UNIVERSITÀ
DEGLI STUDI
FIRENZE

Sensor-driven autonomous inspections and CNN-based Automatic Target Recognition strategies towards underwater full autonomy

A dissertation submitted in partial fulfillment of the requirements for the Degree of Doctor of Philosophy in

Industrial Engineering
Scientific Disciplinary Sector ING-IND/13

Candidate
Leonardo Zacchini

Advisor
Prof. Benedetto Allotta

Co-advisor
Dr. Alessandro Ridolfi

Coordinator
Prof. Giampaolo Manfrida

Department of Industrial Engineering
University of Florence
Florence, Italy

XXXIV Ph.D. Cycle - Years 2018/2021

"Your son is trying, but it's not his fault; he does not comprehend math.
He has no aptitude for science and math.
He should go to another school or, better, look for a job."

To my family.

Tons of people have my gratitude for their contribution to my experience at the University of Florence during the last three years.

Firstly, I would like to express my gratitude to Prof. Benedetto Allotta that introduced me to the world of robotics. Thanks to him, I had the opportunity to work in this field for three years, making a long-lasting wish come true.

Special thanks go to my co-advisor, Dr. Alessandro Ridolfi, for his advice, guidance, and time. I would like to sincerely thank you for trusting me during these years and for your precious help in achieving the results reported in this thesis.

"Teamwork makes the dream work!"; a huge thanks goes to my lab mates. The results of this thesis wouldn't have been possible without them. More friends than just colleagues, they have made these three years stimulating. I want to thank them for their support, for all they have taught me, and for all the fun we have had in these years.

My heartfelt thanks go to my parents and family, who supported and encouraged me during this challenging period. Mom and Dad are the ones who made it possible to start and conclude this journey.

Thanks to my girlfriend who has supported me daily during these years; she has shared with me the joys and the tough times.

Finally, I would like to thank all my friends, who have made these years great and amusing.

Thanks to all the people mentioned here for shaping the person I am!

Leonardo Zacchini. University of Florence, October 2021.

Autonomous Underwater Vehicles (AUVs) have become fundamental tools for marine scientists and industries to explore and monitor underwater areas. Non-predictable environmental conditions and sensor acquisitions make the design of AUV surveys challenging even for expert operators. Multiple attempts are required, and the collected data quality is not guaranteed: the AUV usually passively stores the sensors' acquisitions that are then analyzed offline after its recovery by human operators in charge of identifying and localizing the so-called Objects of Potential Interest (OPIs). When it comes to acoustic images, the lack of features and low resolution make this task even more challenging. As a consequence of these statements, the marine community has sought robots able to meaningfully perceive and model the surroundings and autonomously conduct the assigned task, which are the foundations of fully autonomous vehicles. In this work, these topics have been investigated.

To allow a compact and lightweight AUV to gather knowledge of the surroundings, an Automatic Target Recognition (ATR) strategy based on modern Convolutional Neural Networks (CNNs) for onboard online applications was developed. The ATR methodology was used to identify and localize potential targets of interest in Forward-Looking SONAR (FLS) imagery.

Then, to avoid using pre-planned surveys and make an AUV actively considering the acquired data online, this thesis presents a probabilistic framework for FLS-driven seabed inspections. The realized sensor-driven Receding-Horizon Coverage Approach (RHCA) endows the AUV with the ability to autonomously conducting the survey and ensures adequate coverage of the target area. A Rapidly-exploring Random Tree (RRT) inspired view planning algorithm for underwater inspections was designed. Then, advancements for enhancing the performance of the algorithm have been carried out. In particular, a novel informed tree expansion methodology for guiding the vehicle towards the non-explored regions is proposed. Thanks to this solution based on the Kernel Density Estimation technique, the AUV learns the distribution of the discovered map. In addition, to explore other view planning strategies, a preliminary investigation about the exploitation of a Randomized Model Predictive Control (RMPC) approach for conducting autonomous seabed inspections is reported.

Finally, the proposed ATR methodology and the RHCA have been combined to realize a target-aware planning solution for autonomously inspecting an area of interest. A probabilistic semantic map that includes the knowledge about the presence of the OPIs is created and updated by using the

ATR findings. The semantic map enables the view planning algorithm to generate paths that cover the area of interest and simultaneously reduces the target localization uncertainty. Therefore, this methodology allows an AUV to meaningfully perceive and model the surroundings and autonomously conduct inspections surveys.

The solutions proposed in this thesis have been firstly validated with realistic simulations made by means of the Unmanned Underwater Vehicle Simulator (UUV Simulator), where a dynamic model of FeelHippo AUV was implemented. Moreover, the ATR strategy, the RHCA framework, and the proposed view planning advances have been tested in real experimental campaigns at sea.

Contents

Acronyms

ADCP Acoustic Doppler Current Profiler

AHRS Attitude and Heading Reference System

AI Artificial Intelligence

API Application Programming Interface

AR Augmented Reality

ATR Automatic Target Recognition

AUV Autonomous Underwater Vehicle

BIG Branch Information Gain

BVP Boundary Value Problem

CNN Convolutional Neural Network

COCO Microsoft Common Objects in Context

CPP Coverage Path Planning

CV Computer Vision

DIEF Department of Industrial Engineering

DL Deep Learning

DNN Deep Neural Network

DOFs Degrees Of Freedom

DS Depth Sensor

DVL Doppler Velocity Log

Faster R-CNN Faster Region-based Convolutional Neural Network

FLS Forward-Looking SOund NAvigation and Ranging (SONAR)

FOG Fiber Optic Gyrosope

FoV Field of View

FP Frontier Points

FPN Feature Pyramid Network

fps frames per second

GPS Global Positioning System

GUI Graphical User Interface

IG Information Gain

IMU Inertial Measurement Unit

IoU Intersection over Union

ISM Inverse Sensor Model

KDE Kernel Density Estimation

LiDAR Light Detection and Ranging

mAP mean Average Precision

Mask R-CNN Mask Region-based Convolutional Neural Network

MAV Micro Aerial Vehicle

MBE MultiBeam Echosounder

MCM Mine CounterMeasure

ML Machine Learning

MSIS Mechanically Scanned Imaging SONAR

NBV Next-Best Viewpoint

NN Neural Network

OMPL Open Motion Planning Library

OPI Object of Potential Interest

pRRT Parallel Rapidly-exploring Random Tree

R-FCN Region-based Fully Convolutional Network

RHCA Receding Horizon Coverage Approach

RMPC Randomized Model Predictive Control

ROI Regions Of Interest

ROS Robot Operating System

ROV Remotely Operated Vehicle

RPN Region Proposal Network

RRT Rapidly-exploring Random Tree

RS Random Search

SBP Sub Bottom Profiler

SGD Stochastic Gradient Descent

SLAM Simultaneous Localization And Mapping

SNAME Society of Naval Architects and Marine Engineers

SONAR SOund NAvigation and Ranging

SSD Single Shot Multibox Detector

SSS SideScan SONAR

TSP Traveling Salesman Problem

UAV Unmanned Aerial Vehicle

UGV Unmanned Ground Vehicle

UNIFI University of Florence

UUV Simulator Unmanned Underwater Vehicle Simulator

VI Volumetric Information

WGS84 World Geodetic System

YOLO You Only Look Once

Chapter 1

Introduction and thesis motivation

THE last two decades of mobile robotics developments have been led by the increasing demand coming from scientists, researchers, and industries for robots able to perform complex tasks and acquire data in hazardous scenarios. In this context, the recent advancement in autonomous vehicles aims to develop increasingly intelligent systems capable of interacting with the surrounding environment and independently deciding the best actions to fulfill specific tasks. To this end, modern robotics tries to integrate advanced planning methodologies, environment perceiving and modelling strategies, and Artificial Intelligence (AI) concepts and technologies. In particular, AI is currently used in robotic systems for different purposes, such as making autonomous decisions, planning paths, and extensive data processing, fields where excellent results are being achieved.

