments performed was to assess the *in situ* data gathering capabilities using the proposed navigation framework on a particular hovering vehicle, the AUV Turbot. Four data gathering missions have been executed in Andratx (39.544055, 2.378246). This section describes the set of experiments performed as well as the process for generating an absolute positioning groundtruth.

5.3.1 Missions

All the experiments were carried out in underwater mode with acoustic positioning and communications in a very shallow area of 12m depth maximum. A tethered connection with a bridge buoy was used in order to provide full access to the robot in case of deficient acoustic connection due to the shallowness of the region. The maximum exposure time for the stereo image recording was set to E = 0.01s.

Figure 5.7 provides an overview of the four datasets recorded. The figure shows, for each experiment, the localization of the USBL head, the output of the EKF localization filter, and the USBL measurements in absolute coordinates integrated in the filter. Notice that for the four experiments the USBL head was not kept on a static location but it was moved around attached to the supporting boat.

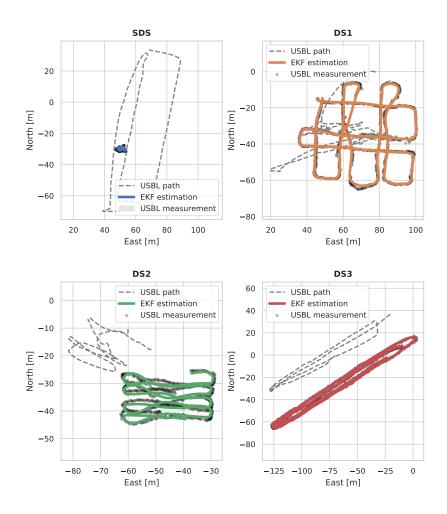


FIGURE 5.7: In situ image recording experiments.

Three datasets (DS1,DS2,DS3) where recorded while the AUV was performing three different types of pre-defined lawn mower trajectories. For each of them the AUV was set with a maximum surge speed for the section maneuvers of s = 0.3m/s. The target altitude was set to 3m for datasets DS1 and DS2, and 3.5m for DS3.

The dataset SDS was generated by keeping the AUV hovering in a constant depth and position setting while the USBL head was moved around to generate absolute position measurements. This experiment was designed to be able to generate an absolute localization groundtruth (GT) to be used for evaluation.

5.3.2 Absolute localization groundtruth

The GT for the static dataset SDS has been obtained by trilateration, using the acoustic range distance provided by the USBL and its location for several sampling positions.

Figure 5.8 represents the trilateration of the landmark L (the AUV location) using a series of range distances d_i and ranging positions P_i with $i \in [1,n]$. The mean square error (MSE) mse_{opt} given in Equation 5.14 is used as cost function for an optimization process; where d_i represents the distance between the USBL head and the AUV and P_i represents the USBL position provided by a GPS as the ranging positions P, given at the same time instants than d.

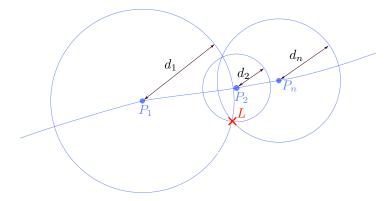


FIGURE 5.8: Trilateration example of the landmark L using a series of range distances d_i and ranging positions P_i with $i \in [1,n]$.

$$mse_{opt} = \frac{\sum [d_i - dist(L, P_i)]^2}{N}$$
(5.14)

The optimization process is executed in three dimensions (x,y,z), where $L = \{l_N, l_E, L_D\}$ and $P = \{p_N, p_E, 0\}$. Instead of using the full set of measurements generated during the test, we used the subset that was closer to the estimated AUV location with a distance threshold $d_{th} = 30m$. This subset contained 87 measurements, and resulted in the AUV location L = (-30.32, 50.85) with an MSE of 0.14m.

Figure 5.9 represents (1) the recorded path followed with the USBL head during the static experiment, (2) the USBL measurements generated, and (3) the trilaterated AUV position. Figure 5.10 shows the ranging locations and distances used for the trilateration, represented with blue dots and blue circles respectively.

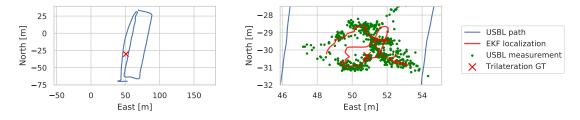


FIGURE 5.9: Path followed with the USBL head to generate USBL measurements while the AUV is maintained static, and triangulated AUV position to be used as localization groundtruth. On the left a large scale view, on the right a small scale view focused on the USBL measurements area.

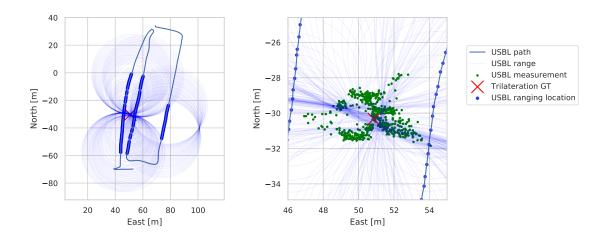


FIGURE 5.10: Ranging locations and distances used for the trilateration, represented with blue dots and blue circles. On the left a large scale view, on the right a small scale view focused on the USBL measurements area.

5.4 Results

This section presents the results obtained from the experiments described in Section 5.3, and provides an evaluation of the navigation stack onboard of the AUV Turbot and a quality assessment of the geolocalized images gathered during experiments.

5.4.1 General navigation

The analysis of the AUV navigation performance starts with the evaluation of the surge and sway speeds, Figure 5.11 shows the speeds estimated by the EKF odometry filter.

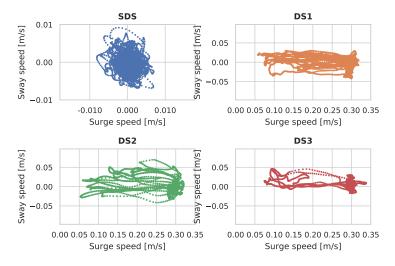


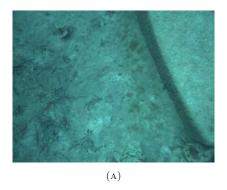
FIGURE 5.11: Surge and sway speeds.

For the SDS the speeds are near zero, and follow a normal distribution centered in zero with a standard deviation of 0.002m/s for both the surge and sway components. Considering that the DVL is the only sensor contributing to the estimation of the linear speeds one may assume that such standard deviation is composed by errors on the DVL measurements and errors on the static position control. By visual

Table 5.4: Mean value and standard deviation of the sway speed estimation for each dataset.

Dataset	Mean $[m/s]$	Std $[m/s]$
DS1	-0.0022	0.0132
DS2	0.0042	0.0170
DS3	0.0055	0.0083

inspection of the initial and final images recorded in the SDS dataset, the displacement results, in fact, very small.



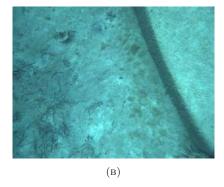


FIGURE 5.12: First and last images of the static dataset SDS.

In the case of datasets DS1-3 the surge speed is maintained in the interval between the target surge speed $s_{max} = 0.3m/s$ and a minimum surge speed $s_{min} = 0.05m/s$ at the extreme of the transects. The sway speed does not present correlation with the surge speed. Table 5.4 provides the resulting mean value and standard deviation of the sway speed estimation for datasets DS1-3. The three datasets present a small bias that is different between them and may be result from extrinsic factors such as currents. Moreover, they present a standard deviation 4x - 8x times bigger than for dataset SDS, probably caused also by currents and by higher noise on the DVL measurements due to more challenging conditions.

With regard to the angular speeds, Figure 5.13 represents the absolute value of the angular speeds for datasets DS1-3. DS1 presents a steady navigation, in which the roll takes maximum values around 0.025rad/s and the maximum yaw speed is around 0.08rad/s, which does not seem to affect the roll component. In contrast, DS2-3 present several peaks on the roll speed around 0.05rad/s that seem to be correlated with the yaw speed.

Figure 5.14 represents the relation between altitude and depth of each dataset. During the SDS the AUV maintained its position at approximately 8m depth and 1.7m altitude. In the case of the datasets DS1-3 the AUV navigated different bathymetric profiles that resulted in a variation of the AUV recording depth while controlling a target altitude; 3m and 3.5m altitude for datasets DS1-2 and DS3 respectively.

The difference between the target and the resulting altitudes might be caused by several factors related to altitude measurements and to the altitude control. First, depending on the benthic environment the different sensors used for altitude estimation could provide erratic behavior s. Figure 5.15 shows the range distances provided by the DVL, the echosounder and the visual altimeter together with the estimated altitude fusing the aforementioned ranges in an EKF filter.

The reason of fusing such set of measurements in an EKF is due to prior issues with the DVL stopping providing measurements when navigating close to rough rocky environments or to very dense seagrass meadows. The experiments shown here do not include any of these extreme environments and, as a

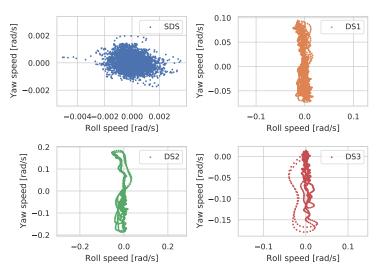


FIGURE 5.13: Angular speeds.

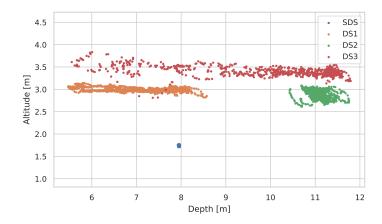


Figure 5.14: Depth and altitude.

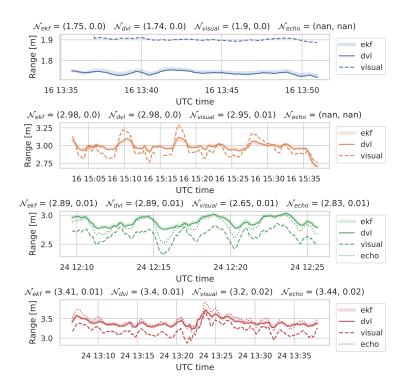


FIGURE 5.15: Altitude measurements from the DVL, echosounder, visual altimeter and filtered EKF altitude estimation. From top to bottom SDS, DS1, DS2 and DS3.

Table 5.5: Altitude error distribution of the EKF altitude estimation, the visual altimeter ranges and echosounder measurements considering the DVL measurements a groundtruth.

Dataset	EKF	Visual altitude	Echosounder
SDS	$\mathcal{N}_{ekf} = (0.01, 0.0021)$	$\mathcal{N}_{visual} = (0.16, 0.0076)$	-
DS1	$\mathcal{N}_{ekf} = (0.0, 0.0103)$	$\mathcal{N}_{visual} = (-0.03, 0.0736)$	-
DS2	$\mathcal{N}_{ekf} = (0.0, 0.0059)$	$\mathcal{N}_{visual} = (-0.23, 0.0755)$	$\mathcal{N}_{echo} = (-0.05, 0.0567)$
DS3	$\mathcal{N}_{ekf} = (0.01, 0.0091)$	$\mathcal{N}_{visual} = (-0.2, 0.0622)$	$\mathcal{N}_{echo} = (0.04, 0.0669)$

consequence, the DVL provides very smooth altitude measurements that might be even considered an altitude groundtruth.

Dataset SDS shows a clear bias of the visual altimeter, that might be due to an imprecise stereo calibration considering that the static transforms between sensors are correctly configured. In contrast, DS1 shows low bias on the error and a bigger standard deviation. DS2 and DS3 result in the highest altitude noise metrics.

Table 5.5 provides the error distribution of the EKF altitude estimation, the visual altimeter ranges and echosounder readings considering the DVL measurements a groundtruth. The EKF altitude estimation results in a maximum bias of 0.01m and standard deviation of 0.01m. The echosounder provides range measurements with a maximum bias of 0.05m and standard deviation of 0.06m. And the visual altimeter ranges result in a maximum bias of 0.23m and standard deviation of 0.08m. From these results we can assume a bound on the altitude estimation error, given by the visual altitude, of $mse = bias^2 + var = 0.059m^2$.

Regarding the altitude control, the tracking error of the altitude reference might be due to a bias on the vehicle pitch. Figure 5.16 represents the roll and pitch of the AUV. The roll seems to be biased by almost -2.0° for all the datasets, and the pitch seems to behave differently depending on the dataset. The pitch on the SDS dataset results in a positive bias of 3.0° and very low variance caused, probably, by an unbalanced buoyancy. In contrast, datasets DS1-3 result in a negative pitch bias with very high variance. The negative pitch bias may be induced by the tethered buoy used for supervision during the validation experiments. Since the buoy is attached to the stern of the vehicle, it pulls up from the stern resulting in a change of the pitch. The impact of this issue increases as the navigation depth augments since the pulling force results in a larger vertical component.

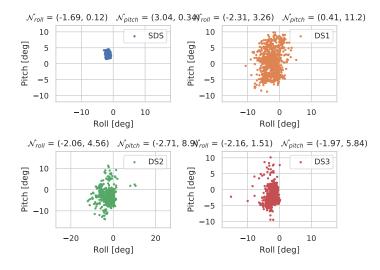


FIGURE 5.16: Roll and pitch.

The bias on the AUV pitch impacts the altitude control. The negative pitch induces a vertical force component caused by the surge thrusts that is not fully compensated with the heave thruster and results in a stationary altitude bias that depends on the pitch sign and magnitude.

During the four experiments the USBL system described in Section 5.2.1 was used. Such system provided absolute localization and acoustic communications to monitor the AUV. Figure 5.17 shows the frequency of the absolute position measurements generated by the USBL and the frequency of the monitoring communications used to transmit the robot status from AUV to CS. The figure shows the results obtained for the datasets SDS, DS1 and DS3. DS2 is not showed because the recorded data to compute the frequencies from such dataset was corrupted.

The absolute position measures achieved maximum frequencies of 0.8Hz, 0.83Hz and 0.67Hz and mean frequencies of 0.66Hz, 0.67Hz and 0.48Hz respectively for the SDS, DS1 and DS3.

5.4.2 Absolute localization

The absolute localization accuracy is computed using the groundtruth location L of the AUV derived in Section 5.3, for dataset SDS. Figure 5.18 shows of the absolute error of localization estimated by the EKF Localization and EKF Odometry filters, and the absolute error of the USBL north-east measurements.

The resulting bias, variance and mse of the USBL measurements and EKF altitude estimation is showed in Table 5.6, where the bias and the variance are computed as follows,

$$bias = \mathbb{E}[\vec{X}] - L \tag{5.15}$$

$$var = \mathbb{E}[(L - \vec{X})^2] \tag{5.16}$$

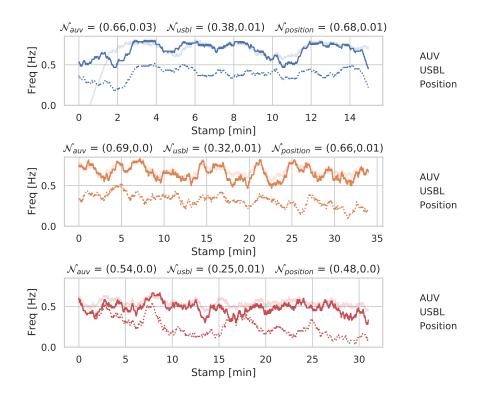


FIGURE 5.17: Acoustic channel frequencies. From top to bottom SDS, DS1 and DS3.

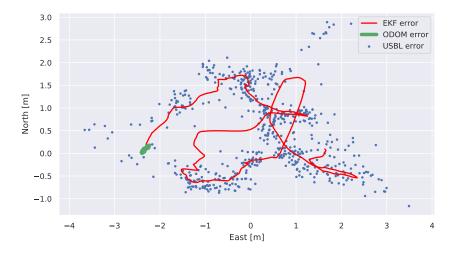


Figure 5.18: Absolute localization error of the EKF Localization and EKF Odometry estimation filters and the USBL north-east measurements.

	Bias $[m]$	Variance $[m^2]$	$MSE [m^2]$
USBL measurements	0.60	1.58	1.94
EKF localization	0.61	1.21	1.58
EKF odometry	2.33	0.0029	5.43

Table 5.6: Pose estimation results

where L is the AUV static location groundtruth obtained by trilateration, and \vec{X} represents a vector containing either the updated USBL measurements or the EKF localization estimation.

For this experiment we consider the same initialization for both EKF filters, using the first USBL measurement. The results show that (1) the EKF localization is biased by the USBL measurements, they have similar error bias, and results in lower bias than the odometric estimation, (2) the integration of the USBL measurements increases significantly the variance of the EKF, and (3) the integration of the USBL measurements in the EKF provides lower MSE than the odometric filter estimation and that the USBL measurements itself.

Nevertheless, these results are valid for measuring distances between the USBL and modem devices closer than 70m. The results represented in Figure 5.19, suggest that there is a linear correlation between the measuring error and the distance. The slope of such regression is m = 0.01391, which means that the localization error would increase a 1.4% of the incremented measuring distance.

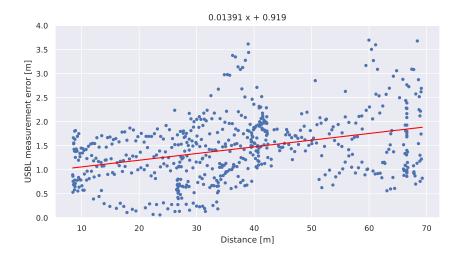


FIGURE 5.19: USBL measurements error with respect to the relative distance between USBL and modem. The blue dots represent the USBL error, the red line represents the linear regression 0.01391x + 0.919.

5.4.3 Data quality assessment

The quality analysis of the data recorded by the AUV is carried out considering the sources of uncertainty enumerated in Section 5.1, here we aim to quantify the localization uncertainty of a given pixel, as well as the resulting spatial resolution of the images.

Pose and scale

Here we provide the results after performing a propagation of the navigation uncertainties resulting from Section 5.4.1, and summarized in Table 5.7.

		Bias [m]	Variance $[m^2]$	$MSE [m^2]$
Position	ϵ^{ekf}	0.61	1.21	1.58
Altitude	$\epsilon^{altitude}$	0.23	0.006	0.059
Roll	ϵ^{roll}	2.0	2.25	6.25
Pitch	ϵ^{pitch}	3.0	2.25	11.25
Yaw	ϵ^{yaw}	1.0	4.0	5.00

Table 5.7: Uncertainty results from navigation analysis.

Figure 5.20 shows the pixel-wise localization uncertainty in terms of MSE. Notice that the pixel uncertainty caused by the position is constant for all the pixels, and equal to the error resulting from the EKF localization filter $mse^{ekf}=1.58m^2$, and that the uncertainty caused by the scale is very small due to the accuracy of the altitude estimation, in despite of considering the highest error margin caused by the visual altitude estimation with a bias of 0.23m. Moreover, the uncertainty caused by the orientation results also in a very low contribution, reaching a maximum uncertainty value in terms of MSE of $0.35m^2$. Finally, the resulting pixel uncertainty reaches a maximum MSE of $2.57m^2$, which is 1.6m in RMSE. This means that the localization error of a feature contained in an in situ sampled image will have a mean localization error bounded on 1.6m, considering the use the navigation framework described in Section 5.2 with USBL corrections in a distance range closer than 70m.

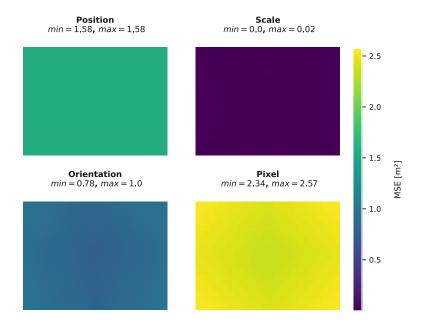


FIGURE 5.20: Pixel MSE distribution considering uncertainty from the position, scale, and orientation.

Image blurring

Figure 5.21 shows a representation of the resulting image blurring that would be obtained from recording an image from an altitude of 3.5m, a surge speed of 0.35m/s and a yaw angular speed of 0.05rad/s. This image blurring has been simulated using the equations introduced in Section 5.1, and the results are provided following a NED reference frame convention. Notice how the blur caused by the angular speed increases the left side blurring. In contrast, the right side blurring is smaller due to the superposition with the blur caused by the surge component.

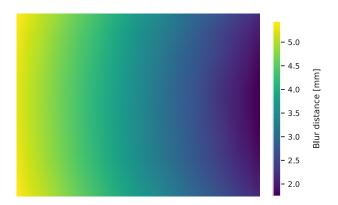


FIGURE 5.21: Blur distance example for an image recorded with an altitude of 3.5m, a surge speed of 0.35m/s and a yaw angular speed of 0.05rad/s.

Moreover, Figure 5.22 shows three examples of images (IMG1-3) recorded by the AUV with different navigation conditions and same maximum exposure time set to 0.01ms. The first row shows the original image, and next rows represent successive zooms of the regions represented with red rectangles. Since the navigation conditions of such image examples are known, their blurring distances at each pixel location can be computed using the equations described in Section 5.1. Figure 5.23 shows the estimated blurring for images IMG1-3 respectively. The first and second images represent the horizontal and vertical blurring components, and the third represent the module.

IMG1 was recorded with very slow surge and angular rotation, and resulted in very low blurring distance. The horizontal component, caused by sway and roll rotation is near zero, and the most of the image blurring is caused by the surge or the pitch rotation. Reaching a maximum value of 1.54mm. Considering that such blur distance is smaller than the pixel size resulting from the navigation altitude $ps_{IMG1} = 1.7mm$, the image blurring caused by the robot motion shouldn't be noticed. However, inspecting the last image amplification of IMG1 from Figure 5.22 and assuming a sharp edge of the recorded cavity, we can notice a blurring distance of 7px = 12mm.

IMG2 was recorded with almost the reference surge speed of 0.3m/s and residual yaw rotation, the AUV was traveling along a mission transect. The burring distance behavior is similar to the one obtained for IMG1, in this case with a larger blurring module of 2.98mm due to a larger vertical component caused by the surge speed. By visual inspection of the third zoom of IMG2 on Figure 5.22 and assuming sharp edges of the long seagrass leaves, an approximate distance blurring of 3px = 5.7mm and 6px = 11.4mm can be inferred respectively for the horizontal and vertical blurring components.

IMG3 shows a different blurring distance distribution. In this case the rotation yaw and the sway speed reach higher values, which superpose with the surge component and causes the larger blurring distance in the bottom right corner of the image. The estimated maximum blurring distance in such corner is 6.5mm, which results in an image blurring that is difficult to quantify by visual inspection using an environmental feature such as a sharp seagrass leave.

The difference between the blurring distance computed from the navigation parameters, and the actual blurring resulting in the images, might be due to numerous sources that have not been covered in this work. Such blurring might be caused, for instance, by the water column turbidity, unclean lenses, or even due to imprecision on the camera focus. Such conditions should be further studied and considered using specific tests and instrumentation.

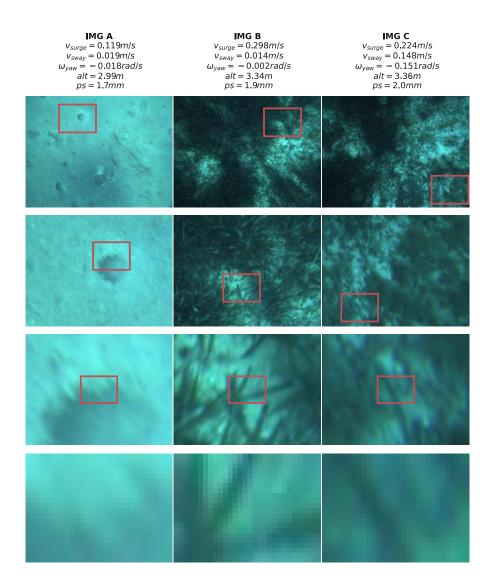


Figure 5.22: Images recorded during the field-tests. The columns represent a different image (IMG1-3) and the rows represent sequential zoom of the red rectangles

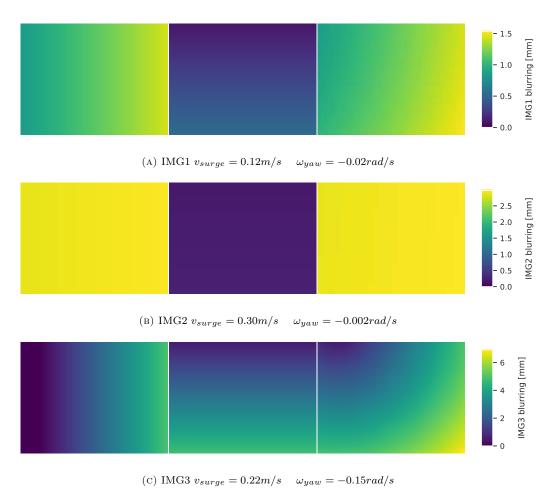


Figure 5.23: Estimated blurring distances for the example images IMG1-3. From left to right the horizontal and vertical components and the module.

Spatial resolution

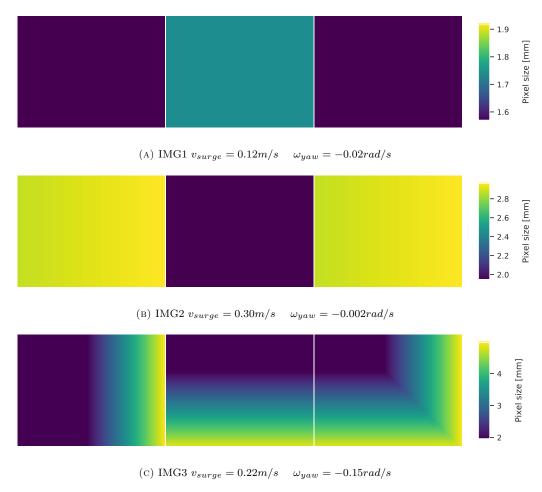


FIGURE 5.24: Resulting pixel size for the example images IMG1-3. From left to right the horizontal and vertical components and the pixel-wise maximum.

The spatial resolution of the images recorded can be considered as the maximum between the computed blurring distance caused by the robot navigation and the pixel size determined by the camera specifications and the recording altitude. Figure 5.24 shows the resulting pixel size for the images IMG1-3, in terms of horizontal and vertical components (two initial columns) and pixel-wise maximum. The pixel-wise maximum metric has been used to describe the pixel resolution in order to provide the higher bound of the *square* pixel resolution.

Another interesting aspect of this analysis is to evaluate the correlation between the navigation settings and the estimated pixel resolution. Figure 5.25 shows such correlation, where the blurring is described using the *mean*, std, min and max metrics for each image, and the navigation is described using: the surge and sway speeds $(v_x \text{ and } v_y)$; the rotation in roll, pitch and yaw $(\omega_x, \omega_y \text{ and } \omega_z)$; and the altitude a. From such navigation descriptors, the surge speed provide the larger correlation with the mean, std, and max pixel size. And the altitude provides the largest correlation with respect to the min pixel size.

The problem of this analysis is that, as shown in Figure 5.26, the navigation descriptors considered are not independent. First notice that the surge v_x and the rotation in yaw ω_z are negatively correlated. This shows two contrasting image recording manners; low v_x and high ω_z at the beginning and end of the transects, and high v_x and low ω_z during the central part of the transects. Secondly, notice that the

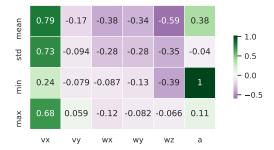


FIGURE 5.25: Correlation between the burring and the navigation. For each image, the blurring is described using the *mean*, std, min and max metrics for each image, and the navigation is described using: the surge and sway speeds $(v_x$ and $v_y)$; the rotation in roll, pitch and yaw $(\omega_x, \omega_y \text{ and } \omega_z)$; and the altitude a.

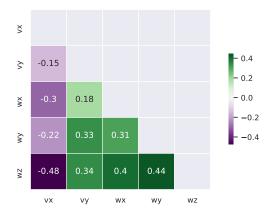


FIGURE 5.26: Correlation between the navigation descriptors: the surge and sway speeds $(v_x \text{ and } v_y)$; the rotation in roll, pitch and yaw $(\omega_x, \omega_y \text{ and } \omega_z)$; and the altitude a.

sway v_y and the rotation in roll ω_x and pitch ω_y are positively correlated with the rotation in yaw ω_z . When the vehicle is set to turn the v_y , ω_x and ω_y increase.

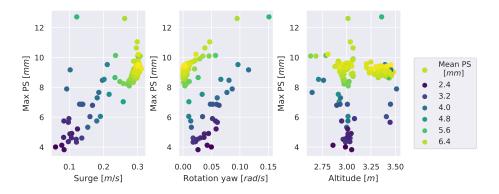


FIGURE 5.27: Correlation between surge v_x , yaw rotation ω_z , altitude a and the mean pixel size PS.

Only v_x , ω_z and a can be controlled and configured as navigation settings, and the rest of the descriptors are an undesired product of the underactuated robot navigation. Figure 5.27 provides more detail on the correlations between surge v_x , yaw rotation ω_z , altitude a, the maximum pixel size PS and the mean pixel size PS. The correlation of the maximum and mean PS with the surge v_x can be clearly distinguished.

Following the definition for obtaining the spatial resolution of an image provided in Section 5.1, as the maximum value between the pixel size, the maximum blurring value in caused by the horizontal recording

displacements, and the maximum blurring value in caused by the vertical recording displacements. The maximum value of the spatial resolution obtained for all the images recorded during the experiments results in 12.71mm. These results provide the upper bound of the spatial resolution that can be obtained with the proposed navigation framework executed onboard of the AUV Turbot and with a maximum exposure time of E = 0.01s.

5.5 Discussion

In situ image recording is a very challenging activity that relays on having accurate and smooth navigation. Minimizing the different sources of uncertainty that condition the quality of the recorded data is a matter of controlling the recording speed depending on the light conditions, and recording close to the seabed. Hovering AUVs are a very suited tool for this task. This chapter has introduced the novel software architecture developed to allow benthic exploration. Introducing the main sources of data uncertainty caused by the robot navigation; pointing out the pre-existent modules that have been used and the novel developed modules such as the navigation, modelling and replanning modules; and providing a detailed description and testing of the navigation module. Defining a surface and an underwater navigation modes, with different absolute localization measurement sources and communication interfaces, an USBL system executed during the underwater navigation mode to provide acoustic localization and communications, and a set of localization filters used provide absolute pose and altitude estimations as well as to remove outliers on the USBL measurements.

In terms of navigation, a key result is that using the proposed navigation module, the uncertainty on the absolute localization provided by the EKF localization filter is correlated with the range distance between USBL and modem, as the distance increases the bias and variance of the measures also increase, resulting in an unbounded absolute localization error. In order to bound such error the range distance between USBL and modem has to be limited. In case of limiting the USBL measuring range to 70m, the localization error has resulted in an MSE of $1.58m^2$.

Then, in terms of data quality, using the uncertainty propagation equations derived in Section 5.1 and the experimental data with the USBL measuring range to limited to 70m, the results show that the localization error of the recorded images was bounded to $2.56m^2$ MSE. And that the pixel resolution of the proposed in situ sampling system has a maximum spatial resolution of 12.71mm. That means that, using the navigation settings proposed, in the worst case considered, the non-blurred pixel size of an image will be inferior to 12.71mm and will have a localization accuracy with RMSE of $\sqrt[2]{2.56} = 1.6m$.

Future work related to the topics studied in this chapter should mostly involve further field-testing, including the following aspects. First, factors that have a direct impact on the image quality and not covered in this study, like light attenuation, scatter and absorption of the water column, should be assessed. Second, as the acoustic communications and positioning have been validated, new field-testing should be extended to cover deeper regions without the tethered buoy to prevent this monitoring system to hamper the robot navigation. Finally, close range navigation on top of different seabed morphologies, with more challenging conditions for the altitude measurement sensors would be desirable.

This chapter has shown that the spatial resolution achieved by means of recording *in situ* visual data is very fine. It is a few orders of magnitude better than other methods used for BHM described in Chapter 2, and could be easily improved by means of using more steady platforms and improved strobe systems. However, the main concern that we face for improving the quality of the recorded data is that the localization uncertainty is far from being that accurate and can not be easily improved.

The integration of a visual simultaneous localization and mapping (VSLAM) algorithm in the navigation module is a possible way to decrease the localization uncertainty. However, the visual odometry and loop closing detection on benthic environments covered by PO seagrass has shown to be very challenging due to the movement of the leaves and to the lack of distinctive features. Another methodology that could be further researched to reduce the localization uncertainty would be the active position of the USBL to reduce its intrinsic inaccuracies related to the bearing estimation. This issue could be tackled by developing a cooperative navigation framework with an ASV carrying the USBL head.

Chapter 6

Data-driven exploration

Autonomous exploration is being an important research area for the last decades. The main purpose of robotic exploration is to gather data in areas that humans can not reach, such as, in deep waters, in the space, or in missions that involve unsafe or hazardous environments or actions. Such exploration could be performed by means of an operator controlling remotely a robot system. However, this relies on having a proper bidirectional communication between the operator and the robot, for instance to transmit robot images and operator commands. The problem is that in many scenarios such communication is not possible, or is very limited, and the robot has to be equipped with some degrees of autonomy.

AUVs have shown a big improvement of their autonomous capabilities during last years. The localization problem has been tackled in many areas [83], different strategies of sensor fusion [143] and simultaneous localization and mapping (SLAM) have shown to provide very good performance [151, 152]. Additionally, CNNs (Convolutional neural networks) have shown to be very effective for online image segmentation [153], which provides robots with a semantic understanding of the environment. And the computing capabilities, energy efficiency and control of autonomous robots are also improving. However, data gathering missions with autonomous robots are normally limited to the use of preprogrammed paths and their performance is supported only on a reliable localization and control.

This chapter describes the last step towards the development of an autonomous exploration architecture pursued in this Thesis, based on the semantic modelling, replanning and autonomous navigation methods described in last chapters. Figure 6.1 represents the proposed scheme.



FIGURE 6.1: Data-driven exploration scheme. The modelling and replanning threads close a feedback loop on top of the autonomous navigation framework that enable the adaptive exploration of a target area given a probabilistic environment model of the environment that is trained during mission time.

Data generated during the robot navigation is processed to build a model of the environment, which is used by the replanning thread to generate informative paths to be followed by the robot. This iterative loop would continue until a given threshold based on an uncertainty metric, budget time or distance are reached.

The structure of the chapter is as follows. First, Section 6.1 describes the proposed architecture providing a brief description about the requirements and the building blocks used for the implementation. Second, Section 6.2 describes a set of simulation experiments performed to validate the performance of the AVIG architecture prior field deployment and compares the execution of the architecture using a DAR planner, a SBSRE* planner and a predefined lawn mower pattern path. Third, Section 6.3 presents two sets of field experiments which are key to evaluate the full architecture executed on board of the AUV Turbot, on different scenarios and conditions. Finally, Section 6.4 concludes the chapter.

6.1 Adaptive visual information gathering (AVIG)

This section introduces a novel adaptive visual information gathering (AVIG) architecture for data-driven exploration of benthic environments using AUVs. The objective of the AVIG architecture is to adapt dynamically the robot navigation using the visual information gathered online.

The principal goal of this architecture is to evolve from a classic *in situ* data gathering framework in which an AUV is deployed to perform a predefined mission path, to an adaptive framework in which the AUV is able to replan the mission path reliant on the semantic understanding of the environment. In such a way we enable the exploration of unknown regions (1) decreasing the reliance on an operator for mission planning, (2) maximizing the data recording time, and (3) maximizing the informativeness of the recorded data.

Remember from Chapter 1 that, due to the underwater communication limitations, the logistics to enable a remote operator to adapt the mission plans to the most recent recorded data would require moving the robot to surface and even recovery for data transmission. In order to speed up exploration the robot should be able to model the environment online and decide the best path to execute to accomplish the data acquisition objective.

6.1.1 Requirements

The execution of the proposed architecture aims to maximize quality and extension of the recorded data. Targeting long-lasting unsupervised navigation, with communication limitations and no (or uncertain) prior environmental knowledge. The following requirements are set:

- Navigation at low altitudes with absolute positioning. The AUV should be able to navigate safely close to the seabed, using absolute position measurements for localization and being able to perform untethered with acoustic supervision in order to provide a high maneuverability.
- Bounded computation times to allow online execution. In terms of semantic habitat modelling, the computations should be efficient, with bounded computation times and should capture accurately the uncertainty of the process under study.
- Smooth navigation and continuous exploration. In terms of replanning, the AUV should move smoothly without stopping, exploring the environment and targeting dense image recording in high information areas.

Moreover, a general design requirement for the architecture is to be modular in order to enable the adaptation of the different modules to changing target applications or environment conditions. Hence, whereas the methods described in this work target the collection of spatial distribution data of P. occanica seagrass for seabed coverage estimation, they could be adapted or substituted for methods that pursue other goals.

For instance, the models involved in the modelling module would vary depending on the process under study. First, the image processing pipeline should be adapted to the type of physical process, by retraining the network or even changing the network model. Second, the SGP contained in the spatial modelling process should be configured accordingly to the type of semantic data used and the properties of the process in terms of spatial distribution.