For what concerns the marine environment, since the demanded tasks of Autonomous Underwater Vehicles (AUVs) have become more and more challenging [1] [2], researchers and scientists are following the tide of change and are pushing the boundaries of AUVs capabilities by integrating cutting-edge technologies. Indeed, autonomous inspection [3] and intervention [4] strategies for underwater installations, exploration planning solutions [5], and autonomous coverage approaches [6] [7], have become essential tools to execute demanding and hazardous subsea operations.

To accomplish such complex tasks, perceiving and understanding the environment is a fundamental hierarchical step. To this end, AUVs can be equipped with several payload sensors, such as optical cameras, Multi-Beam Echosounders (MBEs), SideScan SONARs (SSSs), Forward-Looking SONARs (FLSs), Sub Bottom Profilers (SBPs), and so on. However, collected raw data could not be sufficient for meaningfully understanding the environment. Thus, such data shall be processed online to identify obstacles, objects of interest, or hazardous targets, to mention a few. For exploring unknown areas as well as navigating into highly unstructured environments, free and occupied areas must be carefully identified [8] and 3D occupancy maps must be created using Mechanically Scanned Imaging SONARs (MSISs), optical cameras, and FLSs [5] [9]. Regarding autonomous interventions, the AUV shall correctly detect and localize objects of interest to interact with the external environment. When dealing with structured areas, such as water tanks, Augmented Reality (AR) markers and Computer Vision (CV) techniques represent a simple and extremely effective solution [4]. Nevertheless, robots have to face frequently non-structured and unknown regions. For instance, in sea mining explorations, the AUV performs optical surveys to identify nodules and stones. Once a nodule is detected, a visual-guided landing maneuver to collect the object is performed. In this context, since environmental and light conditions change continuously and cannot be foretold, the performance of CV techniques are limited. Hence, modern Convolutional Neural Networks (CNNs) shall be exploited to achieve satisfying results [10].

Contemporaneously, marine vehicles have become reliable and essential tools for science users (e.g., archaeologists, oceanographers, biologists) and navy personnel for collecting high-quality data to analyze and study an area of interest.

In marine science investigations, field specialists have sought systems to expand their ability to discover and observe the oceans, and AUVs have met their requirements [11]. Consequently, the scientific community has extensively used AUVs endowed with different payload sets in various applications in the last years [12]. In [13], the authors used an AUV to collect MBE data to characterizing the active tectonic and gravitational deformation of the northern Ligurian margin, Italy. To identify mechanisms related to the detachment fault denudation at the seafloor and investigate the relation between the mass wasting and the tectonic extension, the authors in [14] used the collected micro bathymetric data and geologic samples acquired by an AUV and a Remotely Operated Vehicle (ROV), respectively. MBE data acquired by the Eagle Ray AUV was used in the research work presented in [15]. In detail, the MBE dataset was used to locate and map the hydrocarbon seeps that are evident in the MBE measurements in Green Canyon Block 600 in the Gulf of Mexico. For deep ocean seafloor investigations, 6,000 m depth rated AUVs acquired MBE, SSS, and SBP data as reported in [16] and [17]. The collected data allowed geoscientists to analyze the seafloor morphology, the bathymetric changes associated with sediment transport, gravitational driven failure associated features, and ocean bottom-current activities. Besides, marine biologists were able to investigate deep-ocean benthic habitats. Similarly, in [18], the distribution of the benthic communities of the Ningaloo Marine Park, Australia, was analyzed using geomorphological measurements provided by an MBE device, environmental data (such as the water temperature, salinity, to mention a few), and optical images. The exploited dataset was acquired in several surveys conducted by different AUVs. In [19], an AUV able to acquire high-resolution photographic, bathymetric, and sub-bottom data was used to perform a seismo-acoustic characterization. An optical delineation campaign of near-shore benthic habitat was reported in [20]. The exploited AUV was equipped with multi-spectral radiometers and optical cameras; the results demonstrated AUVs' ability to map littoral habitats at high resolution and proved their fundamental role in collecting high-quality data for science users. Marine vehicles were also employed for water quality monitoring and observation of fish behavior in net cage fish farming [21], where collecting data over large areas is of utmost importance.

In seabed inspections, AUVs are used to collect optical as well as acoustic images, using cameras and imaging sonars. Then, the data could be used for underwater navigation [22] and surveillance purposes [23], where image processing techniques and modern Deep Learning (DL) methodologies can accurately find targets of interest (see [24], [25], and [26]), and for archaeological investigations, where optical [27] and acoustic reconstructions [28] emerged as an essential tool to correctly classifying historical finds. These tasks are generally performed by exploiting optical sensors; however, optical cameras are affected by water turbidity and lighting conditions, and gathering satisfactory images does arise as a non-trivial task, not feasible in several scenarios. As a consequence, e.g., SSSs, as well as FLSs, are commonly favored to carry out inspection and exploration tasks; in fact, sonars are not influenced by illumination conditions and can provide high-range data. In particular, FLSs can synthesize satisfactory resolution images, more detailed than SSSs, but at shorter distances. Nevertheless, the high noise and the lack of features make sonar images hard to interpret by using conventional image processing techniques. De facto, a human operator is usually in charge of analyzing the thousands of acquired images to identify the so-called Objects of Potential Interest (OPIs). Once identified, the targets shall be localized, the AUV navigation data and the sonar characteristics are needed. Therefore, Automatic Target Recognition (ATR) strategies that detect and localize OPIs in FLS imagery are fundamental tools that could help human operators in this demanding task. In this context, cutting-edge DL techniques, which have become the state of the art in the classification and object detection tasks [29], are being investigated in marine ATR applications, showing outstanding results [25] [30].

The aforementioned case studies' analysis highlights the science users (e.g., archaeologists, geologists, oceanographers, biologists) and navy personnel necessity for robots for collecting high-quality data to analyze and study an area of interest. Science users cooperate with technicians to plan AUV surveys, often using lawnmower or zig-zag paths, to acquire well distributed and representative data. Generally speaking, despite the different scopes and sensors used, the surveys are designed so that the more the area is covered, the better the data collected. Thus, the better the area characterization.

Nowadays, mostly AUVs passively store the data that is then analyzed offline after their recovery, meaning that exteroceptive sensor feedbacks are not used during the mission. Since environmental conditions cannot be forecasted, and the sensor characteristics affect the quantity and the quality of the collected data, it implies many mission replanning and repeated attempts, increasing the total cost of exploration and monitoring campaigns. This is especially the case of FLS and SSS based seabed inspections. The seafloor morphology and composition, as well as object shapes and positions, affect the performance of such acoustic devices. Consequently, designing offline AUV paths that ensure satisfying coverage of the area is challenging and requires time and experienced operators.