Moreover, depending on the target application, the planning could require a different balance between exploration and exploitation. Different sampling requirements could be integrated in the planner. For instance, instead of generating smooth informative paths, a given application could require the sequential execution of straight paths or even small and dense lawn mower patterns, located in informative spots.

6.1.2 Methods

The AVIG architecture integrates some of the environment modelling, replanning and autonomous navigation methods described in the Chapters 3, 4 and 5. In particular, it executes the six methods illustrated in Figure 6.2 in six parallel threads.

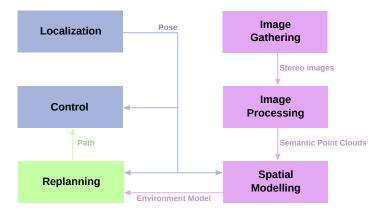


FIGURE 6.2: Relation between threads of the adaptive visual information gathering (AVIG) architecture.

The localization thread involves the execution of the processes related to the localization filters included in the navigation framework discussed in Chapter 5. It inputs all the sensor measurements to estimate the robot pose with a frequency of 25Hz. The image gathering and image processing threads do not include any variation with respect to the methods described in Chapter 3. They involve the execution of the stereo camera drivers with a sufficient frame rate to enable the recording of successive overlapping images and the setting of the maximum exposure time depending on the reference surge speed and the image segmentation using a modified version of the VGG16-FCN8 network as well as the projection to three-dimensional coordinates using the stereo disparity.

The next paragraphs provide specific detail about the control, spatial modelling and replanning threads. These threads contain methods that have not been described previously or that have been updated for the integration.

Control

The control thread is inherited from the COLA2 architecture [154]. It governs the thrusters power in order to follow a desired mission path, and it is composed by three main processes: (i) a *captain* parses the incoming mission paths to low level maneuvers that are sent to (ii) a *pilot* that provides pose and velocity reference commands to a controller using the estimated AUV pose and low level maneuver to

be followed, and (iii) a controller transforms the reference commands to thruster setpoints by following a double PID implementation. The paths commanded to the control thread are described in terms of section maneuvers. Such maneuvers are defined by (a) an initial position, (b) a final position, (c) speed, and (d) tolerance. The pilot adjust the velocity commands using a Line of Sight with Cross Tracking Error (LOSCTE) strategy [155], which is a well suited strategy for tracking mission paths, considering a reference setting of maximum and minimum surge speed values. This thread is also set to be executed with a frequency of 25Hz.

Spatial modelling

The spatial modelling thread has the objective of filtering the semantic point cloud data and use it for training a probabilistic regression. Such regression, based on an SGP, will be used on to compute the recorded data uncertainty and get an information metric for further exploration.

The processes included in this thread are based on the methods described in Section 3.2 of Chapter 3 but upgraded to improve the online performance. In particular, they have been integrated in the AVIG architecture with a set of novel updates to optimize the data filtering stage. Such data filtering stage has the objective of reducing the size of the dataset used for GP training and allow online execution with a bounded training time. For that purpose, it performs three main tasks: (i) efficiently collect the processed data in a raw data dataset, (ii) generate a downsampled dataset when queried, and (iii) compute the density of the raw data dataset when queried. The first objective is attained by the image processing thread that continuously filters each segmented point cloud using a voxel filter with a default resolution of 0.1m. The second is attained by downsampling the raw data dataset using a grid filter with a given cell size (by configuration); such downsampled dataset is used to train the SGP model. The third is attained by building a k-d tree [156] using the raw data dataset and a minimum leaf size of 1.0m to compute the density of the collected raw data. Such raw data density quantifies the data recorded in a particular location and is used by the replanner to compute the utility metric of sampling that particular spot.

In regard to the grid filter, it is important to pay attention to two aspects: (a) the cell size, and (b) the type of filter. The cell size will condition the spatial resolution of the spatial model, and the type of filter will determine the type of content and distribution of semantic labels, for instance:

- Mean grid filtering: Consists in returning, for each particular cell, a new data point located at the center of the cell with a semantic label equal to the mean value of all the data points contained in such cell.
- Median grid filtering: Each grid cell retains the data point with the median label value.

Whilst the use of a median filter keeps an actual data point it is more computation expensive than using a mean value.

The predictions generated by the SGP will have a different meaning depending on the type of grid filtering performed. In case of using a mean filter, it represents the mean seagrass coverage in an area equal to the grid filtering cell size. In contrast, in case of using a median filter, the labels used for training represent the probability of seagrass presence in such punctual location.

Replanning

In terms of adaptive replanning the AVIG architecture includes the exact methods described in Chapter 4, that have been adapted to work in a ROS-based software architecture. This includes the execution of the novel DAR method for decision-time adaptive replanning to generate successive informative paths,

without stopping, considering the newest information obtained from the environment model and the DF-MCTS method to solve the IPP problem. The DF-MCTS method has been designed to explore the state-space in a depth-first fashion, provide solution paths of a given length in an *anytime* manner, and reward smooth paths for field realization with non-holonomic robots.

6.2 Simulation experiments

This section describes a set of simulation experiments that have been executed in order to validate the performance of the proposed architecture and compare with other standard and state-of-the-art methods. The objective of these tests is to assess the image processing, spatial modelling and replanning threads before field deployment. As well as comparing the execution of the proposed architecture using different planning methods. The next subsections provide specific details regarding the simulation environment, the comparison methods and the configurations.

6.2.1 Environment

The simulations have been performed in the ROS-based simulator *StoneFish* [157] using the AUV Turbot dynamics and software architecture. Figure 6.3 shows the simulated AUV Turbot surveying a synthetic *P. oceanica* environment.



FIGURE 6.3: Screenshot containing a AUV Sparus II in the StoneFish environment

Such environment is a recreation of the environment found in Illetes, a beach spot located in Palma Bay represented in Figure 6.4. Besides, Figure 6.5a shows the hand-labeled ground truth GT image, it is a black/white image built from an original aerial picture of an area in Palma Bay. The simulated seabed shown in Figure 6.5b is formed by adding texture (*P. oceanica* and sand) on this aforementioned original GT image, which means that there is no error between the black/white GT and the texture image that occupies the simulated seabed.

6.2.2 Configurations

Comparison methods

The main objective of these tests is to validate the execution of the AVIG architecture including the DAR method and compare the performance if we use a sampling based strategy (SBSRE*) or a predefined

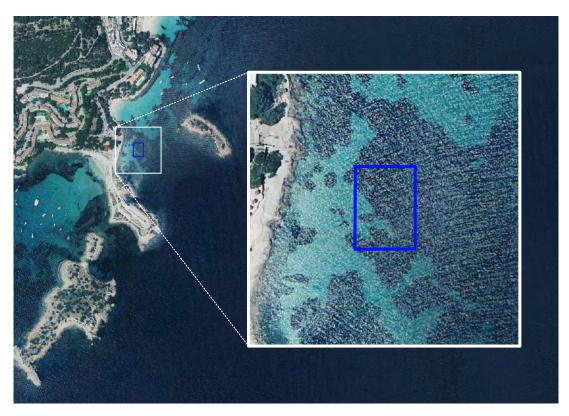


FIGURE 6.4: Target area of the *Illetes* region used for the simulation experiments on top of an aerial image of the region from the *Instituto Geodésico Nacional* (PNOA 2018 campaign).





(a) Seabed hand-labeled ground truth used to build the (b) Synthetic seabed image containing $P.\ oceanica$ and synthetic simulation environment.

Figure 6.5: Synthetic simulation environment.

lawn mower (LM) pattern path instead of using the DAR method on the planning thread. Let us define the configuration of the different planning methods used for simulation.

DAR method

The configuration used for the DAR method is based on the those tested on Chapter 4. Table 6.1 enumerates such parameters.

Parameter Value Planning time 10sInformation function DEAlpha (Utility) 1.0 Omega (Reward) 1.0 Discount factor 0.8 Learning rate 0.9 Rollouts number 32 Epsilon 0.01 Number of nodes 20,000 Neighbor distance d_1 1.5mNeighbor distance d_2 8.5

Table 6.1: Configuration of the DAR method in simulation.

In this case we chose to reduce the planning time from 15s to 10s in order to increase adaptability; since the decision tree is maintained between consecutive planning iterations this time reduction should not have direct impact in the quality of the replanning paths. Moreover, the alpha, omega and neighbor distance d_1 parameters which control respectively the weight of the recorded data density on the utility function and the weight smoothness of the planned paths have been set to an intermediate value from the configurations tested that is expected to provide a proper balance.

33.8m

Budget \mathcal{B}

Additionally, in order to generalize to different target areas, in this case we propose to set by default the distance budget \mathcal{B} for the rollouts to the quarter of the target area perimeter and the neighbor distance d_2 to a quarter of \mathcal{B} .

Sampling-based strategy (SBSRE*)

In order to compare the proposed planning method with a state-of-the-art strategy we propose to use a modified version of the SBSRE sampling strategy proposed by *Viseras et al.* [113]. The original strategy executes sequentially (1) an update of the SGP model hyperparameters using the recorded data, (2) a search station (SS) process based on an RRT algorithm where a path is proposed to reach the highest utility location (station) within a given distance budget, (3) an IPP process based in an RRT* algorithm which tries to get an optimized path to the previously selected station location, and (4) execution of the highest utility path (SS or IPP). In order to introduce the SBSRE* method into the AVIG architecture we take the SGP model update—i.e. step (1)—out of the loop. This process will now be executed on the environment modelling thread in order to provide a more fluent behavior and only stop the robot at the end of the execution of a path for SS and IPP.

Moreover, we used the same planning time (5s for SS and 10s for IPP) and information function (DE) proposed originally and scaled the distance budget, expand distance and path resolution proportionally to the square root of the target region area. The modified parameters are detailed in Table 6.2.

Table 6.2: SBSRE* scaled parameters for the simulation experiment.

	Original	Modified
Area $[m^2]$	18	1,114
Budget $[m]$	2.0	15.7
Path resolution $[m]$	0.1	0.8
Expand distance $[m]$	0.1	0.8

Predefined lawn mower path (LM)

We programmed a lawn mower path (LM) to cover the target area with a distance of 6 meters between transects. Such type of pattern is extensively used to get a partial representation within a fixed time budget. The main difference of this type of predefined paths is that they require very low speeds at the end and beginning of the transects, and uniform sampling speeds along the transects in order to sample homogeneously the seabed.

The next section provides more detail on the configuration of the navigation speeds to be set for the different strategies.

Sampling speeds

Due to the different sampling typologies, the three comparison methods were configured to track the mission paths with different speed configurations. Such speed configurations are used by the LOSCTE controller to set the velocity requests to the AUV controller. The speed configurations used are given in Table 6.3.

Table 6.3: Configuration of the LOSCTE controller used for navigation

LOSCTE configuration	DAR	SBSRE*	$\mathbf{L}\mathbf{M}$
Minimum speed $[m/s]$	0.2	0.2	0.05
Maximum speed $[m/s]$	0.5	0.3	0.2
Lookahead distance $[m]$	2.0	2.0	5.0
Speed transition distance $[m]$	3.0	3.0	3.0
Velocity ratio	0.1	0.1	0.1

The criteria followed to configure such reference speeds is based on the conclusions of Chapter 5. In particular, that chapter showed that the major impact on the resulting pixel size of the recorded images was the surge speed. In order to be able to compare between methods we establish a desired sampling speed of 0.2m/s. The provided LOSCTE speed configurations take into account such desired sampling speed as well as the path tracking constraints of the three methods.

The main difference between the LM with respect to the DAR and SBSRE* strategies is that with the former the AUV is conditioned to have a homogeneous sampling along the sections, while with the latter two methods the sampling is focused on the section edges. Such section edges represent the sampling points considered by both planners while computing the path's utilities. Hence, for the LM the sampling speed has to be maintained along the sections and is configured as the maximum speed. In contrast, for the two adaptive strategies the desired sampling speed has been set at the edges and is configured as the minimum speed. In this case the AUV is allowed to reach higher speeds along the sections.

Moreover, LM and SBSRE* have some other restrictions in order to allow a proper path following. First, the minimum speed of the LM is set to a low value of 0.05m/s to ensure a proper tracking at the extreme of the transects. Second, the maximum speed of the SBSRE* is set to 0.3m/s, a value lower to the maximum speed of the DAR which is set to 0.5m/s for two reasons: (1) the DAR method provides

paths smoother than SBSRE*, which are less prone to sharp turns, and (2) considering that the SBSRE* provides paths with distance between sample locations of 0.8m (expand distance parameter), and that the speed transition distance is fixed to 3.0m, when going through a section in normal conditions (low cross track error) the SBSRE* wouldn't have enough distance to reach the maximum speed, in contrast the DAR provides sufficient distance between sampling points to benefit from such maximum speed.

A key difference in the LOSCTE configuration when using DAR or a LM pattern is that, the LM requires very low acceleration module in order to be able to follow precisely the 90° turns at the extreme of the sections. DAR paths are already optimized to produce soft (smooth) turns and maintaining the vehicle speed is prioritized with respect to the path following precision when performing sharp edges. The path following precision becomes less important since the adaptive behavior will induce revisiting such spot in case of high interest/information, and the vehicle acceleration becomes more important since the section maneuvers tend to be shorter than in the LM case.

Environment perception

Since the objective of the simulation experiments was to validate the execution of the AVIG architecture and compare the performance with other planning methods, the different simulation experiments have been executed using the same configurations for the image processing and modelling threads. The next paragraphs describe such configurations.

First, regarding the image processing it is important to mention that the CNN was not retrained using images gathered from the simulated environment. Instead, we directly used the high resolution output of the network–i.e. M1 of Chapter 3–which provided sufficiently accurate results with some degree of uncertainty that was well suited to build a model and guide the replanning.

Second, the configuration used for the environment modeling is shown in Table 6.4. In addition to the configuration parameters used in Chapter 3 for these experiments we propose to fix the kernel length scale, as well as computing the sampling resolution and the induction points density from the target area extension, considering a maximum number of samples N_S and induction points N_{IP} as described in Remark 1. This resulted in a sampling resolution of 0.75m and an induction points density of $0.18m^{-2}$.

Table 6.4: Modelling thread configuration.

Model	SGPMC
Kernel	Matérn 32
Kernel length scale $[m]$	30
Likelihood	Beta
Likelihood scale	0.5
Optimizer	Scipy
Iterations	Scipy 4,000
N_S	2,000
N_{IP}	200

6.2.3 Results

This section provides the results obtained from the simulation experiments performed. Analyzing (a) the quality of the generated semantic data used for modelling, (b) the resulting surge speeds and replanning paths for the three tested methods, and (c) the performance of the DAR strategy.

Semantic data generation

In terms Figure 6.6 shows an example of the images gathered during simulation and the resulting segmentation using the high resolution output of the CNN. The segmentation results are very good on the boundaries and sand regions.



FIGURE 6.6: Sample images recorded in simulation and their corresponding high resolution segmentation image.

However, even when using the high resolution output of the CNN, some artifacts appear in some areas covered with synthetic *P. oceanica*. In any case, the intrinsic characteristics of an SGP for environment learning from semantic data points makes it a good fit for modelling a physical process under noisy input data conditions; heterogeneous data regions will produce high variance values on the SGP prediction.

Moreover, using DAR we aim to visit, or revisit, high variance (uncertainty) regions, leveraging exploration vs. exploitation. With exploration, we refer to navigate on high variance non-visited regions; with exploitation, we refer to navigate on high variance visited regions. Hence, corrupted data is handled effectively by means of the same mathematical and stochastic background that forms the system, firstly by the SGP regression and secondly by the planner. As a consequence, the process was continued without retraining the CNN.

Furthermore, in these simulations the pipeline for producing the samples' depth was modified to use a simulated depth camera instead of the disparity of the AUV stereo images. We generated the segmented point clouds, combining the point cloud obtained from the depth camera and the segmented image. Figure 6.7 shows a representation of the raw data generated in simulation; the semantic point clouds are published at a 0.290Hz frequency, and are completely flat due to the flat environment used and the null noise in the simulated depth image.

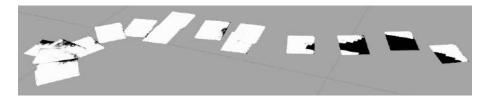


FIGURE 6.7: Sequential spatial data generated in simulation; white pixels represent areas covered with *P. oceanica*, whereas black pixels represent background areas.

Environment modelling

Figure 6.8 shows the resulting modelling times obtained from an execution of the AVIG framework using the DAR planning strategy. Such modelling time include the sampling time and the learning of the SGP hyperparameters. Considering that the latter is the higher contributor and that it is proportional to the number of training samples and inducing points, the modelling time is proportional to the data coverage and not to the gathered data. Hence, the modelling time will converge as the data coverage converges, using the maximum number of samples and number of inducing points.

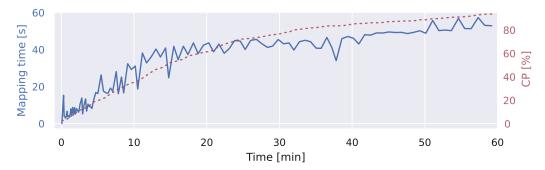


FIGURE 6.8: Resulting modelling time and coverage percent (CP) obtained during the AVIG execution in simulation using DAR.

In this case, the modelling time seems to converge around 55s. The fact that, at the beginning of the exploration the modelling frequency is higher, allowing fast adaptation of the IPP planner to the changing environment model, is a good consequence of the behavior described above. Since at the last part of the exploration the map is partially known, and the high information areas are already located, a low frequency of the modelling thread can be tolerated. However, even at the end of the exploration we require a bound on the modelling time in order to integrate newer data into the belief within a reachable time.

Benchmark

Figure 6.9 shows the resulting speeds for the three tests during an intermediate interval of 7min of the simulation tests. While the DAR oscillates between 0.1m/s and 0.5m/s, SBSRE* reaches 0.2m/s when a mission is enabled and stops when the mission is disabled and LM reaches sequentially 0.2m/s while going through a transect and lowers to 0.05m/s at the end of the transects.

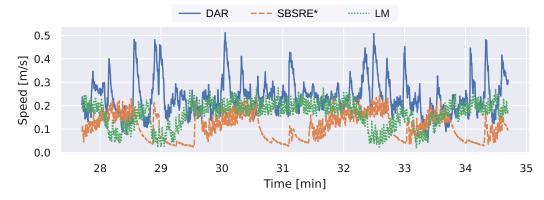


FIGURE 6.9: Resulting AUV speed in the forward direction for the tree strategies tested in a 7min time interval.