Coverage Path Planning (CPP) and exploration strategies represent an excellent solution. Sensor-driven algorithms can actively consider sensor feedbacks and environmental information; they can monitor the quality and quantity of acquired data and guarantee adequate coverage levels or exploration goals, avoiding repeated missions. Since the problem of computing a path that optimally covers an area of interest with a sensor is common to many robotic domains, several CPP and exploration methodologies for Unmanned Ground Vehicles (UGVs), Micro Aerial Vehicles (MAVs), AUVs have been proposed in the last years. Generally speaking, the authors developed solutions to enable the robot to calculate the shortest route online through unknown areas to create representative maps and acquire the necessary data, requiring various sensor sets.

In order to summarize, recent AUVs advancements have focused on three aspects, among others, that arose as of utmost importance for industries and the scientific community:

- perceiving and modeling the environment;
- autonomously planning the best path for carrying out assigned tasks;
- making autonomous decisions and deciding the best action to fulfill specific tasks.

In this context, this research work has focused on the first two intrinsically connected points. In particular, the research activity conducted during the Ph.D. period, at the Department of Industrial Engineering (DIEF) of the University of Florence (UNIFI), aimed at developing a framework that can lead an AUV toward fully autonomous surveys. To this end, a vehicle shall autonomously compute the mission path to inspect an area of interest, guaranteeing the acquisition of representative data by considering the sensor feedbacks and environmental characteristics.

Firstly, since optical cameras' exploitation in the underwater domain presents non-negligible drawbacks, strategies for modeling and meaningfully perceiving the surroundings using acoustic exteroceptive sensors were investigated. In detail, the use of state-of-the-art DL object detection methodologies for developing an onboard acoustic-based ATR solution for detecting and localizing OPIs in FLS images was deepen. Indeed, recognizing OPIs represents a preliminary and fundamental hierarchical stage for effectively understanding and gathering knowledge of the environment and thus, accomplishing interactive tasks and creating fully autonomous robots that can help human operators in challenging assignments.

Then, a sensor-driven CPP framework was developed in light of developing intelligent vehicles able to autonomously carrying out inspection surveys that can guarantee to acquire well-distributed high-quality data to model and investigate an area of interest. The framework, composed of a planning and a mapping module, is based on receding-horizon view planning methodology and was applied to FLS-based seabed inspections. To correctly plan and re-plan the survey, the CPP strategy uses a 3D probabilistic occupancy grid mapping system based on FLS imagery. Consequently, the AUV actively considers the sensor feedbacks during the mission by planning the survey according to the created map. Thus, it can guarantee adequate coverage of the area of interest.

Finally, to make the AUV actively taking into account the presence of OPIs, an object ATR results were included in the inspection framework, realizing a target-aware inspection solution. The FLS 3D probabilistic occupancy mapping system has been extended to include ATR findings and create a semantic occupancy map of the environment. It is used by the planning module so that, during the mission, OPIs are correctly recognized and localized.

1.1 Overall framework

The research activity described in this work has been carried out at the Mechatronics and Dynamic Modeling Laboratory (MDM Lab) of the Department of Industrial Engineering of the University of Florence (UNIFI DIEF). UNIFI DIEF is operative in the field of underwater robotics since 2010 thanks to the participation in the Tuscany-funded project TecnicHe per l'Esplorazione Sottomarina Archeologica mediante l'Utilizzo di Robot aUtonomi in Sciami (THESAURUS). Since then, the MDM Lab research group has taken part in several cutting-edge national and international projects in the marine robotics field. In 2013, the MDM Lab joined as the coordinating partner of the European FP7 project ARcheological RObot systems for the World's Seas (ARROWS). Afterward, in 2016, the research group participated in the Bridging Robots for Underwater Communication Enrichment (BRUCE) project, a sub-project of the European FP7-funded Sensing, monitoring and actuating on the UNderwater world through a federated Research InfraStructure Extending the future Internet (SUNRISE). Together with the official spin-off company MDM Team S.r.l. of UNIFI DIEF, the MDM Lab took part in the Autonomous underwater Robotic and sensing systems for Cultural HEritage discovery COnservation and in sitU valorization (ARCHEOSUb) European project in 2018. The group joined the H2020 EUMarineRobots (EUMR) and Operational Platform managing a fleet of semi-autonomous drones exploiting GNSS high Accuracy and Authentication to improve Security Safety in port areas (PASSport) pojects in 2019 and 2020, respectively. While PASSport has the goal of improving security in port areas through the combined use of Unmanned Aerial Vehicles (UAVs) and AUVs, EUMR aimed to open up key national and regional marine robotics research infrastructures to all European researchers from both academia and industry. Regarding the EUMR project, the MDM Lab acted as a node of Interuniversity Center of Integrated Systems for the Marine Environment (ISME). In fact, since 2014, the University of Florence has joined the ISME consortium that gathers academic institutions from all over Italy and whose main goal is to act as a common platform for joint operations for what concerns marine robotics. The support of the Italian Navy through the Naval Support and Experimentation Center (Centro di Supporto e Sperimentazione Navale) (CSSN) has led to the institution of the SEALab joint laboratory between CSSN and ISME. In the context of the EUMR project, the SEALab allowed conducting several experimental campaigns in the Ligurian Sea that led to the results obtained during the Ph.D. activity.

During the last ten years, according to the goal of the aforementioned projects, several AUVs were developed by the UNIFI DIEF MDM Lab. Each AUV of the MDM Lab fleet has its own characteristics and was developed for a specific purpose. The Typhoon class AUVs [31] were created in the context of the THESAURUS project as a swarm of AUVs to perform cooperative autonomous surveys of areas of archaeological interest in the Tuscan Archipelago. MArine Robotic Tool for Archaeology (MARTA) AUV [32] is a torpedo-shaped modular vehicle built during the European project ARROWS and designed to reduce the cost of archaeological operations. On the other hand, Zeno Environment Nautical Operator (Zeno) was developed with a compact design and high portability, together with a rapid battery replacement for the ARCHEOSUb project targets: surveying, conservation, and protection of new and existing underwater Underwater Cultural Heritage (UCH) sites. These vehicles were developed before this Ph.D. period, and since they were not used to test the developed methodologies, their descriptions will be omitted.

FeelHippo AUV completes the MDM Lab AUV fleet. It was initially built for participating in the Student Autonomous Underwater Vehicles Challenge-Europe (SAUC-E) competition in 2013. Since then, it was used in many robotics competitions during the last years. Among the milestones achieved by the team in these competitions, recently, the award of "Best Marine Team" was won during the European Robotics League - Emergency Robots (ERL) 2018 and ERL 2019 (see [33] and [34] for further details). Continuous overhauls and upgrades have made FeelHippo AUV a versatile platform for undertaking research topics. For instance, FeelHippo AUV was used for conducting the experiments of EUMR project. Besides, it was used as a testing platform for the methodologies proposed in this thesis. Thus, a detailed description of its main characteristics and the list of the equipped sensors is provided in the next section.