The mean AUV speed in the advancing direction obtained for the entire mission tests is 0.23m/s, 0.11m/s and 0.16m/s for the DAR, SBSRE* and LM, respectively.

Moreover, the Figure 6.10 and 6.11, 6.12 represent the results obtained for the simulation tests with regard to the exploration performance. First, Figure 6.10 represents the executed paths for the three tests (DAR, SBSRE* and LM); the green shade represents the raw data density, which is normalized for each test. Since the frequency of the data gathering is equal for the three tests, the quantity of data gathered during the same time interval is also equal. However, the distribution of such data differs.

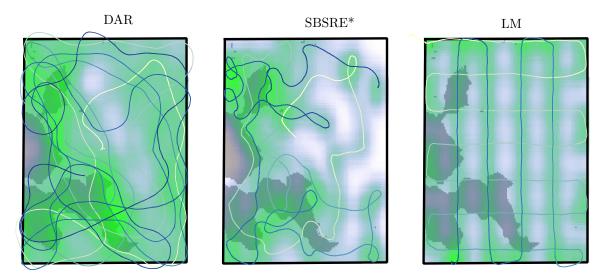


FIGURE 6.10: Paths followed and raw data density obtained during the execution of the simulation tests; from left to right, following DAR, SBSRE and LM strategies. The color in the path represents time, starts in yellow and ends in dark blue. The raw data density is represented with the green shade. The background represents the groundtruth, the white area contains *P.oceanica*

The differences regarding the raw data density distribution are the following: (i) the DAR method provides the highest data density in the meadow boundaries and that the lowest density regions belong to homogeneous data regions, (ii) the SBSRE* raw data density distribution is irregular, whilst it shows high density values in some boundary regions, it does not visit some others, and over records some regions due to being stopped while planning the next path, (iii) the predefined LM pattern results in evenly distributed data on the transects, and higher density spots on the section limits

Furthermore, the high density region located on the left part of the region of the DAR test corresponds to an area where noisy data was acquired due to inaccuracies on the segmentation process. This is the behavior that we expected to appear, revisiting regions where the recorded data is uncertain. In contrast, the SBSRE* does not target revisiting such are until the end of the test.

Whilst the LM tests provide more data density in the extreme of the section maneuvers due to a slow vehicle speed required for precise path following, the DAR and SBSRE* tests provide higher data density in high uncertainty areas, such as meadow bounds or heterogeneous data regions, being the DAR faster and more fluid visiting high uncertainty areas. In contrast with the smooth paths provided by the DAR method, the SBSRE* results in complex paths that are difficult to be followed by the AUV. This difference arises from the fact that the DAR method penalizes path turns in the DF-MCTS planner, whilst SBSRE* does not apply any restriction to path turns.

Figure 6.11 presents the results obtained during the three tests in terms of the MDE, SDE, AUC ROC and CP metrics described at the beginning of the section.

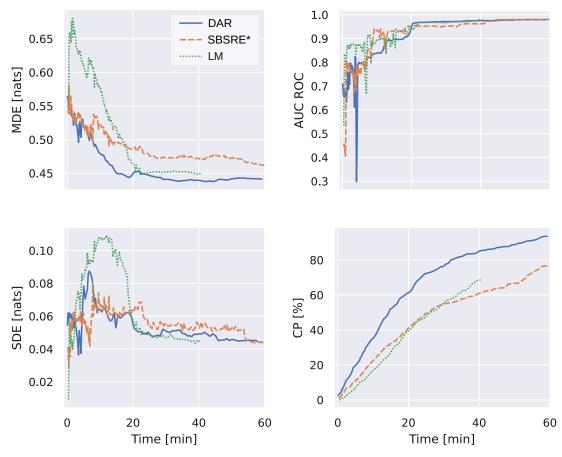


FIGURE 6.11: Results of the DAR, SBSRE* and LM in simulation.

- The DAR method presents the fastest reduction of the MDE metric and a low SDE, which means that the AUV efficiently retrieves data from high uncertainty regions and reduces the entropy of the predicted seagrass distribution.
- The SBSRE* provides a fast reduction of the MDE at the beginning of the mission. At the beginning it reduces the entropy by targeting the regions with the highest predicted variance values (the SDE maintains low values). However, approximately after the minute 10, the MDE reduction rate gets slower.
- The LM results in a final value of the MDE lower than the SBSRE*. However, during the mission execution, it achieves the highest MDE and SDE values, which means that generated predictions provide high uncertainty.
- In terms of CP, the DAR method provides the best results. Thanks to the smoothness of the informative paths computed, the robot is able to achieve higher speeds that result in an increased CP.

At the beginning of the exploration the AUC ROC metric is very unstable due to that the modelling process has very limited data for training. In particular, at the instant 4'25min of DAR test the AUC ROC of the estimated map shows a large negative peak. As more data is recorded the AUC ROC improves and stabilizes at minutes 20', 22', and 33' for the DAR, LM and SBSRE*, respectively. Which means that, from those points the modelling thread has sufficient data to learn a proper model.

Finally, Figure 6.12 represents the acquired data and the online estimated maps at four different time instants from the DAR execution. As the AUV navigates the target area, the environment model gets

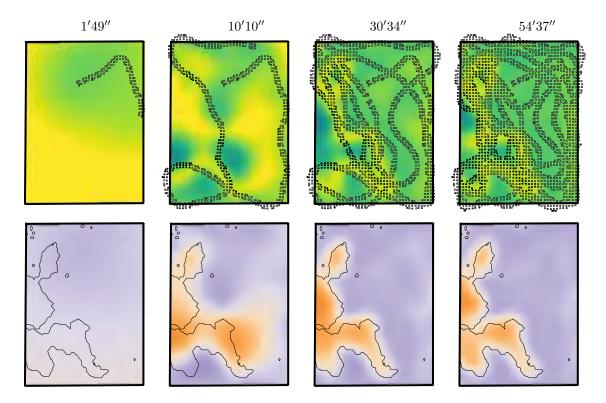


Figure 6.12: Representation of four different instants during the execution of the DAR method in simulation. In the top row, the square markers represent the samples (black squares for background, white squares for $P.\ oceanica$ detection) with resolution R used to train the GP; and the color-map represent the predicted variance map, blue for low variance regions and yellow for high variance regions. In the bottom row the black line represents the contour of the seagrass meadow generated from the groundtruth image, and the color-map represent the predicted map; purple for $P.\ oceanica$ and orange for background.

more accurate, starting by a naive prediction at the time instant 1'49'' to a more confident prediction at 45'37'', where only the boundary regions present high uncertainty value (yellow on the top row), and the P. oceanica distribution (purple on the bottom row) is predicted with high precision, given the spatial resolution used.

Notice how the right part of the map presents noisy samples (false negative samples) that increase the uncertainty value and attracts the robot navigation to gather more data, path in Figure 6.10.

6.3 Field experiments

This section describes a set of three field experiments performed to evaluate the proposed exploration framework. The main objective of the experiments was to explore some predefined locations to obtain a fast representation of the seagrass distribution. The experiments have been carried out in the Mallorca Island, using the AUV Turbot with the payload configuration described in Chapter 5.

- The first field-test (CSJ) was executed at the south of Mallorca in the Colonia de Sant Jordi (39.326284, 2.986278). The target area had an extension of 10,212m² and was located in a shallow water region which depth ranged between 3.5 and 5.5 meters. Covering a seagrass meadow that was visible form aerial images in order to be able to assess the modelling performance by comparing the online estimated maps with the existing aerial images of the area. For this experiment we set a maximum exploration time of 90min, and due to the low altitude the AUV was commanded to navigate on surface.
- The second field-test (PA1) was executed at the south-west of the island in the *Port d'Andratx* (39.543330, 2.377940). Due to the limitations imposed by maritime traffic in this case the target area was smaller, with an extension of $4,156m^2$ and a depth ranged between 10 and 18 meters. In this case the seabed distribution could not be inferred from existing aerial images, it was unknown at the time of planning the target area for exploration. For this experiment we set a maximum exploration time of 50min and a recording altitude of 3.0m.
- The third field-test (PA2) was executed on the same target area as PA1, using the same configurations. The unique difference with respect to PA1 was that in this case the localization filter of the AUV used the absolute positioning measurements provided by the USBL framework.

The Tables 6.5a and 6.5b provide the common parameter configurations, respectively for replanning and the modelling threads, for both experiment campaigns CSJ and PA.

Table 6.5: Common parameter configurations for the CSJ and PA experiments.

(A) (B) Replanning Parameters CSJ/PA **Modelling Parameters** CSJ/PA SGPMC Planning time 15sModel Kernel Information function DEMatérn 32 Kernel scale 1.0 Alpha (Utility) 30 Omega (Reward) Likelihood Beta 1.0 Discount factor 0.8 Likelihood Scale 0.5Learning rate 0.9Optimizer Scipy 32 Rollouts number Iterations 4,000 Epsilon 0.01Induction points N_{IP} 200 Number of nodes 20,000 Point cloud resolution [24,18]px

The configuration used for the DAR method was based on the configurations tested on Chapter 4. The distance budget B and the maximum neighbor distance d_2 were automatically set using the perimeter of its corresponding target area.

6.3.1 Colonia de Sant Jordi (CSJ)

Figure 6.13 provides a representation of the CSJ region as well as the target area used for exploration, such area (in pink) has an extension of $10,212m^2$ and a perimeter of 370.8m.



FIGURE 6.13: Target area for the *Colonia de Sant Jordi* (CSJ) region, 39.326284, 2.986278, on top an aerial image of the region from the *Instituto Geodésico Nacional* (PNOA 2018 campaign)

Table 6.6 details the navigation configuration used for the LOSCTE in this location. Please notice that in this case we used a configuration with higher speeds than in the simulation tests in order to provide a faster exploration performance at expenses of reducing data quality. Here we (i) doubled the minimum speed and lookahead distance, allowing higher path tracking errors, and (ii) increased a 20% the maximum speed, allowing higher image blurring.

Table 6.6: Navigation configuration for the CSJ region

Navigation Parameters	CSJ
Minimum speed	0.1m/s
Maximum speed	$\begin{array}{c c} 0.1m/s \\ 0.6m/s \end{array}$
Lookahead distance	4.0m
Speed transition distance	3.0m
Velocity ratio	0.1
Altitude	_

The objective of this test is the validation of the AVIG architecture in field. The test was executed with the vehicle on surface in order to abstract of the underwater navigation localization errors and to explore a seagrass meadow that could be easily inferred from aerial images to serve as a groundtruth.

Regarding the specific configuration parameters of the replanning thread, the neighbor distance d_1 was conserved the same as the used in the simulation experiments and neighbor distance d_2 and the budget \mathcal{B} were computed from the perimeter of target area. Table 6.7a provides such set of specific parameters.

Table 6.7: Specific replanning and modelling configurations for the CSJ experiment.

(A) (B)			
Replanning Parameters	CSJ	Modelling Parameters	CSJ
Neighbor distance d_1	1.5m	Training samples N_S	2000
Neighbor distance d_2	23.2	Sample resolution R	2.26m
Budget \mathcal{B}	92.7m	Induction points density IPD	$0.02m^{-2}$

Moreover, regarding the configuration of the environment modelling thread, the maximum number of training samples was set to 2000. The resulting sample resolution and the induction points density is provided in Table 6.7b. The next sections review in detail the results obtained in terms of data processing, modelling and replanning.

Data processing

The data processing results include the image recording, the image segmentation and image projection. Figure 6.14 shows an example of the kind of images acquired together with their segmentation output.

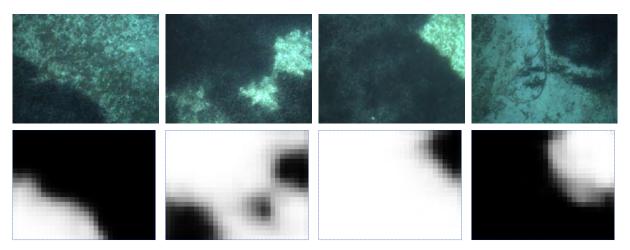


FIGURE 6.14: Image examples form the CSJ experiment and their corresponding low resolution segmentation output.

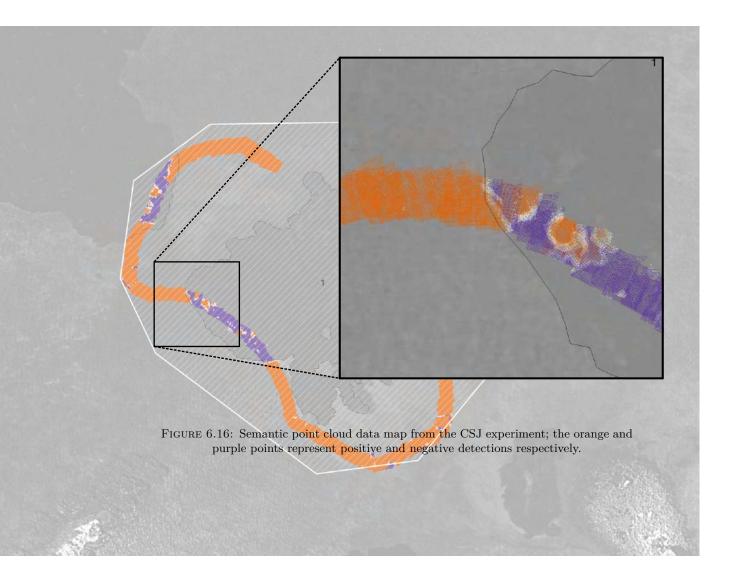
From visual inspection and without the generation of groundtruth for assessing the accuracy they show a rather good segmentation performance using the low resolution output of the CNN, resulting in a mean segmentation frequency of 0.461Hz that sufficient overlap between successive semantic point clouds.

Figure 6.15 provides a representation of the semantic point clouds generated during mission time. The black points represent areas with low probability of P. oceanica presence, whereas white points represent higher probabilities.

In addition, Figure 6.15 provides a deeper insight on the level of detail obtained with the semantic point cloud data. In this case the positive detections ($P.\ oceanica$) are represented with orange points, while negative detections (not $P.\ oceanica$) are represented with purple.

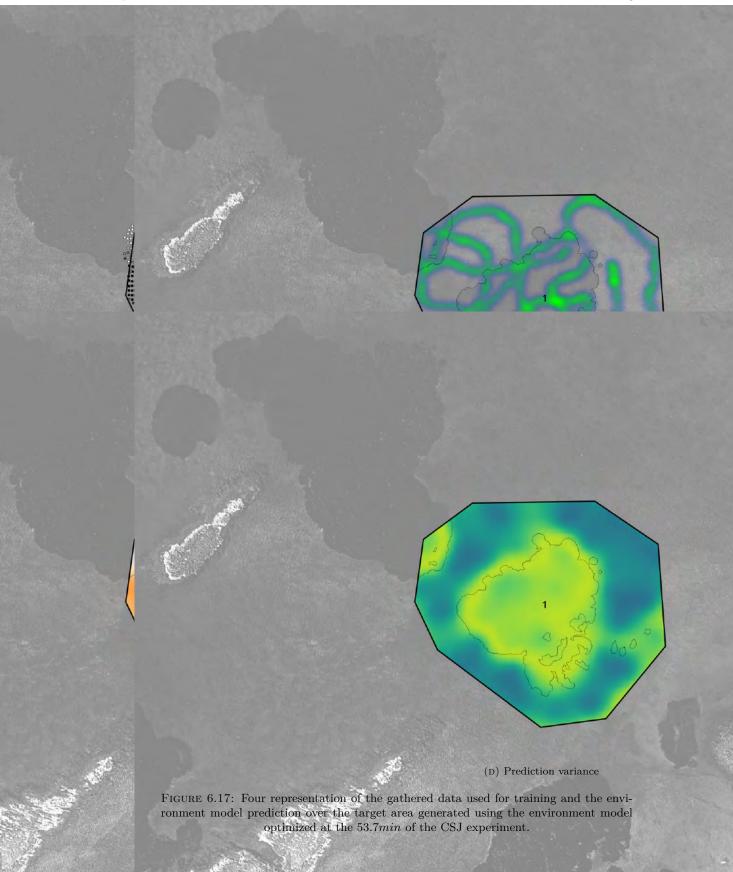


Figure 6.15: Semantic point clouds from the CSJ experiment projected to global coordinates; white points represent areas covered with $P.\ oceanica$, and black points represent background areas.



The white points represent intermediate probabilities. Please notice (i) the contrasting measurements in the same spots caused by the measurement noise and (ii) the accuracy of the meadow bounds inferred from the point cloud data with respect to the bounds inferred from the aerial image.

The processed data is considered to be consistent with what is inferred from the aerial image and valid



From this figure take a closer look to the training samples distribution of Figure 6.17a. From visual inspection it can be noticed that the sample distribution is biased towards negative values—i.e. black squares on the figure—. Figure 6.18 illustrates the histogram of the samples used for training the last iteration of the mapping thread of this experiment. Besides, the red plot represents the probability density function (PDF) of the beta function used for the SGP likelihood. Using a scale parameter of the beta distribution of s=0.5 the SGP likelihood matches the samples' distribution, providing higher probabilities to the extreme values with particular intensity to the negative extreme due to the biased training samples distribution.

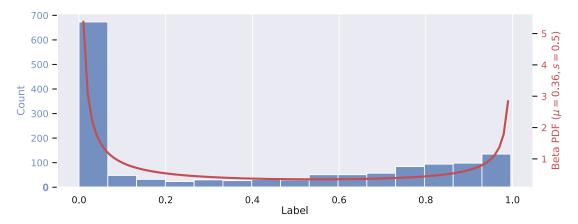


FIGURE 6.18: Histogram of the training samples used for environment modelling for the CSJ experiment and probability density function of the beta function used as likelihood function of the SGP.

The likelihood function used for SGP modelling takes the credit of such high variance in heterogeneous data regions. The beta function provides higher beliefs for extreme data samples; either *P. oceanica* or background. However, since the resolution size used for sampling is too large to capture the data complexity in heterogeneous data regions, such regions produce intermediate sample values that result in high uncertainty. This is a key result that enables keeping high entropy values in meadow boundaries and patchy regions to reinforce the data gathering in such areas.

Moreover, notice that each of the representations on Figure 6.17 shows a black contours that pictures the meadow bounds inferred from the aerial image. The values of the samples used for training correspond with the value inferred, the samples inside the contours show higher probabilities of positive detections that contrast with the values of the samples that are found outside. Figures 6.17c and 6.17d show the prediction provided by the environment model trained with the aforementioned training samples. The predictions of the SGP model coincide with the inferred bounds, providing higher prediction variance to heterogeneous data regions such a meadow bounds and patchy regions.