1.2 FeelHippo AUV

FeelHippo AUV, whose last version is depicted in Fig.1.1, is a compact and lightweight AUV developed by the MDM Lab in 2013 for joining student and non-student robotics competitions. In the last years, it has faced several hardware and software upgrades that have made FeelHippo AUV a reliable and versatile platform for conducting research activities and project-related tasks.

FeelHippo AUV has a central body made of a Plexiglass® hull, where the hardware and electronics are housed two aluminum pipes attached under the main body, which contain the batteries. At the time of writing, FeelHippo AUV propulsion is realized through six thrusters (two at the stern, two at the bow, and one each on both sides inclined at 45°) arranged in a vectored configuration that allows for the control of all the Degrees Of Freedom (DOFs) of the vehicle, except for the pitch motion that depends entirely on the inner mass distribution and the vehicle configuration.

FeelHippo AUV main characteristics are summarized in Table 1.1.

Table 1.1: FeelHippo AUV Main Features

Weight [kg]	35
Dimensions [mm]	600×640×500
Controlled DOFs	5
Thrusters	6
Maximum Depth [m]	35
Maximum Longitudinal Speed [m/s]	1
Battery Life [h]	4

The list of the primary electronic components and the sensor sets with which FeelHippo AUV is equipped are reported below:

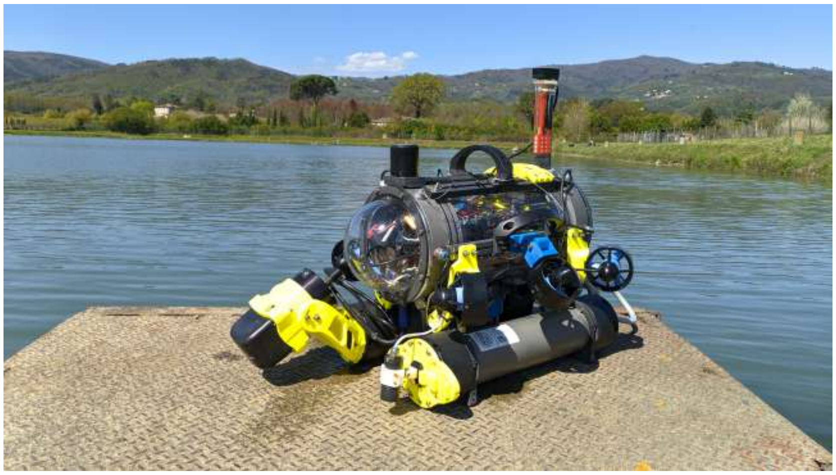


Figure 1.1: FeelHippo AUV before an on-field underwater mission.

- Intel i7-based LP-175-Commel motherboard (main computer);
- NVIDIA Jetson Nano (payload computer);
- U-blox 7P precision Global Positioning System (GPS);
- Orientus Advanced Navigation Attitude and Heading Reference System (AHRS);
- KVH DSP 1760 single-axis high precision Fiber Optic Gyrosope (FOG);
- Nortek DVL1000 Doppler Velocity Log (DVL), measuring linear velocity and acting as Depth Sensor (DS);
- EvoLogics S2CR 18/34 acoustic modem;
- Teledyne BlueView M900 2D FLS.
- one Microsoft Lifecam Cinema forward-looking camera;
- one Microsoft Lifecam Cinema bottom-looking cameras;

The main computer (Intel i7-based LP-175-Commel motherboard) runs the software architecture based on the Robot Operating System (ROS) framework [35], and it is used for onboard processing of guidance, navigation, and control algorithms, for supervising the state of the vehicle, and for managing the communication channels. The three payload computers (one NVIDIA Jetson Nano and two Intel Neural Compute Stick 2), connected to the main computer, are exploited for running onboard machine learning algorithms and the ATR solutions developed in the Ph.D. activity. An overview of FeelHippo AUV hardware is reported in Fig.1.2.

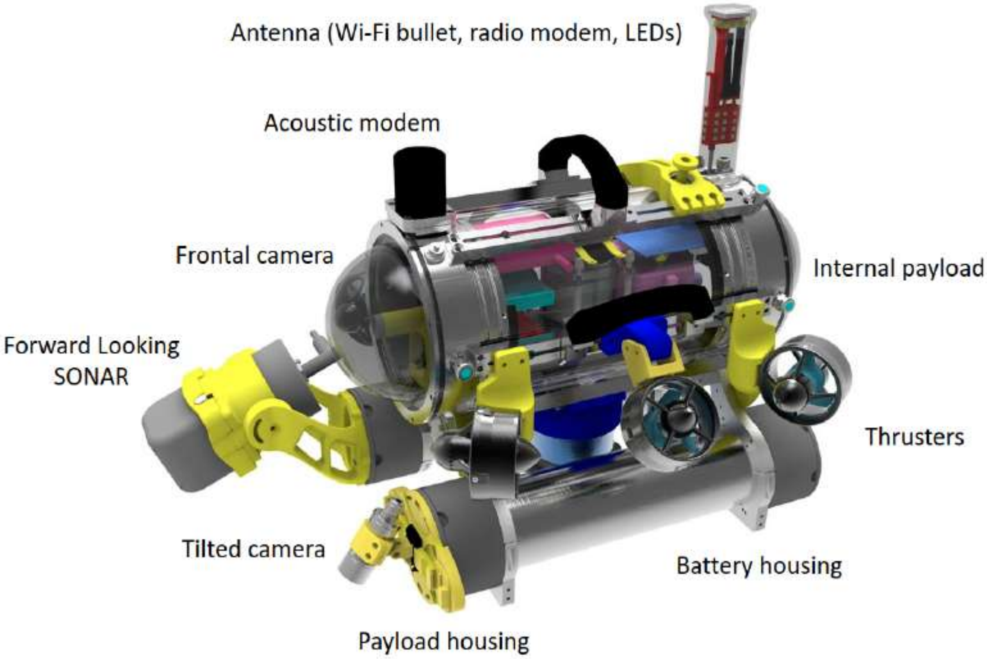


Figure 1.2: FeelHippo AUV hardware overview.

1.3 Thesis structure

This thesis is organized as follows.

Chapter 2 reviews state-of-the-art FLS-based ATR solutions and related works about coverage and exploration strategies. Chapter 2 also highlights the main contributions of this thesis.

Chapter 3 is dedicated to the notation used in this work, and introducing the ATR preliminaries, and planning and mapping concepts.

Then, Chapter 4 firstly describes the developed ATR framework and details the criteria used for selecting the CNNs models. The Chapter presents the validation of the ATR solution and the performance assessment. Finally, it reports the results of the conducted experimental campaign.

Chapter 5 introduces and discusses the designed framework for autonomous inspections. The structure of the framework is detailed, and the developed algorithm is presented. Besides, this Chapter reports a mathematical formulation of the developed solution and a novel theoretical analysis of the exploited volumetric information formulation's effects. The results of the realistic simulations and the dedicated experimental campaign at sea that aimed at validating the developed framework and the proposed analysis are reported in this Chapter.

In Chapter 6 advances for enhancing the performance of the developed coverage algorithm are proposed. In particular, Chapter 6 describes a parallel implementation of the algorithm and presents a novel sampling strategy for guiding the expansion of the tree towards unexplored areas. This novel sampling strategy, is based on estimating the density probability function of the discovered map that is used for sampling new points. Simulation and experimental results demonstrate the effectiveness of the proposed enhancements.