The likelihood function used for SGP modelling takes the credit of such high variance in heterogeneous data regions. The beta function used for the likelihood representation provides higher beliefs for extreme data samples; either *P. oceanica* or background. However, since the resolution size used for sampling is too large to capture the data complexity in heterogeneous data regions, such regions produce intermediate sample values that result in high uncertainty. This is a key result that enables keeping high entropy values in meadow boundaries and patchy regions to reinforce the data gathering in such areas.

Finally, Figure 6.19 illustrates the resulting modelling time and the coverage percent as the exploration mission goes forward. The figure shows a correlation between coverage and modelling time, resulting in 40s of modelling time after 90min of mission, with a CP around 60%.

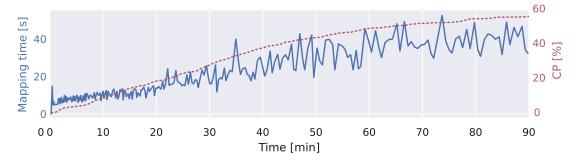


FIGURE 6.19: Resulting modelling time and coverage percent obtained during field-testing the AVIG framework.

Replanning

Figure 6.20 represents the path traversed during the exploration, given by the absolute localization filter of the AUV. The color in the path represents time, starts in yellow and ends in dark blue. The figure also represents the raw data density at the end of the mission with a green shade and the hand-labeled contours of the seagrass meadow interpreted from an aerial image. Observe that the more repeatedly recorded areas, providing the highest data density, correspond to the areas covered by seagrass. The AUV has the desired behavior of revisiting uncertain spots that contain heterogeneous data regions.

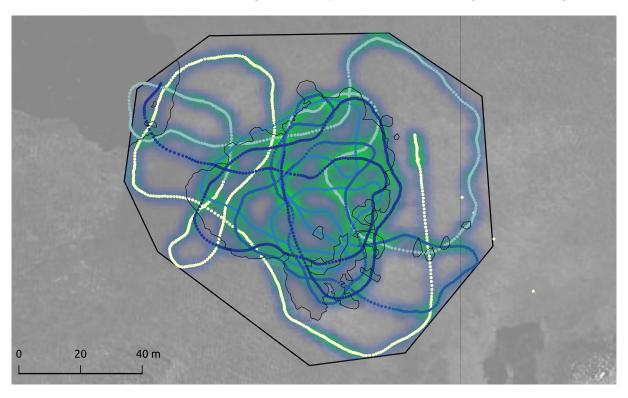


FIGURE 6.20: Path followed and raw data density obtained in field-test following DAR method. The color in the path represents time, starts in yellow and ends in dark blue.

The raw data density is represented with the green shade.

Figure 6.21 shows five representations of the gathered data and the estimated map of different time instants of the mission in order to represent the evolution of the exploration.

The top row of the figure shows the samples used to train the SGP as square markers and the predicted variance map with a color-map, dark blue for low variance regions and yellow for high variance regions. The bottom row represents the predicted seagrass distribution with a color-map where purple indicates

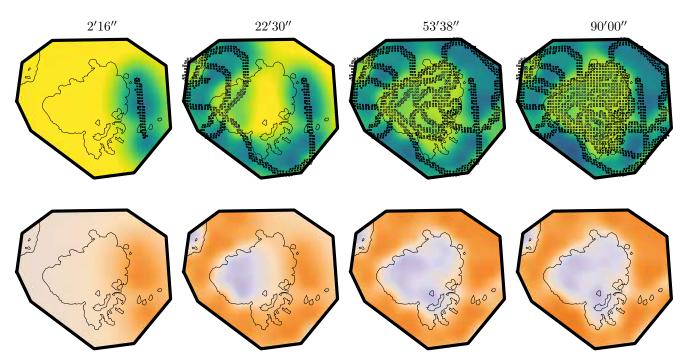


FIGURE 6.21: Representation of five different instants during the execution of the DAR method in field-test. In the top row, the square markers represent the samples (black squares for background, white squares for *P. oceanica* detection) used to train the SGP and the color-map represent the predicted variance map; blue for low variance regions and yellow for high variance regions. In the bottom row the black line represents the contour of the seagrass meadow obtained from the groundtruth image, and the color-map represent the online predicted map; purple for *P. oceanica* and orange for background.

presence of *P. oceanica* and orange indicates background. The third step corresponds to the model studied in the modelling section. As the exploration mission advances and more training samples are stored, the environment model predictions improve, providing results that are closer to the bounds estimated from aerial images. Notice that during the first two represented steps the robot tends to cover the area, the higher uncertainty is located on unrecorded spots, while during the last two represented steps, the robot tends to exploit the environment knowledge by prioritizing the recording of heterogeneous data regions.

Finally, Figure 6.22 represents the MDE and SDE computed from the successive estimated maps during the exploration mission. Both metrics show a fast convergence during the initial 30min, that corresponds to the results illustrated in Figure 6.21; the AUV performs a fast exploration of the target area up to that time increasing the map informativeness. Then, the exploration framework exploits the environment knowledge by gathering data mostly in heterogeneous data regions. Whilst the AUV gathers more data in such regions, the use of a fixed resolution size of the samples used for training bound the MDE reduction.

6.3.2 Port d'Andratx (PA)

Figure 6.23 provides a representation of the PA region as well as the target area used for exploration, such area (in orange) has an extension of $4{,}156m^2$ and a perimeter of 262.0m.

Table 6.8 details the navigation configuration used for the LOSCTE in this location. Please notice that the configuration used for the PA tests is very conservative when compared with the CSJ configuration. Whereas a fast navigation configuration has shown to provide good results during the CSJ experiments, in this case we decided to use a more prudent navigation configuration in order to provide a first validation of the AVIG architecture working in underwater mode. Here we used a homogeneous low speed sampling

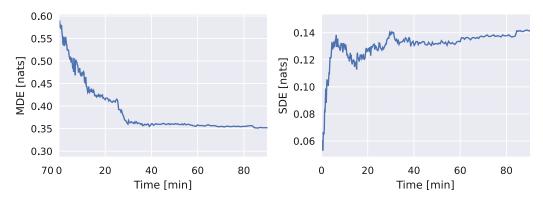


Figure 6.22: Results obtained in field-test.



FIGURE 6.23: Target area for the $Port\ d$ 'Andratx (PA) region, 39.543330, 2.377940, on top an aerial image of the region from the $Instituto\ Geod\'esico\ Nacional\ (PNOA\ 2018\ campaign)$

navigation configuration typically used to follow preprogrammed lawn mower patterns. This configuration provided higher resolution data (low image blurring) at expenses of a smaller coverage for the given exploration time.

Table 6.8: Navigation configuration for the PA region.

Navigation Parameters	PA
Minimum speed	0.05m/s
Maximum speed	0.35m/s
Lookahead distance	4.5m
Speed transition distance	3.0m
Velocity ratio	0.05
Altitude	3.0m

Table 6.9a provides the specific replanning parameters used for these tests. In this case the neighbor distance d_1 was increased to 5.0m to reduce the size of the graph and the neighbor distance d_2 and budget \mathcal{B} calculated from the target area resulted in 16.4m and 65.7m respectively. Finally, regarding the configuration of the modelling thread, for these experiments we tested a higher maximum number of samples in order to model the environment with a smaller spatial resolution, the specific configuration is provided in Table 6.9b.

Table 6.9: Specific replanning and modelling configurations for the PA experiment.

PA PA Replanning Parameters **Modelling Parameters** Neighbor distance d_1 5.0mTraining samples N_S 5000 Neighbor distance d_2 16.4mSample resolution R0.92m $0.12m^{-2}$ Budget \mathcal{B} 65.7mInduction points density IPD

Summarizing, the key differences between the PA and the CSJ experiments is that in this case (a) the navigation speed is slower, (b) the vehicle navigates underwater maintaining a constant altitude, (c) the neighbor distance d_1 is set to a larger value in order to reduce the number of neighbor candidates for each node, and (d) the environment modelling use the highest number of samples.

The next sections review in detail the results obtained in terms of navigation, data processing, modelling and replanning.

Navigation

These field-tests were more challenging for the altitude estimation than other of the presented field-test on this Thesis, in this scenario the DVL provided many failed measurements. In any case, the altitude estimation was robust enough to generate a proper altitude value for navigation. Figure 6.24 shows the altitude estimated by the filter presented in Chapter 5, together with the vehicle depth.

Figure 6.25 provides the histogram of the altitude measurements of each of the individual inputs of the estimation filter—i.e. DVL, echosounder and visual—. Notice that the negative measurements indicate that the measurement is invalid, the DVL has the largest amount of invalid measurements followed by the visual altimeter.

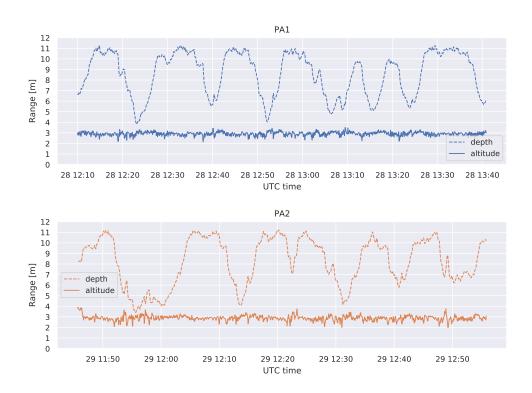


FIGURE 6.24: Altitude and depth during the PA field-tests

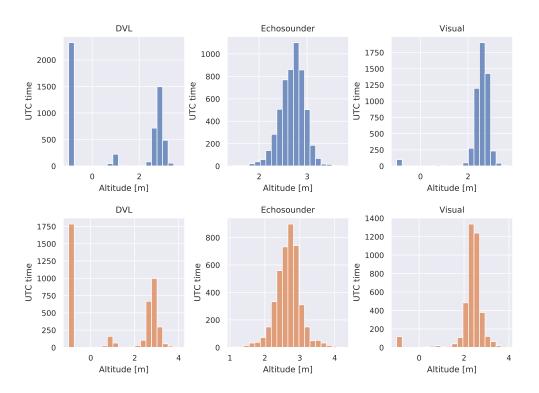


FIGURE 6.25: Altitude measurement histograms during the PA field-tests

Data processing

Figure 6.26 shows an example of a set of images acquired in PA together with their segmentation output. From visual inspection the segmentation seems accurate enough to guide the exploration. For these experiments the segmentation of the images was executed at 0.2Hz average frequency.

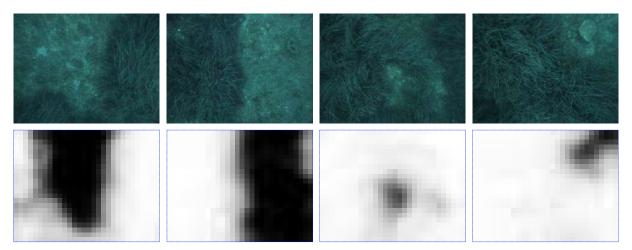


FIGURE 6.26: Field-test image examples form PA and their corresponding low resolution segmentation output.

Modelling

Figure 6.27 provides a representation of the environment prediction at the last modelling iteration, respectively for the (a) PA1 and (b) PA2 field experiments.

First notice that the two environment models provide contrasting predictions, that are the result of the use of different samples for training. The lack of correspondence between the training samples of both models is attributed to errors on the robot localization. The key difference between both tests is that the first test did not use USBL measurements for navigation, while the second did. Hence, in the absence of a groundtruth the data generated during PA2 test is believed to be more accurate. The data recorded during the PA1 test was prone to localization drift.

In terms of environment modeling, the resulting mean and variance maps for environment model predictions are well corresponded with the samples used for training for both tests. In both cases the positive detection regions (probable seagrass meadows) provide higher variances. Figures 6.28 and 6.29 provide the histograms generated from the training samples of PA1 and PA2 respectively. Each figure provide (a) the complete histogram and two partial histograms representing the samples contained in the interval (b) [0.0,0.1] and (c) [0.9,1.0].

Firstly, from the complete histogram figures notice that the corresponding beta PDF matches the distribution of the training samples. These results support the decision of using a beta function to represent the likelihood function.

Secondly, from the partial histogram figures notice that there is a clear difference on the resulting distribution at the extreme values. The negative detection samples seem to be more biased towards the extreme of the label interval than the positive detections. This certainty on the negative detection samples seems to be the cause of the reduced predicted variance of the environment model on the regions that are covered by such samples. The cause of that uneven distribution at the extremes should be further addressed.

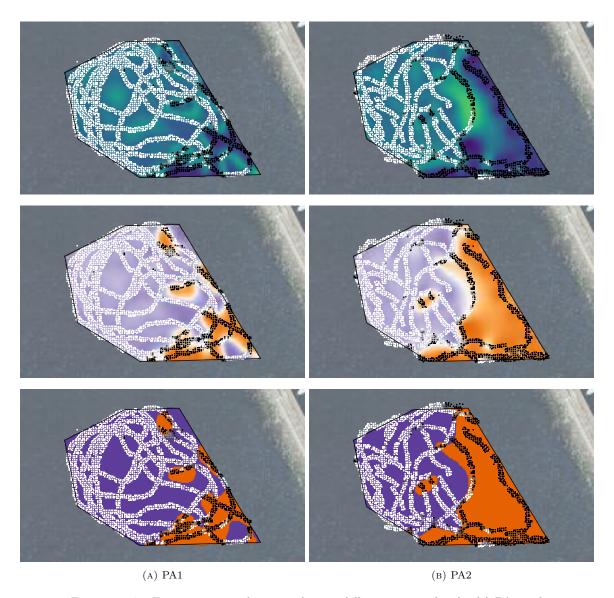
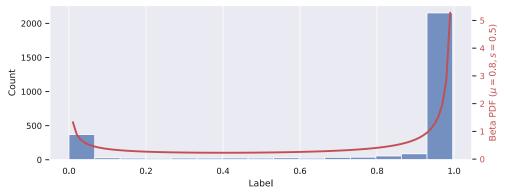
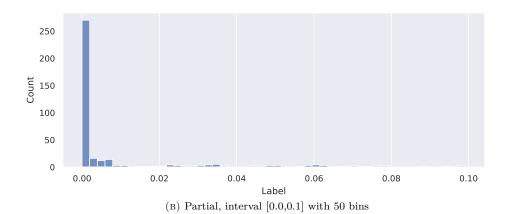


FIGURE 6.27: Environment prediction at last modelling iteration for the (a) PA1 and (b) PA2 field experiments. From top to bottom: (1) the model variance, (2) the model mean and (3) thresholded model mean. The model variance is represented with dark blue for low variance regions and yellow for high variance regions. The model mean is represented with orange for negative predictions and purple for positive predictions. The three representations include a superposition of the samples used for training the environment model in form of square markers, the fill color from black to white represent the probability of being positive [0,1].



(a) Complete histogram, interval [0.0,1.0] with 15 bins.



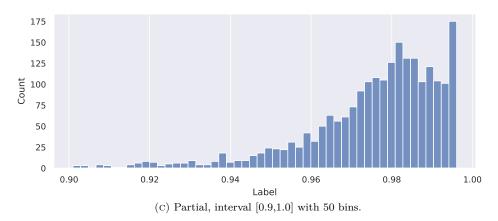
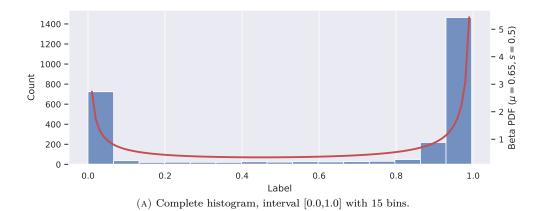
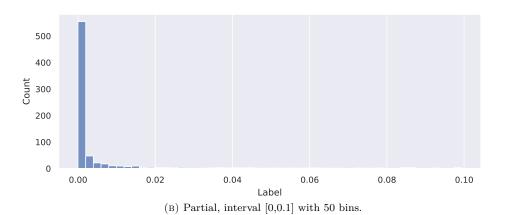


FIGURE 6.28: Training samples histogram for PA1 experiment.





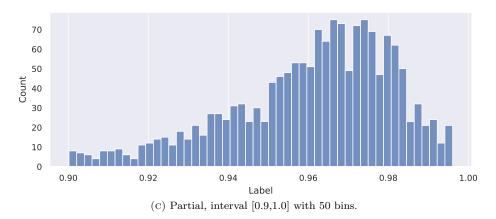


FIGURE 6.29: Training samples histogram for PA2 experiment.

Replanning

In this section we reach the last step of the analysis, the replanning. Figures 6.30 and 6.31 provide a representation of three replanning steps (left to right), respectively for the field experiments (a) PA1 at the minutes 0.8, 16 and 54 and (b) PA2 at the minutes 0.4, 14 and 48. From top to bottom, these figures show (1) raw data density, the more yellow, the higher; (2) model variance, the more yellow, the higher; (3) model mean, purple for positives and orange for negatives; and (4) node utility, dark blue for the highest.

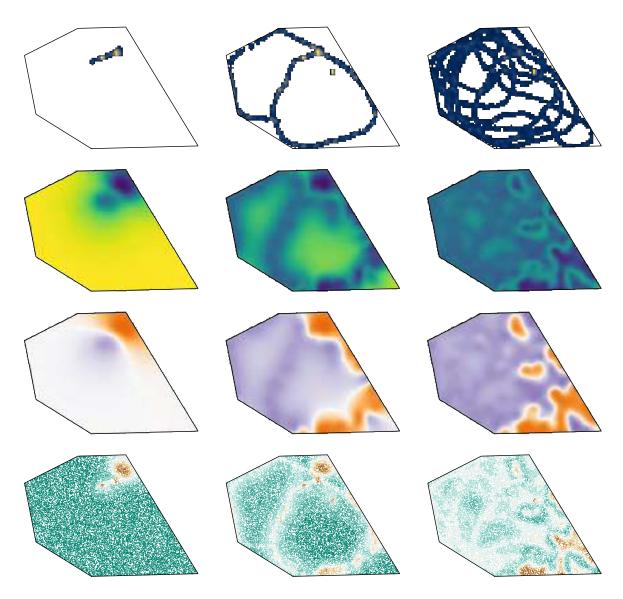


FIGURE 6.30: Representation of three replanning steps (left to right) for the PA1 field experiment at the minutes 0.8, 16 and 54. From top to bottom, these figures show (1) raw data density, the more yellow, the higher; (2) model variance, the more yellow, the higher; (3) model mean, purple for positives and orange for negatives; and (4) node utility, dark blue for the highest.

At the first step of both experiments the scarcity of training samples results in a very uncertain environment model that provides a large predicted variance. In the same step, these areas of high variance result in high utilities of the nodes used to build the decision tree for replanning. Since the farther regions from the recorded area provide the higher uncertainties, the nodes of the DFMC used for planning provide

the highest utility values around such areas and condition the navigation to explore the farther regions of the target area.