Chapter 7 describes the development of a Randomized Model Predictive Control (RMPC) algorithm for planning coverage surveys. The aforementioned solution for estimating the density probability function of the discovered map, detailed in Chapter 6 is utilized for developing a fast methodology for evaluating if a path is leading the vehicle toward non-covered regions. Realistic simulations demonstrated the effectiveness of the proposed solution.

Chapter 8 is dedicated to fusing the coverage and ATR methodologies for creating a target-aware planning solution for enabling an AUV to autonomously inspect an area of interest while actively considering the presence of OPIs by creating a semantic map. The solution was validated with realistic simulations.

Finally, Chapter 9 concludes the thesis by providing an analysis of the main achieved results and discussing possible future developments.

Chapter 2

Related works and thesis contribution

This section reviews the related works that inspired this Ph.D. research work. Firstly, state-of-the-art and innovative ATR approaches for the underwater domain are presented. Then, coverage and exploration strategies that influenced the solution developed during the Ph.D. period are reviewed. Lastly, this section highlights the main contribution of this thesis.

2.1 State-of-the-art ATR solutions

With the growing demand for intelligent systems capable of performing complex interactive tasks, reacting to the environment while inspecting areas, and cooperating meaningfully with human operators, object detection has become a fundamental feature of modern robots. UGVs and UAVs can rely on a large variety of sensors, ranging from optical cameras to Light Detection and Ranging (LiDAR) devices, to detect objects. Due to the wide use of modern cameras, several image-based target identification solutions have been developed. In particular, CNN-based approaches have shown outstanding results, becoming the golden standard in the image classification and target recognition tasks [29].

On the contrary, marine robots have limited recognition capabilities due to the underwater domain. Water turbidity, low-light conditions, and poor visibility degrade the quality of the optical images (Fig.2.1), making the subsea object detection hardly achievable in many cases. Acoustic sensors, such

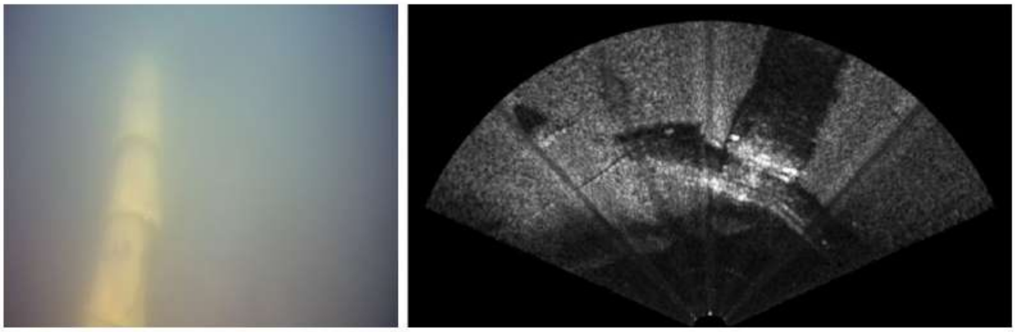


Figure 2.1: On the left, an underwater pipeline structure in the corresponding optical image (native resolution of 704 x 576 pixels); on the right, the 2D FLS acoustic frame (native resolution of 894 x 477 pixels). These images were acquired by FeelHippo AUV during the ERL competition in 2018. It is worth highlighting that the optical image was acquired by a distance of about 1.5 m, while the range of FLS used to gather the optical frame was set to 10 m.

as FLS or SSS, represent a valid alternative. Indeed, these sensors provide high-range data that are not as affected by water conditions. Besides, even though recognizing object patterns in the high-noise acoustic sonar images can be challenging, FLS has the potential to be a functional device in underwater ATR tasks by providing decent resolution images (an example is provided in Fig. 2.1), at high frame rates, and not requiring the vehicle to move.

Different Template-Matching-based object recognition approaches for FLS imagery have been developed and tested with different similarity measures and feature-trained classifiers [36] [37]. Nonetheless, these techniques cannot generalize the template patterns; additionally, their performance degrades in the handling of multi-scale objects. Therefore, these limitations led many marine researchers to investigate the use of CNN-based solutions also in acoustic imagery. In [30], custom CNN architectures to classify FLS images have been evaluated. The reported performance comparison with classical template matching solutions shows that CNNs could provide better performance while keeping a low number of parameters [38]. Nevertheless, developing a custom CNN architecture is time expensive and requires plenty of images to train the network. Besides, AUVs cannot rely on bulky hardware and usually have limited onboard computational power. In addition, ATR should be performed in real-time to be effective during underwater missions. Thus, developing a CNN for onboard applications does emerge as a real challenge.

Turning to a more detailed overview, these solutions follow a common approach. The first network layers, called the backbone (or feature extractor) of the network, are in charge of extracting the dominant features, while the last layers classify those features and localize objects in the image. Generally, the backbone is tricky to train and requires a large dataset. Conversely, by using transfer learning, the last layers can easily be trained on a custom dataset by fine-tuning higher-order feature representations, speeding up the training phase. In fact, the idea of transfer learning is simple: CNNs learn generic feature vectors, and thus feature learned from a dataset can be used to solve a different problem [39]. Consequently, for what concerns the object recognition task, pre-trained CNNs solutions can be used. The backbone can be frozen, and only the last layers are trained on a new dataset.

As regards the subsea environment, since gathering a large dataset in an underwater scenario is by no means straightforward, such Deep Neural Network (DNN) approaches, relying on transfer learning, could be used to tackle object detection in FLS images.

Several remarkable DNN architectures have been proposed in the last few years for several and disparate fields of application; considering that for FLS the scientific literature highlights sparse applications, the most performing and promising DNN architectures have been taken into account in order to find which ones best fit our system. The You Only Look Once (YOLO) network [40] was developed as an optimized end-to-end structure composed of 24 convolutional layers and 2 fully connected layers. This simple structure allows predicting bounding boxes and class probabilities from full images in one evaluation. Thus, the network achieves real-time image processing with an extremely high frame per second (fps). In [41] and [42], improvements to make YOLO faster and more accurate were introduced. In addition to these two upgraded networks, namely YOLOv2 and YOLOv3, several versions have also been developed that benefit from various speed/accuracy tradeoffs. The YOLO architecture was tested on FLS images ATR in [43], where the authors developed a system to detect divers.

A different approach was used to design the Single Shot Multibox Detector (SSD) [44], a convolutional network able to detect and classify objects at different scales at a high frame per second (fps). Its native version used the Visual Geometry Group (VGG) network [45] as a backbone to extract the image features. Also, different feature extraction networks, such as Inception [46] and Mobilenets [47], were tested. As will be shown in Chapter 4, the choice of the feature extractor plays a fundamental role since it affects the speed/accuracy tradeoff, which is of utmost importance when developing an ATR strategy for onboard real-time applications [48]. Small convolutional filters are then applied to different scale feature maps in the final layers to detect and classify objects. The network training aimed to optimize a multi-task loss that took into account both the classification error and the bounding box coordinate error. This simple structure lets the SSD reach high-accuracy detections at high fps (up to 45). As the underwater domain is concerned, SSD was used to recognize objects in optical images [25], but as far as the author known, it has not been tested on FLS imagery yet.