Notice that at the second step shown of both experiments the robot has already traveled a distance enough to record sufficient benthic data to provide a fairly good environment estimation. At this step the exploration agent has already visited the higher variance areas and the higher utility values of the DFMC node network are located on regions that have not been recorded or that the recorded data is predicted to be heterogeneous. Figure 6.32 illustrates the results obtained from both experiments using the validation metrics described in the experiments' section of Chapter 4. Notice that the MDE is reduced around a 50% for the time the mission reaches the aforementioned second step. That happens at the minute 14 and 16 respectively for PA1 and PA2.

At the third illustrated step, minutes 54 and 48 respectively for PA1 and PA2, the results show a further reduction of the predicted variance due to the large amount of recorded data. At that point the MDE has been reduced a 65% and seems to have converged to a value conditioned by the sampling resolution and trajectory smoothness. The number of samples used for training the SGP in these experiments has shown to result on a very accurate modelling. However, it comes at the expenses of higher modelling times than the resulted in the CSJ experiments. Figure 6.32 shows that the initial mapping iterations took around 20s to complete which is a very good time considering that the planning is launched every 15s. Nonetheless, the last iterations took around 10 min. The resulting modelling time depends on the sampling resolution, the induction points density, the computational load of the computer and the number of samples used for training.

6.4 Discussion

The AVIG framework has been successfully integrated in a ROS-based software architecture and tested in simulation and in field. The results show that using this framework the vehicle is driven to cover the region with higher uncertainty given by the GP model whilst producing smooth paths easy to follow. Regions with increased GP variance imply that the gathered information is highly uncertain or incomplete thus it needs to be revisited.

The experiments performed showed that the configurations used can be further tuned for generalization offering a proper trade off between image segmentation, environment modelling and replanning. A parameter configuration in between of the configurations used for the CSJ and PA experiments could offer such trade off. First, the navigation configuration will depend on the desired pixel size and position accuracy for the recorded data as described in Chapter 5. Second, the modelling configuration will depend only on the capacity of the computer on board of the AUV by defining the maximum number of training samples to be used. This would apply for the studied application of seagrass modelling. The set-up of other applications would require a review of other configuration parameters such as the kernel and likelihood functions to be used. Third, the replanning configuration has been mostly reliant on the robot navigation constraints, only considering the commanded path roughness that could be allowed. However, the experiments have shown that such configuration should also rely on the environment model to be learned. More detailed models should require less restricted trajectory paths in order to be able to execute more complex trajectories to target, for instance, more complex seagrass meadows or small patchy regions. Hence, the neighbor distance (d_1) of the planning configuration could be defined proportionally to the sampling resolution used by the environment modelling thread, which is in turn reliant of the maximum number of samples. Using such proposed parameterization, the exploration of a new target area would exclusively depend on the target area shape and the computer capacity of the exploring agent.

Moreover, the resulting convergence of the MDE metric from all the simulation and field experiments has shown to provide very good quality to be used, in conjunction with the time budget, as a stop condition for the exploration. The slope of MDE could be used as a threshold to either stop the exploration or refine the configuration (increasing the maximum number of samples) in order to be able to target further key spots that will improve the performance of the exploration.

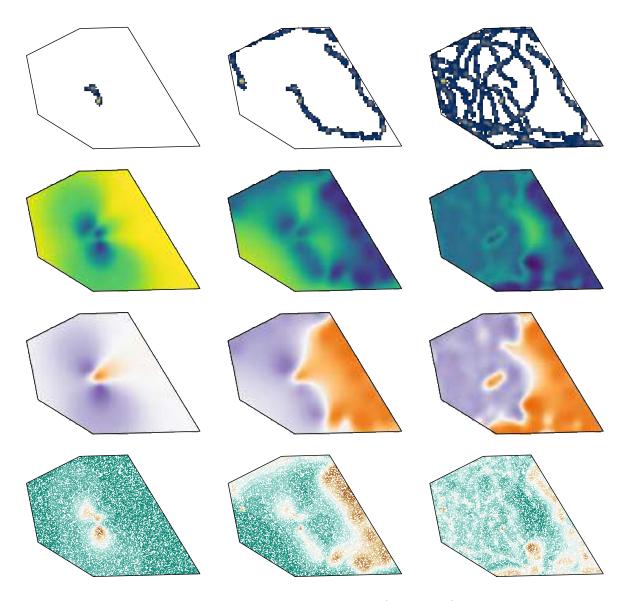


Figure 6.31: Representation of three replanning steps (left to right) for the PA2 field experiment (a) PA2 at the minutes 0.4, 14 and 48. From top to bottom, these figures show (1) raw data density, the more yellow, the higher; (2) model variance, the more yellow, the higher; (3) model mean, purple for positives and orange for negatives; and (4) node utility, dark blue for the highest.

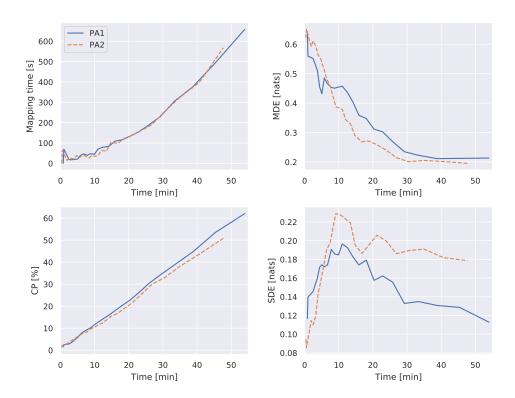


FIGURE 6.32: Results obtained during the PA field-tests.

Chapter 7

Conclusions

The main goal of robotic exploration is to collect data in areas that humans cannot reach, such as in deep water, in space, or in missions involving hazardous environments. Such exploration can be performed by means of an operator controlling remotely a robot system. However this relies on having a proper bidirectional communication between the operator and the robot, for instance to transmit robot images and operator commands. The main issue is that in many scenarios such communication is not possible, or is very limited, and the robot has to be equipped with some degrees of autonomy.

Having a reliable autonomous navigation and an online processing of the data recorded by the robot and an online environment modelling have supposed a great advantage during benthic habitat mapping campaigns. Enabling a higher degree of autonomy for adaptive mission replanning when the robot can not be tethered and a proper bidirectional communication can not be established. Furthermore, such online data processing method has proved to be useful also to provide a deeper insight about the benthic habitat distribution to AUV operators, easing the targeting of high-informative areas and planning sampling missions when the AUV can be tethered.

All in all, this Thesis has enlarged the applications and methods of autonomous robotic exploration by proposing different methods to improve online environment perception and proposing a software architecture that includes an adaptive visual information gathering method to maximize information gain when sampling seabed images for benthic habitat mapping.

This Chapter concludes the Thesis, presenting the summary of the completed work in Section 7.1, the main contributions in Section 7.2 and future work suggestions in Section 7.3.

7.1 Summary

Chapter 3 describes and field validates a method to ease data-gathering campaigns for P. oceanica meadows by performing (a) an online image segmentation using an encoder-decoder CNN, (b) a projection of such processed data to world coordinates using stereo image processing and the estimated robot trajectory and (c) an estimation of the collected data uncertainty using an SGP. A series of field-tests were carried out in order to provide enough data to select the most suitable CNN and SGP configuration. This chapter validated the use of a modified VGG16-FCN8 network, reducing the computation time in a 12.8%. With regard to the selection of an SGP configuration, the SGPMC model with kernel $\phi_{M_{12}}$ provided the best overall performance. Providing the shorter training times and best uncertainty representation from the

models considered. Furthermore, field-testing shown that using a likelihood beta function with scaling factor s=1 in the SGP model is fundamental in order to model uncertainty associated to seagrass meadow bounds. Validation tests using such a likelihood function have shown that (1) the surveyed areas which are considered to be more salient show higher SGP variance than areas containing similar sample values and (2) the estimation shows good prediction properties in non-sampled areas too, inferring, for instance, meadow bound continuations.

Chapter 4 describes and validates a novel design of a novel method for decision-time adaptive replanning (DAR). Such DAR method is coupled with a DF-MCTS strategy for IPP, and joins two advantages of graph-based and sampling-based methods: (a) initializes a node network to set neighbor relations between sampling locations (which reduces computation during online execution), and (b) samples paths in the node network through tree search following a decision-time strategy (that provides near-optimal solutions in an anytime manner). The results showed that the DAR method with DF-MCTS provided the fastest MDE reduction when compared with the very relevant strategy proposed by *Viseras et al.* [113] and with a predefined lawn mower pattern mission. The results showed that the variable distance between sampling nodes and the penalization of sharp turns in the reward function provide smooth paths that are secure to be followed by the LOSCTE controller.

Chapter 5 enumerates the different sources of uncertainty when using AUVs for recording geo-localized images of benthic environments, and derives the equations to model the localization uncertainty propagation from camera to image pixel, and to model the pixel-wise image blurring cased by the robot motion. Then, it introduces the proposed software architecture, and describes the build in navigation module that include several features to reduce data uncertainties and for monitoring. Such features include (a) the integration of an USBL framework to enable acoustic communications and positioning and allow untethered supervision off the AUV, (b) the integration of a set of localization filters implemented to reduce the uncertainty on the altitude and pose estimations. Then, the results obtained from a series of field experiments provide the accuracy of the localization of the robot, and the quality of the recorded data in terms of localization error and spatial resolution.

Chapter 6 describes and validates a novel architecture for autonomous exploration. Such architecture joins an upgraded version of the methods described in previous chapters for modelling, replanning and navigation in a framework that can be executed online on-board of the AUV. Moreover, the chapter evaluates the performance of the architecture on a series of simulation and field experiments when performing autonomous exploration of a target region. The results showed that using the proposed architecture the vehicle is driven to cover the region with higher uncertainty given by the SGP model, whilst producing smooth paths easy to follow. Regions with increased SGP variance imply that the gathered information is highly uncertain or incomplete thus it need to be revisited.

7.2 Contributions

This Thesis has contributed to the field of autonomous robotics by providing a field validated data-driven architecture for stereo image recording using AUVs. More precisely, this Thesis has contributed in terms of underwater navigation, online environment perception and adaptive mission replanning. Providing several high-level features to build an AUV software architecture based on ROS.

First, in terms of underwater navigation, this Thesis has contributed to the field by developing different methods to enable close range navigation to the seabed in shallow water environments covered by dense seagrass meadows. In particular this Thesis has contributed to tackle the challenging conditions of this type of navigation in terms of pose and altitude estimation. This includes:

- The integration of an USBL framework to provide acoustic positioning and communications with the AUV. Enabling the generation of absolute positioning measurements to bound the localization error of the recorded data and an untethered navigation to improve the AUV autonomy and endurance.
- The integration of a localization layer to include the integration of the USBL measurements on the pose estimation, and the integration of different altitude measurements on a new altitude filter to provide continuous altitude estimation.
- The experimental assessment of the developed methods and modelling of the recorded data quality in terms of localization accuracy and spatial resolution.

Second, in terms of online environment perception, has contributed by developing and introducing a set of features that provide a semantic understanding of the environment that it is being recorded by the robot, during mission time. This includes:

- Running a semantic image segmentation CNN encoder-decoder architecture trained to differentiate *P. oceanica* from background, and modifying such CNN to improve online performance by reducing computation times.
- Generating a data-base containing the three-dimensional projection of the semantic images. Where the projection to global coordinates is performed using the robot pose and stereo disparity.
- Building a probabilistic environment model based on a sparse Gaussian process using the recorded data.

Third, in terms of adaptive mission replanning, this Thesis has contributed by developing novel strategies to provide adaptive mission paths that maximize data informativeness. This includes:

- Growing a tree by sampling mission paths following a novel depth-first Monte Carlo tree search (DF-MCTS) strategy based on Reinforcement Learning.
- Successively selecting the best path to follow according to an utility metric using a novel decision-time adaptive replanning (DAR) method.
- Coordinating the decision-making with the environment perception and the mission control using a novel adaptive visual information gathering (AVIG) architecture.

Furthermore, this Thesis provides an extensive validation of the features proposed in simulation and field-test. As well as complete model selection assessments to optimize the configurations proposed. The proposed methods have been integrated to be executed onboard of the AUV Turbot, and have shown very good performance to generate high-resolution data of benthic environments colonized by *P. oceanica*, an endemic seagrass of the Mediterranean in regression.

7.3 Future Work

This Thesis provides a proof of concept and opens the door of using AUVs for adaptive visual information gathering of benthic environments, and there are many lines of research to explore in the future.

The semantic understanding of the benthic environment could be improved by means of:

- Testing different supervised learning methods to allow multiclass segmentation of images, and pointcloud segmentation.
- Introducing the use of unsupervised learning methods for exploration of environments colonized by previously unknown habitats.

Improvements on the online modelling of the benthic environment will target the following: Such improvements will require also the use of a new environment model able to be trained with multiclass data:

- Using a multiclass kernel for the SGP.
- Testing of a Bayesian Neural Network.
- Defining and using heteroscedastic models to make use of the sampling uncertainty (localization and imaging conditions) for environmental modelling.

Moreover, it could be very interesting to be able to use prior existing IS and RS data:

- Fuse prior existing IS data to the environment modelling following a Bayesian Commite Maching algorithm.
- Integrate prior knowledge from different RS sources following a multimodal learning method.

In terms of adaptive planning, the most important future research lines are the following:

- Multirobot exploration, sharing low resolution maps for exploration in a collaborative manner.
- Include variable altitude and speeds for automatic determination of the image footprint and resolution trade-off.

- [1] C.M. Fortezzo, P.D. Spudis, and S.L. Harrel. "Release of the Digital Unified Global Geologic Map of the Moon at 1:5,000,000-Scale". In: 51st Annual Lunar and Planetary Science Conference. Lunar and Planetary Science Conference. 2020, p. 2760.
- [2] Larry Mayer et al. "The Nippon Foundation-GEBCO seabed 2030 project: The quest to see the world's oceans completely mapped by 2030". In: *Geosciences (Switzerland)* 8.2 (2018). ISSN: 20763263. DOI: 10.3390/geosciences8020063.
- [3] Christopher Whitt et al. "Future Vision for Autonomous Ocean Observations". In: Frontiers in Marine Science 7.September (2020). ISSN: 22967745. DOI: 10.3389/fmars.2020.00697.
- [4] Veerle A I Huvenne et al. Submarine Geomorphology. Ed. by Aaron Micallef, Sebastian Krastel, and Alessandra Savini. Springer Geology. Cham: Springer International Publishing, 2018, pp. 93–108. ISBN: 978-3-319-57851-4. DOI: 10.1007/978-3-319-57852-1.
- [5] Russell B. Wynn et al. "Autonomous Underwater Vehicles (AUVs): Their past, present and future contributions to the advancement of marine geoscience". In: *Marine Geology* 352 (2014), pp. 451–468. ISSN: 00253227. DOI: 10.1016/j.margeo.2014.03.012.
- [6] Robert Costanza et al. "The value of the world's ecosystem services and natural capital". In: Nature 387.6630 (1997), pp. 253–260. ISSN: 0028-0836. DOI: 10.1038/387253a0.
- [7] A. Ruiz-Frau et al. "Current state of seagrass ecosystem services: Research and policy integration". In: Ocean and Coastal Management 149 (2017), pp. 107–115. ISSN: 09645691. DOI: 10.1016/j.ocecoaman.2017.10.004.
- [8] J. Emmett Duffy et al. "Toward a coordinated global observing system for seagrasses and marine macroalgae". In: *Frontiers in Marine Science* 6.JUL (2019). ISSN: 22967745. DOI: 10.3389/fmars. 2019.00317.
- [9] Robert Orth et al. "A Global Crisis for Seagrass Ecosystems". In: BioScience 56.12 (2006), pp. 987–996. DOI: 10.1641/0006-3568(2006)56[987:AGCFSE]2.0.C0;2.
- [10] Luca Telesca et al. "Seagrass meadows (Posidonia oceanica) distribution and trajectories of change". In: Scientific Reports 5.1 (2015), p. 12505. ISSN: 2045-2322. DOI: 10.1038/srep12505.
- [11] Carlos M. Duarte et al. "Rebuilding marine life". In: Nature 580.7801 (2020), pp. 39–51. ISSN: 0028-0836. DOI: 10.1038/s41586-020-2146-7.
- [12] Gabriel Jordà, Núria Marbà, and Carlos M Duarte. "Mediterranean seagrass vulnerable to regional climate warming". In: *Nature Climate Change* 2.11 (2012), pp. 821–824. ISSN: 1758-678X. DOI: 10.1038/nclimate1533.
- [13] Arnaud Abadie et al. "From mechanical to chemical impact of anchoring in seagrasses: The premises of anthropogenic patch generation in Posidonia oceanica meadows". In: *Marine Pollution Bulletin* 109.1 (2016), pp. 61–71. ISSN: 0025326X. DOI: 10.1016/j.marpolbul.2016.06.022.

[14] Sotiris Kiparissis et al. "Illegal trawling and induced invasive algal spread as collaborative factors in a Posidonia oceanica meadow degradation". In: *Biological Invasions* 13.3 (2011), pp. 669–678. ISSN: 1387-3547. DOI: 10.1007/s10530-010-9858-9.