When the detection accuracy shall be favored over the inference speed, Region-based architectures, such as the Faster Region-based Convolutional Neural Network (Faster R-CNN) [49], are the recommended choice. The Faster R-CNN's backbone is composed of a feature extractor network and a Region Proposal Network (RPN) to produce the Regions Of Interests (ROIs) in the feature maps and predict the bounding boxes. Two fully-connected sibling layers take each ROI as input, and classify possible objects and refine the bounding boxes. The loss function used to train the network was a trade-off between the classification and the localization tasks. Generally speaking, compared with the YOLO and SSD, the Faster R-CNN is more accurate but cannot reach the exceptionally high inference speed.

Finally, the Mask Region-based Convolutional Neural Network (Mask R-CNN) [50] extended the Faster R-CNN. Firstly, the backbone was improved through the Feature Pyramid Network (FPN) that can better represent objects at multiple scales. Besides, the authors added in the final layers a convolutional branch to generate a segmentation mask for the selected ROIs. The training loss also considered the segmentation tasks, improving the network performance. In fact, instance segmentation enables identifying object outlines at the pixel level, enhancing the localization precision. R-CNN architectures were tested on optical underwater images [51] and on FLS imagery [25]. However, in [25] an analysis of their performance on FLS images was not reported.

2.2 Related works on Autonomous Inspections and Explorations

The exploration and CPP task is integral to several robotic applications and domains. It can be defined as the problem of autonomously inspecting an area of interest, i.e., moving the robot so that it can cover with a sensor all the points in a target area. Exploration and CPP algorithms have to compute a feasible path for the robot using simple motion trajectories [52] that is optimal, according to specific criteria, such as the path length or the total time to complete the mission. According to [53] and [54], such algorithms are classified as offline, which requires full prior knowledge of the environment and relies on on stationary information, and online, as they consider the sensor feedback. Offline algorithms are built on assumptions that might be unrealistic in many scenarios, as the underwater domain, where the conditions change continuously and rapidly, affecting both optical and acoustic payload sensors' performance. On the other hand, online strategies, also called sensor-based or sensor-driven, can deal with changing or unknown environments by considering as feedbacks the sensor measurements and reacting to the perceived environment. In this research, the problem of collecting FLS data over an area of interest, i.e., FLS-based inspection, is considered, and since AUVs are commonly used in unknown areas where a prior map is not available, the focus has shifted to online algorithms.

In the context of underwater robotics, the CPP problem is a typical dual-use topic and was considered, for instance, for Mine CounterMeasure (MCM) applications and also extended to exploration tasks.

An online approach for SSS seabed coverage for MCM was proposed in [6]. The AUV employed a multi-objective optimization that combines information theory with the concept of branch entropy to compute the heading reference. The branch entropy was proposed to overcome the limitations of the information gain, which is a useful tool for calculating the potential next moves' expected benefits. Still, it is not sufficient to achieve global goals when there is incomplete prior knowledge about the environment. In fact, it becomes a greedy-first search when applied to coverage tasks that could limit sensor-driven methods performance. The AUV performed constant altitude surveys in an equal-sized hexagon cell decomposed workspace. The MCM problem was also tackled in [55]. The authors presented a two-phase solution that considers Acoustic Doppler Current Profiler (ADCP) measurements to compensate for currents and sonar data processed with an object detection algorithm to adapt the survey to reinspect potential targets immediately. These methods, however, are limited to MCM or object hunting applications since they consider as a measurement the belief about the presence or absence of an object at a location.

Inspections of underwater structures have carried the development of the research described in [56]. The authors proposed a planning algorithm to compute the inspection path and a guidance strategy that uses the data perceived during the mission to adapt the vehicle trajectory to the terrain. However, to calculate the initial inspection mission, the algorithm required a prior bathymetric map of the area. For a similar application, in [57], an algorithm that uses a prior map to calculate the minimum number of viewpoints that provide the maximum amount of information was presented. Then, a trajectory that guides the AUV through the computed viewpoints is computed by solving a Traveling Salesman Problem (TSP). Again, this strategy needs a prior map, meaning that either it was available or the AUV had to perform a pre-survey to gather a preliminary bathymetric map of the area.

To carry out underwater exploration in 3D unknown environments, Vidal and colleagues developed a two-layer planning system that considers optical and acoustic data to select the viewpoints [5]. The environment was represented with a cubic cell-based map, where each cell was labeled according to multisensor measurements. Acoustic data, provided by a multibeam sonar, were used to determine whether a cell was occupied or not, while an estimation of the camera Field of View (FoV) marked the cells as viewed. The view planner utilized the map to compute range and camera candidates deterministically. Then the viewpoints were generated along the surface normal at a distance determined by the sensor parameters (range and FoV). To select the best viewpoint, candidates were evaluated according to a metric function that used the distance between the robot and the viewpoint and the orientation difference. The selected Next-Best Viewpoint (NBV) was sent to the motion planner layer, responsible for computing a safe and feasible path to lead the AUV to the viewpoint. To this end, the asymptotic optimal Rapidly-exploring Random Tree (RRT) (RRT*) algorithm [58] was employed. Outstanding results were reached both in simulations and in real sea trials; the Girona 500 AUV, used as a testing platform, managed to explore challenging unknown scenarios. Nevertheless, the view planner computed the NBV among the Frontier Points (FP), i.e., points in the regions between known and unknown spaces. This strategy is a proper approach for exploration tasks, but it limits the performance of inspection algorithms. De facto, it ends up being a greedy search strategy and leads to longer paths for adequate coverage levels in CPP applications.

In the context of 3D object reconstructions, it is possible to find several similarities with the CPP problem considered in this research work. In this context, researchers considered the problem of selecting the NBV to perform an active volumetric 3D reconstruction of an object with a camera attached to a mobile robot. Deterministic [59] [60] and stochastic [61] candidates selection processes have been proposed in the last years. Then, in [62], which extends the work of [63], an analysis of volumetric information gain metrics is proposed. The voxels' entropy was used to consider the volumetric map uncertainty in the NBV selection criteria, and the authors proposed different ways to quantify the information contained in the voxels. In this paper, the voxels' entropy idea and the information metrics, shown in the works mentioned above, were considered. However, as for FP methods, considering only the next best action makes those approaches greedy.

Finally, the coverage methodology proposed in this research was inspired by the strategies presented in [64] and [65], where MAVs used novel path planning methods to explore unknown 3D spaces with a camera. Random trees were expanded in the working space. Each node of the tree represented a viewpoint that was evaluated according to the amount of unmapped space visible from the viewpoint. The best branch in the constructed tree was selected. Then, only the first viewpoint was executed (becoming the NBV), and the process was repeated in a receding-horizon manner. For the sake of completeness, in [64], a second planning layer (based on random trees) was employed to find a path that minimizes the expected localization and mapping uncertainty. By considering the branch information gain, these methods predicted the effects of more actions than just one and overcame the limitations of the information gain method in unknown environments. Besides, random trees have the non-negligible advantage of being able to implement nonholonomic constraints conveniently and can run onboard in real-time on small robots with limited computational capability. These methodologies were developed for the aerial domain for carrying out autonomous explorations with MAVs. To the best of the authors' knowledge, such random tree-based strategies have not been extended to the underwater domain yet. This research work investigates the use of receding-horizon sensor-driven random tree-based approaches for AUV coverage tasks. The here presented methodology enables an AUV to autonomously inspect an unknown area and, by actively considering the sensor feedbacks, guarantees satisfying coverage levels.