- [15] Carlos M. Duarte. "The future of seagrass meadows". In: Environmental Conservation 29.2 (2002), pp. 192–206. ISSN: 0376-8929. DOI: 10.1017/S0376892902000127.
- [16] M Waycott et al. "Accelerating loss of seagrasses across the globe threatens coastal ecosystems". In: Proceedings of the National Academy of Sciences 106.30 (2009), pp. 12377–12381. ISSN: 0027-8424. DOI: 10.1073/pnas.0905620106. arXiv: arXiv:1011.1669v3.
- [17] Charles F Boudouresque et al. "Regression of Mediterranean seagrasses caused by natural processes and anthropogenic disturbances and stress: a critical review". In: *Bot. Mar.* 52.5 (2009), pp. 395– 418. ISSN: 1437-4323. DOI: 10.1515/BOT.2009.057.
- [18] X. Durrieu de Madron et al. "Marine ecosystems' responses to climatic and anthropogenic forcings in the Mediterranean". In: *Progress in Oceanography* 91.2 (2011), pp. 97–166. ISSN: 00796611. DOI: 10.1016/j.pocean.2011.02.003.
- [19] Carmen B. de los Santos et al. "Recent trend reversal for declining European seagrass meadows". In: Nature Communications 10.1 (2019), p. 3356. ISSN: 2041-1723. DOI: 10.1038/s41467-019-11340-4.
- [20] Cecilia Lopez y Royo et al. "Seagrass (Posidonia oceanica) monitoring in western Mediterranean: implications for management and conservation". In: Environmental Monitoring and Assessment 171.1-4 (2010), pp. 365–380. ISSN: 0167-6369. DOI: 10.1007/s10661-009-1284-z.
- [21] Arnaud Abadie et al. "Seascape ecology in Posidonia oceanica seagrass meadows: Linking structure and ecological processes for management". In: *Ecol. Indic.* 87.December 2017 (2018), pp. 1–13. ISSN: 1470160X. DOI: 10.1016/j.ecolind.2017.12.029.
- [22] C. Pergent-Martini et al. "Descriptors of Posidonia oceanica meadows: Use and application". In: Ecological Indicators 5.3 (2005), pp. 213-230. ISSN: 1470160X. DOI: 10.1016/j.ecolind.2005. 02.004.
- [23] Francisco Bonin-Font, Antoni Burguera, and Jose-Luis Lisani. "Visual Discrimination and Large Area Mapping of Posidonia Oceanica Using a Lightweight AUV". In: *IEEE Access* 5 (2017), pp. 24479–24494. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2017.2764998.
- [24] Yolanda Gonzalez-Cid et al. "Machine learning and deep learning strategies to identify Posidonia meadows in underwater images". In: OCEANS 2017 Aberdeen. IEEE, 2017, pp. 1–5. ISBN: 978-1-5090-5278-3. DOI: 10.1109/OCEANSE.2017.8084991.
- [25] E. P. Green et al. "A review of remote sensing for the assessment and management of tropical coastal resources". In: Coastal Management 24.1 (1996), pp. 1–40. ISSN: 15210421. DOI: 10.1080/ 08920759609362279.
- [26] Tiit Kutser et al. "Remote sensing of shallow waters A 50 year retrospective and future directions". In: Remote Sensing of Environment 240.February (2020), p. 111619. ISSN: 00344257. DOI: 10.1016/j.rse.2019.111619.
- [27] Tien Dat Pham et al. "A review of remote sensing approaches for monitoring blue carbon ecosystems: Mangroves, sea grasses and salt marshes during 2010–2018". In: Sensors (Switzerland) 19.8 (2019). ISSN: 14248220. DOI: 10.3390/s19081933.
- [28] Bijeesh Kozhikkodan Veettil et al. "Opportunities for seagrass research derived from remote sensing: A review of current methods". In: *Ecological Indicators* 117.May (2020), p. 106560. ISSN: 1470160X. DOI: 10.1016/j.ecolind.2020.106560.

[29] M. S. Hossain et al. "The application of remote sensing to seagrass ecosystems: an overview and future research prospects". In: *International Journal of Remote Sensing* 36.1 (2015), pp. 61–114. ISSN: 13665901. DOI: 10.1080/01431161.2014.990649.

- [30] V. Pasqualini et al. "Mapping of Posidonia oceanicausing Aerial Photographs and Side Scan Sonar: Application off the Island of Corsica (France)". In: *Estuarine, Coastal and Shelf Science* 47.3 (1998), pp. 359–367. ISSN: 02727714. DOI: 10.1006/ecss.1998.0361.
- [31] Peter J. Mumby and Alasdair J. Edwards. "Mapping marine environments with IKONOS imagery: Enhanced spatial resolution can deliver greater thematic accuracy". In: *Remote Sensing of Environment* 82.2-3 (2002), pp. 248–257. ISSN: 00344257. DOI: 10.1016/S0034-4257(02)00041-X.
- [32] R Matarrese et al. "Applications of Remote Sensing Techniques for Mapping Posidonia Oceanica Meadows". In: IGARSS 2008 2008 IEEE International Geoscience and Remote Sensing Symposium. IEEE, 2008, pp. IV -906-IV -909. ISBN: 978-1-4244-2807-6. DOI: 10.1109/IGARSS.2008.4779870.
- [33] Jeffrey Barrell et al. "Evaluating the complementarity of acoustic and satellite remote sensing for seagrass landscape mapping". In: *International Journal of Remote Sensing* 36.16 (2015), pp. 4069–4094. ISSN: 13665901. DOI: 10.1080/01431161.2015.1076208.
- [34] Ruiliang Pu and Susan Bell. "Mapping seagrass coverage and spatial patterns with high spatial resolution IKONOS imagery". In: *International Journal of Applied Earth Observation and Geoinformation* 54 (2017), pp. 145–158. ISSN: 0303-2434. DOI: https://doi.org/10.1016/j.jag. 2016.09.011.
- [35] Bo Huang, Bei Zhao, and Yimeng Song. "Urban land-use mapping using a deep convolutional neural network with high spatial resolution multispectral remote sensing imagery". In: *Remote Sensing of Environment* 214.October 2017 (2018), pp. 73–86. ISSN: 00344257. DOI: 10.1016/j.rse.2018.04.050.
- [36] Fabien H. Wagner et al. "Using the U-net convolutional network to map forest types and disturbance in the Atlantic rainforest with very high resolution images". In: Remote Sensing in Ecology and Conservation 5.4 (2019), pp. 360–375. ISSN: 20563485. DOI: 10.1002/rse2.111.
- [37] F Eugenio, J Marcello, and J Martin. "High-Resolution Maps of Bathymetry and Benthic Habitats in Shallow-Water Environments Using Multispectral Remote Sensing Imagery". In: *Geoscience and Remote Sensing, IEEE Transactions on* 53.7 (2015), pp. 3539–3549. ISSN: 0196-2892. DOI: 10.1109/TGRS.2014.2377300.
- [38] Patrick C. Gray et al. "Integrating drone imagery into high resolution satellite remote sensing assessments of estuarine environments". In: *Remote Sensing* 10.8 (2018). ISSN: 20724292. DOI: 10.3390/rs10081257.
- [39] Eva Kovacs et al. "Seagrass habitat mapping: How do landsat 8 OLI, sentinel-2, ZY-3A, and worldview-3 perform?" In: Remote Sensing Letters 9.7 (2018), pp. 686–695. ISSN: 21507058. DOI: 10.1080/2150704X.2018.1468101.
- [40] Camila Brasil Louro da Silveira et al. "Coral reef mapping with remote sensing and machine learning: A nurture and nature analysis in marine protected areas". In: *Remote Sensing* 13.15 (2021). ISSN: 20724292. DOI: 10.3390/rs13152907.
- [41] Sepideh Tavakkoli Piralilou et al. "Landslide detection using multi-scale image segmentation and different machine learning models in the higher himalayas". In: *Remote Sensing* 11.21 (2019). ISSN: 20724292. DOI: 10.3390/rs11212575.
- [42] Pramaditya Wicaksono and Wahyu Lazuardi. "Assessment of PlanetScope images for benthic habitat and seagrass species mapping in a complex optically shallow water environment". In:

International Journal of Remote Sensing 39.17~(2018), pp. 5739-5765. ISSN: 13665901. DOI: 10. 1080/01431161.2018.1506951.

- [43] Dimitris Poursanidis et al. "Cubesats allow high spatiotemporal estimates of satellite-derived bathymetry". In: Remote Sensing 11.11 (2019). ISSN: 20724292. DOI: 10.3390/rs11111299.
- [44] Nam Thang Ha et al. "A comparative assessment of ensemble-based machine learning and maximum likelihood methods for mapping seagrass using sentinel-2 imagery in Tauranga Harbor, New Zealand". In: Remote Sensing 12.3 (2020), pp. 1–16. ISSN: 20724292. DOI: 10.3390/rs12030355.
- [45] Dimosthenis Traganos and Peter Reinartz. "Mapping Mediterranean seagrasses with Sentinel-2 imagery". In: *Marine Pollution Bulletin* 134.June (2018), pp. 197–209. ISSN: 18793363. DOI: 10.1016/j.marpolbul.2017.06.075.
- [46] Dimosthenis Traganos et al. "Towards global-scale seagrass mapping and monitoring using Sentinel-2 on Google Earth Engine: The case study of the Aegean and Ionian Seas". In: *Remote Sensing* 10.8 (2018), pp. 1–14. ISSN: 20724292. DOI: 10.3390/rs10081227.
- [47] Richard G. Lathrop, Paul Montesano, and Scott Haag. "A multi-scale segmentation approach to mapping seagrass habitats using airborne digital camera imagery". In: *Photogrammetric Engineering and Remote Sensing* 72.6 (2006), pp. 665–675. ISSN: 00991112. DOI: 10.14358/PERS.72.6.665.
- [48] Amy V. Uhrin and Philip A. Townsend. "Improved seagrass mapping using linear spectral unmixing of aerial photographs". In: *Estuarine, Coastal and Shelf Science* 171 (2016), pp. 11–22. ISSN: 02727714. DOI: 10.1016/j.ecss.2016.01.021.
- [49] Kenneth Clarke et al. "Using hyperspectral imagery to investigate large-scale seagrass cover and genus distribution in a temperate coast". In: Scientific Reports 11.1 (2021), p. 4182. ISSN: 2045-2322. DOI: 10.1038/s41598-021-83728-6.
- [50] I. Colomina and P. Molina. "Unmanned aerial systems for photogrammetry and remote sensing: A review". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 92 (2014), pp. 79–97. ISSN: 09242716. DOI: 10.1016/j.isprsjprs.2014.02.013.
- [51] Yuri Taddia et al. "Multispectral UAV monitoring of submerged seaweed in shallow water". In: Applied Geomatics 12.S1 (2020), pp. 19–34. ISSN: 1866-9298. DOI: 10.1007/s12518-019-00270-x.
- [52] Jan Komárek, Tomáš Klouček, and Jiří Prošek. "The potential of Unmanned Aerial Systems: A tool towards precision classification of hard-to-distinguish vegetation types?" In: *International Journal of Applied Earth Observation and Geoinformation* 71.May (2018), pp. 9–19. ISSN: 1872826X. DOI: 10.1016/j.jag.2018.05.003.
- [53] Natasha K. Nahirnick et al. "Mapping with confidence; delineating seagrass habitats using Unoccupied Aerial Systems (UAS)". In: *Remote Sensing in Ecology and Conservation* 5.2 (2019), pp. 121–135. ISSN: 20563485. DOI: 10.1002/rse2.98.
- [54] Craig J. Brown et al. "Benthic habitat mapping: A review of progress towards improved understanding of the spatial ecology of the seafloor using acoustic techniques". In: *Estuarine*, *Coastal and Shelf Science* 92.3 (2011), pp. 502–520. ISSN: 02727714. DOI: 10.1016/j.ecss.2011.02.007.
- [55] Mustafa Umit Gumusay et al. "A review of seagrass detection, mapping and monitoring applications using acoustic systems". In: European Journal of Remote Sensing 52.1 (2019), pp. 1–29. ISSN: 22797254. DOI: 10.1080/22797254.2018.1544838.
- [56] A. Lefebvre et al. "Use of a high-resolution profiling sonar and a towed video camera to map a Zostera marina bed, Solent, UK". In: Estuarine, Coastal and Shelf Science 82.2 (2009), pp. 323– 334. ISSN: 02727714. DOI: 10.1016/j.ecss.2009.01.027.
- [57] Roberta Ferretti et al. "Machine learning methods for acoustic-based automatic Posidonia meadows detection by means of unmanned marine vehicles". In: OCEANS 2017 Aberdeen 2017-Octob (2017), pp. 1–6. DOI: 10.1109/OCEANSE.2017.8084721.

[58] Luba Reshitnyk et al. "Evaluation of WorldView-2 and acoustic remote sensing for mapping benthic habitats in temperate coastal Pacific waters". In: Remote Sensing of Environment 153 (2014), pp. 7–23. ISSN: 00344257. DOI: 10.1016/j.rse.2014.07.016.

- [59] Daphnie S. Galvez et al. "Detecting shifts of submarine sediment boundaries using side-scan mosaics and GIS analyses". In: *Marine Geology* 430.November 2019 (2020), p. 106343. ISSN: 00253227. DOI: 10.1016/j.margeo.2020.106343.
- [60] Monica Montefalcone et al. "Evaluating change in seagrass meadows: A time-framed comparison of Side Scan Sonar maps". In: *Aquatic Botany* 104.4 (2013), pp. 204–212. ISSN: 03043770. DOI: 10.1016/j.aquabot.2011.05.009.
- [61] Austin Greene et al. "Side scan sonar: A cost-efficient alternative method for measuring seagrass cover in shallow environments". In: *Estuarine*, *Coastal and Shelf Science* 207. November 2017 (2018), pp. 250–258. ISSN: 0272-7714. DOI: 10.1016/j.ecss.2018.04.017.
- [62] Teruhisa Komatsu et al. "Use of multi-beam sonar to map seagrass beds in Otsuchi Bay on the Sanriku Coast of Japan". In: Aquatic Living Resources 16.3 (2003), pp. 223–230. ISSN: 09907440. DOI: 10.1016/S0990-7440(03)00045-7.
- [63] Vanessa Lucieer et al. "User expectations for multibeam echo sounders backscatter strength datalooking back into the future". In: Marine Geophysical Research 39.1-2 (2018), pp. 23–40. ISSN: 15730581. DOI: 10.1007/s11001-017-9316-5.
- [64] Craig J. Brown and Philippe Blondel. "Developments in the application of multibeam sonar backscatter for seafloor habitat mapping". In: Applied Acoustics 70.10 (2009), pp. 1242–1247. ISSN: 0003682X. DOI: 10.1016/j.apacoust.2008.08.004.
- [65] Mariacristina Prampolini et al. "Habitat mapping of the Maltese continental shelf using acoustic textures and bathymetric analyses". In: *Estuarine, Coastal and Shelf Science* 207 (2018), pp. 483–498. ISSN: 02727714. DOI: 10.1016/j.ecss.2017.06.002.
- [66] Alessandro Bosman et al. "Short-term evolution of Po della Pila delta lobe from time lapse high-resolution multibeam bathymetry (2013–2016)". In: Estuarine, Coastal and Shelf Science 233.November 2019 (2020). ISSN: 02727714. DOI: 10.1016/j.ecss.2019.106533.
- [67] Lukasz Janowski et al. "Spatial and temporal changes of tidal inlet using object-based image analysis of multibeam echosounder measurements: A case from the Lagoon of Venice, Italy". In: Remote Sensing 12.13 (2020). ISSN: 20724292. DOI: 10.3390/rs12132117.
- [68] Daniel Ierodiaconou et al. "Combining pixel and object based image analysis of ultra-high resolution multibeam bathymetry and backscatter for habitat mapping in shallow marine waters". In:

 *Marine Geophysical Research 39.1-2 (2018), pp. 271–288. ISSN: 15730581. DOI: 10.1007/s11001-017-9338-z.
- [69] G. Montereale Gavazzi et al. "Evaluation of seabed mapping methods for fine-scale classification of extremely shallow benthic habitats Application to the Venice Lagoon, Italy". In: *Estuarine*, Coastal and Shelf Science 170 (2016), pp. 45–60. ISSN: 02727714. DOI: 10.1016/j.ecss.2015.12.014.
- [70] Karolina Trzcinska et al. "Spectral features of dual-frequency multibeam echosounder data for benthic habitat mapping". In: Marine Geology 427. November 2019 (2020), p. 106239. ISSN: 00253227. DOI: 10.1016/j.margeo.2020.106239.
- [71] Lukasz Janowski et al. "Nearshore benthic habitat mapping based on multi-frequency, multibeam echosounder data using a combined object-based approach: A case study from the Rowy Site in the Southern Baltic Sea". In: *Remote Sensing* 10.12 (2018), pp. 1–21. ISSN: 20724292. DOI: 10.3390/rs10121983.

[72] Timo C. Gaida et al. "A multispectral bayesian classification method for increased acoustic discrimination of seabed sediments using multi-frequency multibeam backscatter data". In: *Geosciences* (Switzerland) 8.12 (2018). ISSN: 20763263. DOI: 10.3390/geosciences8120455.

- [73] Bryan Costa. "Multispectral Acoustic Backscatter: How Useful Is it for Marine Habitat Mapping and Management?" In: *Journal of Coastal Research* 35.5 (2019), pp. 1062–1079. ISSN: 15515036. DOI: 10.2112/JCOASTRES-D-18-00103.1.
- [74] CB Tuit and AD Wait. "A review of marine sediment sampling methods". In: *Environmental Forensics* 21.3-4 (2020), pp. 291–309. ISSN: 1527-5922. DOI: 10.1080/15275922.2020.1771630.
- [75] D. Scaradozzi et al. "Innovative technology for studying growth areas of Posidonia oceanica". In: 2009 IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems. IEEE, 2009, pp. 71–75. ISBN: 978-1-4244-4847-0. DOI: 10.1109/EESMS.2009.5341312.
- [76] Matthew Johnson-Roberson et al. "High-Resolution Underwater Robotic Vision-Based Mapping and Three-Dimensional Reconstruction for Archaeology". In: *Journal of Field Robotics* 34.4 (2017), pp. 625–643. ISSN: 15564967. DOI: 10.1002/rob.21658.
- [77] Oscar Pizarro et al. "A simple, fast, and repeatable survey method for underwater visual 3D benthic mapping and monitoring". In: *Ecology and Evolution* 7.6 (2017), pp. 1770–1782. ISSN: 20457758. DOI: 10.1002/ece3.2701.
- [78] S. F. Rende et al. "Advances in micro-cartography: A two-dimensional photo mosaicing technique for seagrass monitoring". In: *Estuarine, Coastal and Shelf Science* 167 (2015), pp. 475–486. ISSN: 02727714. DOI: 10.1016/j.ecss.2015.10.029.
- [79] Alexander R. Ilich et al. "Integrating towed underwater video and multibeam acoustics for marine benthic habitat mapping and fish population estimation". In: *Geosciences (Switzerland)* 11.4 (2021), pp. 1–40. ISSN: 20763263. DOI: 10.3390/geosciences11040176.
- [80] Rozaimi Che Hasan, Daniel Ierodiaconou, and Jacquomo Monk. "Evaluation of four supervised learning methods for benthic habitat mapping using backscatter from multi-beam sonar". In: Remote Sensing 4.11 (2012), pp. 3427–3443. ISSN: 20724292. DOI: 10.3390/rs4113427.
- [81] Peter I. Macreadie et al. "Eyes in the sea: Unlocking the mysteries of the ocean using industrial, remotely operated vehicles (ROVs)". In: Science of the Total Environment 634 (2018), pp. 1077–1091. ISSN: 18791026. DOI: 10.1016/j.scitotenv.2018.04.049.
- [82] Katleen Robert et al. "New approaches to high-resolution mapping of marine vertical structures". In: Scientific Reports 7.1 (2017), pp. 1–14. ISSN: 20452322. DOI: 10.1038/s41598-017-09382-z.
- [83] Liam Paull et al. "AUV Navigation and Localization: A Review". In: IEEE J. Ocean. Eng. 39.1 (2014), pp. 131–149. ISSN: 0364-9059. DOI: 10.1109/JOE.2013.2278891.
- [84] Mitch Bryson et al. "True Color Correction of Autonomous Underwater Vehicle Imagery". In: Journal of Field Robotics 33.6 (2016), pp. 853–874. ISSN: 15564959. DOI: 10.1002/rob.21638.
- [85] S. D. Ling et al. "Stereo-imaging AUV detects trends in sea urchin abundance on deep overgrazed reefs". In: *Limnology and Oceanography: Methods* 14.5 (2016), pp. 293–304. ISSN: 15415856. DOI: 10.1002/lom3.10089.
- [86] Daniel L. Bongiorno et al. "Coregistered Hyperspectral and Stereo Image Seafloor Mapping from an Autonomous Underwater Vehicle". In: *Journal of Field Robotics* 35.3 (2018), pp. 312–329. ISSN: 15564967. DOI: 10.1002/rob.21713. arXiv: 10.1.1.91.5767.
- [87] Ayoung Kim and Ryan M. Eustice. "Active visual SLAM for robotic area coverage: Theory and experiment". In: *The International Journal of Robotics Research* 34.4-5 (2015), pp. 457–475. ISSN: 0278-3649. DOI: 10.1177/0278364914547893.