The previously discussed solutions rely on a mapping strategy. In detail, the robot must create a map that reflects the structure of the environment and other meaningful information for the tackled task. For instance, the map can be used to distinguish between known/unknown and free/occupied regions so that the robot can autonomously conduct a safe exploration or CPP survey. To this end, the well-known theory of occupancy grid mapping has shown outstanding results [66]. Besides, to increase environmental awareness, maps can include high-level data. In [67], the authors proposed a semantic mapping strategy that added to the geometrical data, the results of a CNN-based object detection solution. An analogous mapping strategy was used in [68] to realize a semantically–enhanced path planning strategy for UAVs for exploring unknown environments and simultaneous searching for objects of interest. Similarly, in [69] an object-aware map was used for exploration tasks with UAVs, and a benchmark of different strategies was reported. For the sake of completeness, in the field of aerial vehicles, maps are also used for active perception, i.e., path planning that incorporates the expected outcomes of perception, where they can include the amount of information of an area [70], [71].

While UGVs and UAVs can exploit a large variety of sensors, marine robots usually have limited perception capabilities. Indeed, optical devices (e.g., optical cameras) [72] and [73] or LASER-based solution [74], even if unquestionably worth, lay themselves open to water conditions. Thus, due to the favorable property of acoustic propagation in the underwater environment, acoustic devices have emerged as a strong alternative in underwater sensing.

Briefly, for what concerns imaging-sonars, such as the FLS device used in the methodologies developed in this work, to the authors' best knowledge, the first mapping solution dates back to [75]. Here, the method is applied to MSIS data, where 2D reconstruction only is involved. More recently, in [76] a probabilistic 3D occupancy mapping framework is presented, where the OctoMap library [66] is employed. The group of the University of Girona, Spain, presented remarkable solutions for acoustic [77] and multisensor (opto-acoustic) mapping [5]. In [78], a volumetric map representation that supports FLS devices is developed; however, a concentrator lens is employed to reduce the aperture of imaging devices. In [9], the authors present an FLS-based probabilistic 3D occupancy mapping framework tailored explicitly to AUVs, where the mapping layer takes advantage of the OctoMap library [66]. A novel line of work [79], [80], and [8] based on the OctoMap framework, tries to provide a unique map representation useful both localization, under the Simultaneous Localization And Mapping (SLAM) paradigm, and planning. Nevertheless, this is not the core of this work.

2.3 Thesis contribution

This research work investigates the development of a methodology for making an AUV able to autonomously conducting an inspection survey, ensuring adequate coverage of the target area while looking for OPIs. In the underwater domain, state-of-the-art inspection missions are usually represented by pre-planned surveys, such as lawnmower and zig-zag paths, where the AUVs passively store the collected data. Such paths can be optimized by considering the utilized sensor characteristics, but environmental conditions cannot be forecasted, and thus, they do not guarantee the acquisition of data over the entire area of interest and could lead to repeated attempts. In the last years, some solutions to overcome these limitations have been proposed. However, as reported in Section 2.2, such CPP solutions were developed for MCM applications and only considered as feedback the presence or absence of an object. Solutions for inspections of underwater structures or exploration of unknown areas could seem reasonable. Still, they either rely on a prior map or end up being a greedy strategy when it comes to CPP applications. As the aerial domain is concerned, some remarkable solutions, which could inspire an AUV tailored CPP solution, have recently been presented, but to the best of authors' knowledge, their exploitation has not been investigated yet by the marine community.

Therefore, this research work aims at investigating the use of receding-horizon sensor-driven random tree-based approaches solutions for conducting FLS seabed inspections, but suitable for any acoustic or optical sensor. In particular, the authors have sought a solution that overcomes the drawbacks of the previously mentioned approaches; that is, it does not require a prior map; it uses the expected benefit of more actions than just one, i.e., does not turn into a greedy approach that leads to longer paths, and can re-plan online the mission by considering as feedbacks the covered area and not only the presence or absence of objects.

In detail, the 3D probabilistic occupancy mapping system for FLS reconstructions presented in [9] (UNIFI DIEF's work) was used, and a receding-horizon sensor-driven view planning methodology was developed. Based on random trees, the here proposed solution considers the vehicle kinematics using the Dubins constraints and evaluates the tree nodes according to the updated occupancy map and information metrics. At each iteration, the best branch is selected, and only the first node becomes the NBV that is then reached by the AUV. The process is repeated in a receding-horizon manner: when the vehicle arrives at the waypoint, the process is repeated by maintaining the best previously computed solution in order to find the new NBV. A comparison of volumetric information gain metrics, along with a theoretical analysis of the developed Receding Horizon Coverage Approach (RHCA) is reported in Section 5.3 and Section 5.4. In addition, a novel sampling methodology for guiding the expansion of the tree towards unexplored areas was developed and tested. This innovative sampling solution is based on estimating the probability density function of the discovered map by using a Kernel Density Estimation (KDE) approach [81] [82], and improves the performance of the RHCA.

Moreover, to enable the AUV to autonomously detecting and localizing OPIs during the inspection survey, the use of state-of-the-art CNNs for an ATR strategy for real-time onboard applications was investigated. In particular, the research activity has focused on developing and evaluating a CNN-based ATR solution on FLS imagery. Since the ATR solution allows to understand and gather knowledge of the environment effectively, the exploited FLS mapping system was extended to include ATR findings and create a semantic occupancy map of the environment. Thus, an object-aware planning system that leads the AUV to correctly localize the OPIs, during the survey, was realized.

Therefore, FeelHippo AUV, selected as the testing platform, by using the here proposed ATR and planning strategies, can autonomously conduct an inspection survey, ensuring to acquire well-distributed high-quality data while actively considering the presence of OPIs.

In conclusion, the main contributions of this work are:

- The development and testing of an ATR methodology to identify and localize real-time onboard OPIs in FLS imagery by using state-of-the-art CNN architectures. The conducted experimental campaign showed that the ATR strategy can endow an AUV with the ability of autonomously recognizing OPIs;

- A sensor-driven RHCA that uses a random trees-inspired algorithm. The view planning algorithm uses the map created through an FLS-based reconstruction system to evaluate the randomly generated tree node and selects the NBV. The proposed view planning strategy was firstly validated with realistic simulations performed by means of the Unmanned Underwater Vehicle Simulator (UUV Simulator). Then, it was tested in real experiments conducted at sea to assess the performance of the proposed RHCA;

- A mathematical formulation of the developed algorithm and a theoretical analysis of the effects of the exploited volumetric information formulation. The results of the simulations and the real trails performed at sea constitute a benchmark of information gain metrics for AUVs inspection tasks. The RHCA algorithm was tested with a volume-based and an entropy-based gain metric. The comparison results are reported and discussed. As far as the author knows, this is the first study of information gain metrics for coverage tasks in the underwater domain;

- A novel sampling strategy based on estimating the probability density function of the discovered map by using a KDE approach for guiding the expansion of the tree towards unexplored areas was developed and tested. After the validation of the solution through realistic simulations, an experimental campaign at sea was conducted. The results show that this informed tree expansion methodology improves the performance of the RHCA;

- A preliminary investigation of the use of an approach for view planning. In particular, a RMPC strategy is utilized for generating possible inspection paths that steer the AUV during the inspection mission. Here, the KDE-based discovered map density function is employed for rapidly evaluating the viewpoint candidates. The effectiveness of the proposed strategy was demonstrated with simulations;

- To make the AUV able to inspect the area of interest, while simultaneously searching and mapping for OPIs in the environment, the mapping system has been extended to include the ATR findings and thus creates a semantic map. Then, an object-aware planning strategy was realized. Realistic simulations made using the UUV Simulator validated the proposed methodology;

Chapter 3

Preliminaries and notation

This Chapter describes the notation and reference systems employed in the rest of the work and provides a complete review of the fundamental theoretical and mathematical concepts used throughout this thesis.