[88] Stephen M. Chaves et al. "Opportunistic sampling-based active visual SLAM for underwater inspection". In: *Autonomous Robots* 40.7 (2016), pp. 1245–1265. ISSN: 0929-5593. DOI: 10.1007/s10514-016-9597-6.

- [89] Geoffrey A. Hollinger and Gaurav S. Sukhatme. "Sampling-based robotic information gathering algorithms". In: The International Journal of Robotics Research 33.9 (2014), pp. 1271–1287. ISSN: 0278-3649. DOI: 10.1177/0278364914533443.
- [90] Jnaneshwar Das et al. "Data-driven robotic sampling for marine ecosystem monitoring". In: Int. J. Rob. Res. 34.12 (2015), pp. 1435–1452. ISSN: 0278-3649. DOI: 10.1177/0278364915587723.
- [91] Gregory Hitz et al. "Fully autonomous focused exploration for robotic environmental monitoring". In: 2014 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2014, pp. 2658–2664. ISBN: 978-1-4799-3685-4. DOI: 10.1109/ICRA.2014.6907240.
- [92] Jeffrey A. Caley and Geoffrey A. Hollinger. "Data-driven comparison of spatio-temporal monitoring techniques". In: *OCEANS 2015 MTS/IEEE Washington*. IEEE, 2016, pp. 1–7. ISBN: 9780933957435. DOI: 10.23919/oceans.2015.7401988.
- [93] Jonathan Binney and Gaurav S. Sukhatme. "Branch and bound for informative path planning". In: Proceedings - IEEE International Conference on Robotics and Automation (2012), pp. 2147—2154. ISSN: 10504729. DOI: 10.1109/ICRA.2012.6224902.
- [94] Alberto Candela et al. "Planetary robotic exploration driven by science hypotheses for geologic mapping". In: IEEE International Conference on Intelligent Robots and Systems 2017-Septe (2017), pp. 3811–3818. ISSN: 21530866. DOI: 10.1109/IROS.2017.8206231.
- [95] Nicholas R. J. Lawrance, Jen Jen Chung, and Geoffrey A. Hollinger. "Fast Marching Adaptive Sampling". In: *IEEE Robotics and Automation Letters* 2.2 (2017), pp. 696–703. ISSN: 2377-3766. DOI: 10.1109/LRA.2017.2651148.
- [96] Ryan N Smith et al. "Adaptive Path Planning for Tracking Ocean Fronts with an Autonomous Underwater Vehicle". In: Proceedings, International Symposium on Experimental Robotics. Vol. 109. 2016, pp. 761–775. DOI: 10.1007/978-3-319-23778-7_50.
- [97] Lionel Heng et al. "Efficient visual exploration and coverage with a micro aerial vehicle in unknown environments". In: *Proceedings IEEE International Conference on Robotics and Automation* 2015-June.June (2015), pp. 1071–1078. ISSN: 10504729. DOI: 10.1109/ICRA.2015.7139309.
- [98] Valentina Karolj et al. "An integrated strategy for autonomous exploration of spatial processes in unknown environments". In: Sensors (Switzerland) 20.13 (2020), pp. 1–27. ISSN: 14248220. DOI: 10.3390/s20133663.
- [99] Maani Ghaffari Jadidi, Jaime Valls Miro, and Gamini Dissanayake. "Gaussian processes autonomous mapping and exploration for range-sensing mobile robots". In: *Autonomous Robots* 42.2 (2017), pp. 1–18. ISSN: 15737527. DOI: 10.1007/s10514-017-9668-3.
- [100] Maani Ghaffari Jadidi, Jaime Valls Miro, and Gamini Dissanayake. "Sampling-based incremental information gathering with applications to robotic exploration and environmental monitoring". In: The International Journal of Robotics Research 38.6 (2019), pp. 658–685. ISSN: 0278-3649. DOI: 10.1177/0278364919844575. arXiv: 1607.01883.
- [101] Dushyant Rao et al. "Multimodal information-theoretic measures for autonomous exploration". In: 2016 IEEE International Conference on Robotics and Automation (ICRA). Vol. 2016-June. IEEE, 2016, pp. 4230–4237. ISBN: 978-1-4673-8026-3. DOI: 10.1109/ICRA.2016.7487618.
- [102] Suhit Kodgule, Alberto Candela, and David Wettergreen. "Non-myopic Planetary Exploration Combining In Situ and Remote Measurements". In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2019, pp. 536–543. ISBN: 978-1-7281-4004-9. DOI: 10.1109/IROS40897.2019.8967769. arXiv: 1904.12255.

[103] Tal Regev and Vadim Indelman. "Decentralized multi-robot belief space planning in unknown environments via identification and efficient re-evaluation of impacted paths". In: *Autonomous Robots* 42.4 (2018), pp. 691–713. ISSN: 15737527. DOI: 10.1007/s10514-017-9659-4.

- [104] Alberto Viseras, Zhe Xu, and Luis Merino. "Distributed multi-robot information gathering under spatio-temporal inter-robot constraints". In: Sensors (Switzerland) 20.2 (2020), pp. 1–25. ISSN: 14248220. DOI: 10.3390/s20020484.
- [105] Graeme Best et al. "Dec-MCTS: Decentralized planning for multi-robot active perception". In: International Journal of Robotics Research 38.2-3 (2019), pp. 316–337. ISSN: 17413176. DOI: 10.1177/0278364918755924.
- [106] Kai Chieh Ma et al. "Data-driven learning and planning for environmental sampling". In: *Journal of Field Robotics* 35.5 (2018), pp. 643–661. ISSN: 15564967. DOI: 10.1002/rob.21767. arXiv: 1702.01848.
- [107] Gregory Hitz et al. "Adaptive continuous-space informative path planning for online environmental monitoring". In: *Journal of Field Robotics* 34.8 (2017), pp. 1427–1449. ISSN: 15564959. DOI: 10.1002/rob.21722.
- [108] Marija Popović et al. "An informative path planning framework for UAV-based terrain monitoring". In: Autonomous Robots 44.6 (2020), pp. 889–911. ISSN: 15737527. DOI: 10.1007/s10514-020-09903-2. arXiv: 1809.03870.
- [109] Carl Edward Rasmussen and Christopher K I Williams. Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning). The MIT Press, 2005. ISBN: 026218253X.
- [110] Andreas Krause, Ajit Singh, and Carlos Guestrin. "Near-Optimal Sensor Placements in Gaussian Processes: Theory, Efficient Algorithms and Empirical Studies". In: *Journal of Machine Learning Research* 9 (2008), pp. 235–284. ISSN: 15324435. DOI: 10.1145/1102351.1102385.
- [111] Eric Guerrero-Font et al. "Sparse Gaussian process for online seagrass semantic mapping". In: Expert Systems with Applications 170.December 2020 (2021), p. 114478. ISSN: 09574174. DOI: 10.1016/j.eswa.2020.114478.
- [112] Geoffrey A. Hollinger, Arvind A. Pereira, and Gaurav S. Sukhatme. "Learning uncertainty models for reliable operation of Autonomous Underwater Vehicles". In: 2013 IEEE International Conference on Robotics and Automation. IEEE, 2013, pp. 5593–5599. ISBN: 978-1-4673-5643-5. DOI: 10.1109/ICRA.2013.6631380.
- [113] Alberto Viseras, Dmitriy Shutin, and Luis Merino. "Robotic Active Information Gathering for Spatial Field Reconstruction with Rapidly-Exploring Random Trees and Online Learning of Gaussian Processes". In: Sensors 19.5 (2019), p. 1016. ISSN: 1424-8220. DOI: 10.3390/s19051016.
- [114] Andreas Krause and Carlos Guestrin. "Submodularity and its applications in optimized information gathering". In: ACM Transactions on Intelligent Systems and Technology 2.4 (2011), pp. 1–20. ISSN: 2157-6904. DOI: 10.1145/1989734.1989736.
- [115] Roman Marchant and Fabio Ramos. "Bayesian optimisation for Intelligent Environmental Monitoring". In: *IEEE International Conference on Intelligent Robots and Systems* (2012), pp. 2242–2249. ISSN: 21530858. DOI: 10.1109/IROS.2012.6385653.
- [116] Oliver M. Cliff et al. "Online localization of radio-tagged wildlife with an autonomous aerial robot system". In: *Robotics: Science and Systems* 11 (2015). ISSN: 2330765X. DOI: 10.15607/RSS.2015. XI.042.
- [117] David R. Thompson, David S. Wettergreen, and Francisco J. Calderóon Peralta. "Autonomous science during large-scale robotic survey". In: *Journal of Field Robotics* 28.4 (2011), pp. 542–564. ISSN: 15564959. DOI: 10.1002/rob.20391. arXiv: 10.1.1.91.5767.

[118] Kwangjin Yang, Seng Keat Gan, and Salah Sukkarieh. "A Gaussian process-based RRT planner for the exploration of an unknown and cluttered environment with a UAV". In: Advanced Robotics 27.6 (2013), pp. 431–443. ISSN: 01691864. DOI: 10.1080/01691864.2013.756386.

- [119] Lukas Schmid et al. "An Efficient Sampling-Based Method for Online Informative Path Planning in Unknown Environments". In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 1500–1507. ISSN: 23773766. DOI: 10.1109/LRA.2020.2969191. arXiv: 1909.09548.
- [120] Chengke Xiong et al. "Rapidly-exploring adaptive sampling tree*: A sample-based path-planning algorithm for unmanned marine vehicles information gathering in variable ocean environments". In: Sensors (Switzerland) 20.9 (2020). ISSN: 14248220. DOI: 10.3390/s20092515.
- [121] L.E. Kavraki et al. "Probabilistic roadmaps for path planning in high-dimensional configuration spaces". In: *IEEE Transactions on Robotics and Automation* 12.4 (1996), pp. 566–580. ISSN: 1042296X. DOI: 10.1109/70.508439. arXiv: arXiv:1011.1669v3.
- [122] Steven M. LaValle. Planning Algorithms. Vol. s1-9. 48. Cambridge: Cambridge University Press, 2006, pp. 693–695. ISBN: 9780511546877. DOI: 10.1017/CB09780511546877.
- [123] Sertac Karaman and Emilio Frazzoli. "Sampling-based Algorithms for Optimal Motion Planning".
 In: (2011), pp. 1–76. ISSN: 0278-3649. DOI: 10.1177/0278364911406761. arXiv: 1105.1186.
- [124] Paul Stankiewicz, Yew T. Tan, and Marin Kobilarov. "Adaptive sampling with an autonomous underwater vehicle in static marine environments". In: *Journal of Field Robotics* June (2020). ISSN: 15564967. DOI: 10.1002/rob.22005.
- [125] F. S. Rende et al. "Pilot Application of 3D Underwater Imaging Techniques for Mapping Posidonia Oceanica (L.) Delile Meadows". In: ISPRS International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XL-5/W5 (2015), pp. 177–181. ISSN: 2194-9034. DOI: 10.5194/isprsarchives-XL-5-W5-177-2015.
- [126] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation". In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2015, pp. 3431–3440. ISBN: 978-1-4673-6964-0. DOI: 10.1109/CVPR.2015.7298965.
- [127] Miguel Martin-Abadal et al. "Deep Semantic Segmentation in an AUV for Online Posidonia Oceanica Meadows Identification". In: *IEEE Access* 6 (2018), pp. 60956–60967. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2018.2875412.
- [128] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Ed. by Nassir Navab et al. Vol. 9351. Lecture Notes in Computer Science Cvd. Cham: Springer International Publishing, 2015, pp. 234–241. ISBN: 9783319245737. DOI: 10.1007/978-3-319-24574-4_28.
- [129] Vijay Badrinarayanan et al. "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation". In: 39.12 (2017), pp. 2481–2495.
- [130] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". 2014. arXiv: 1409.1556.
- [131] Vincent Dumoulin and Francesco Visin. "A guide to convolution arithmetic for deep learning". 2016. arXiv: 1603.07285.
- [132] Martín Abadi et al. "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems". 2015. arXiv: 1603.04467.
- [133] Thomas Moore and Daniel Stouch. "A generalized extended Kalman filter implementation for the robot operating system". In: *Advances in Intelligent Systems and Computing* 302 (2016), pp. 335–348. ISSN: 21945357. DOI: 10.1007/978-3-319-08338-4 25.

[134] Byron M Yu, Krishna V Shenoy, and Maneesh Sahani. "Derivation of Extended Kalman Filtering and Smoothing Equations". 2004.

- [135] Carl Edward Rasmussen and Christopher K. I. Williams. Gaussian processes for machine learning. Adaptive computation and machine learning. MIT Press, 2006, pp. I–XVIII, 1–248. ISBN: 026218253X.
- [136] Alexander G. de G. Matthews et al. "GPflow: A Gaussian Process Library using TensorFlow". In: Journal of Machine Learning Research 18.40 (2017), pp. 1–6.
- [137] Michalis K Titsias. "Variational Learning of Inducing Variables in Sparse Gaussian Processes". In: Journal of Machine Learning Research - Proceedings Track 5 (2009), pp. 567–574.
- [138] James Hensman et al. "MCMC for Variationally Sparse Gaussian Processes". In: (2015). arXiv: 1506.04000.
- [139] James Hensman and Alexander G De G Matthews. "Scalable Variational Gaussian Process Classification". In: 18th International Conference on Artificial Intelligence and Statistics (AISTATS). Vol. 38, 2015.
- [140] Morgan Quigley et al. "ROS: an open-source Robot Operating System". In: *Icra* 3.Figure 1 (2009), p. 5. ISSN: 0165022X. DOI: http://www.willowgarage.com/papers/ros-open-source-robot-operating-system.
- [141] Marc Carreras et al. "Sparus II AUV: A Hovering Vehicle for Seabed Inspection". In: *IEEE Journal of Oceanic Engineering* 43.2 (2018), pp. 344–355. ISSN: 0364-9059. DOI: 10.1109/JOE.2018. 2792278.
- [142] Antoni Martorell Torres et al. "Xiroi ASV: a Modular Autonomous Surface Vehicle to Link Communications". In: *IFAC-PapersOnLine* 51.29 (2018), pp. 147–152. ISSN: 2405-8963. DOI: 10.1016/j.ifacol.2018.09.484.
- [143] Eric Guerrero-Font et al. "USBL Integration and Assessment in a Multisensor Navigation Approach for AUVs". In: *IFAC-PapersOnLine* 50.1 (2017), pp. 7905–7910. ISSN: 24058963. DOI: 10.1016/j.ifacol.2017.08.754.
- [144] David Powers. "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation". In: *International Journal of Machine Learning Technology* 2 (2011), pp. 37–63. DOI: 10.9735/2229-3981.
- [145] Richard S. Sutton and Andrew G. Barto. *Reinforcement Leaning*. 2018, p. 481. ISBN: 9780262039246. DOI: 006.3/1--dc23.
- [146] Stanford Artificial Intelligence Laboratory et al. Robotic Operating System. Version ROS Melodic Morenia. May 23, 2018.
- [147] Alexander Tiderko. Multimaster FKIE. https://github.com/fkie/multimaster_fkie. 2019.
- [148] Eric Guerrero Font. Evologics ROS Sync. https://github.com/srv/evologics_ros_sync. 2022.
- [149] Valerio De Carolis. *Evologics Driver*. https://github.com/oceansystemslab/evologics_driver. 2016.
- [150] T. Moore and D. Stouch. "A Generalized Extended Kalman Filter Implementation for the Robot Operating System". In: *Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13)*. Springer, 2014.
- [151] Takafumi Taketomi, Hideaki Uchiyama, and Sei Ikeda. "Visual SLAM algorithms: A survey from 2010 to 2016". In: *IPSJ Transactions on Computer Vision and Applications* 9 (2017). ISSN: 18826695. DOI: 10.1186/s41074-017-0027-2.
- [152] Jie Li et al. "Pose-Graph SLAM Using Forward-Looking Sonar". In: IEEE Robotics and Automation Letters 3.3 (2018), pp. 2330–2337. ISSN: 23773766. DOI: 10.1109/LRA.2018.2809510.

[153] Evan Shelhamer, Jonathan Long, and Trevor Darrell. "Fully Convolutional Networks for Semantic Segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.4 (2017), pp. 640–651. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2016.2572683. arXiv: 1411.4038.

- [154] Narcis Palomeras et al. "COLA2: A Control Architecture for AUVs". In: *IEEE Journal of Oceanic Engineering* 37.4 (2012), pp. 695–716. ISSN: 03649059. DOI: 10.1109/JOE.2012.2205638.
- [155] Thor I. Fossen and Kristin Y. Pettersen. "On uniform semiglobal exponential stability (USGES) of proportional line-of-sight guidance laws". In: *Automatica* 50.11 (2014), pp. 2912–2917. ISSN: 00051098. DOI: 10.1016/j.automatica.2014.10.018.
- [156] Jon Louis Bentley. "Multidimensional Binary Search Trees Used for Associative Searching". In: Communications of the ACM (1975). ISSN: 00010782. arXiv: arXiv:1011.1669v3.
- [157] Patryk Cieślak. "Stonefish: An Advanced Open-Source Simulation Tool Designed for Marine Robotics, With a ROS Interface". In: *OCEANS 2019 Marseille*. 2019. DOI: 10.1109/OCEANSE. 2019.8867434.