3.1 Notation, reference systems, and FLS concepts

Firstly, the mathematical notation used in this thesis is introduced. Given a generic reference system $\{O^i x^i y^i z^i\}$, a vector $\boldsymbol{p} \in \mathbb{R}^3$ expressed in this frame will be denoted as ${}^i\boldsymbol{p}$. A rotation matrix $R \in \mathrm{SO}(3)$, for which it holds that $R \in \mathbb{R}^{3\times 3}$, $RR^\top = I_3$, where I_3 is the 3×3 identity matrix, and $det(R) = 1$, is referred as ${}^kR_i^j$; it rotates unit vectors of the frame $\{O^i x^i y^i z^i\}$ in unit vectors of the frame $\{O^j x^j y^j z^j\}$, both expressed in the frame $\left\{O^k x^k y^k z^k\right\}$. If $k = j$, the three-indexes notation could be simplified in the following form: ${}^jR_i^j = R_i^j$. Introducing the transformation matrix $\boldsymbol{T}$ of the special Euclidean group in $\mathbb{R}^3$:

$$\mathrm{SE}(3) := \left\{ \boldsymbol{T} = \begin{bmatrix} R & \boldsymbol{p} \\ \boldsymbol{0}^T & 1 \end{bmatrix} \mid R \in \mathrm{SO}(3), \boldsymbol{p} \in \mathbb{R}^3 \right\}, \tag{3.1}$$

the relation between two reference frames can be described in a compact notation using homogeneous transformations and the four-dimensional representation vector $\widetilde{\boldsymbol{p}}$. In particular, it holds that:

$$ {}^{j}\widetilde{\boldsymbol{p}} = \begin{bmatrix} {}^{j}p \\ 1 \end{bmatrix} = \begin{bmatrix} R_i^j & \boldsymbol{t}_i^j \\ \boldsymbol{0}^T & 1 \end{bmatrix} \begin{bmatrix} {}^{i}p \\ 1 \end{bmatrix} = \boldsymbol{T_i^j} {}^{i}\widetilde{\boldsymbol{p}}, \tag{3.2} $$

where $\boldsymbol{t}_i^j$ is the translation vector between the center of the frames $< i >$ and $< j >$.

Then, the reference systems can be defined and the Society of Naval Architects and Marine Engineers (SNAME) notation is used throughout this thesis [83]. The North-East-Down frame, denoted as $< N >$, $\left\{O^N x^N y^N z^N\right\}$, is commonly used in marine robotics as the inertial reference system. It is a local Earth-fixed frame whose axes point, North, East, and Down (NED) respectively, and its center is placed on the Earth's surface at a specific latitude and longitude pair, depending on the specific application [84], [85]. Attached to the vehicle, a reference system called body frame $< b >$, $\left\{O^b x^b y^b z^b\right\}$, is defined assuming the x-axis along the longitudinal axis of the vehicle, the z-axis pointing downwards, and the y-axis completes a right-handed system (see [83]).

The vehicle considered as a testing platform for the methodologies proposed in this research work is FeelHippo AUV, described in Section 1.2, which estimates its pose with sufficient accuracy for conducting monitoring, mapping, and inspection surveys, by using high-grade sensors and navigation strategies developed by the UNIFI DIEF. Further information about the exploited navigation solutions can be found in [86], [87], and [88]. Thus, as far as this thesis does not deal with the development of underwater navigation strategies, the relation between the $< N >$ frame and the $< b >$ frame, $\boldsymbol{T_b^N}$, is considered completely known [83].

Generally speaking, for each sensor mounted on an AUV, a reference frame $< s >$, $\{O^s x^s y^s z^s\}$, can be defined. Thus, a generic sensor acquires measurements in the $< s >$ frame, and by knowing the sensor mounting pose with respect to the AUV, i.e, $\boldsymbol{T_s^b}$, the measurements can be projected in to the $< b >$ frame. This holds for both interoceptive (e.g., Inertial Measurement Unit (IMU), FOG, DVL, and so on) and exteroceptive sensors (e.g., optical cameras, FLS, just to mention a few). Regarding the FLS, which is the main sensor considered in this thesis, it is rigidly attached in front of the AUV, and a right-handed reference system, denoted as $< F >$, $\left\{O^F x^F y^F z^F\right\}$, can be considered. The introduced $< F >$ frame center corresponds with the FLS center; its x-axis points forward, while the z-axis points downwards. Since the FLS mounting position and orientation with respect to the AUV are known, the homogeneous transformation $\boldsymbol{T}_{\boldsymbol{F}}^{\boldsymbol{b}}$ is determined. Fig.3.1 depicts the overall situation.

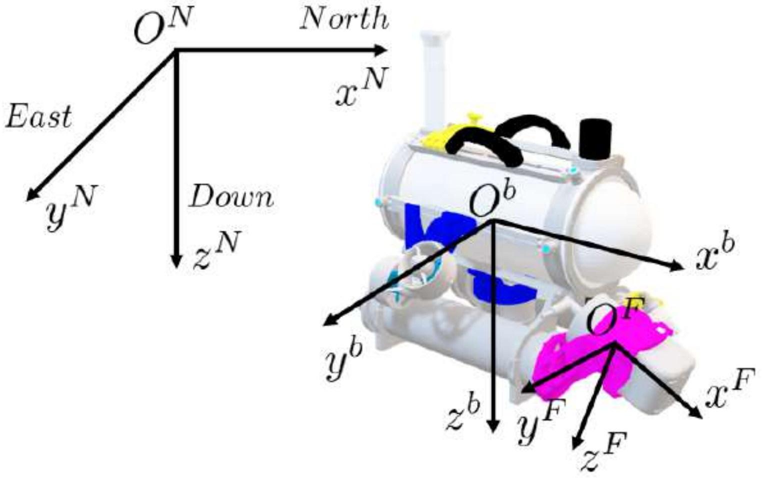


Figure 3.1: Representation of the NED frame $< N >$, the body frame $< b >$, and the FLS frame $< F >$.

According to [88], [89], and [90], in the $< F >$ frame, a point $\boldsymbol{P} \in \mathbb{R}^3$ represented in Cartesian coordinates ${}^F\boldsymbol{P} = [X, Y, Z]^\top$ can be expressed in spherical coordinates ${}^F\boldsymbol{P} = [\bar{R}, \alpha, \phi]^\top$, where $\bar{R}$ is the FLS delivering range, α is the azimuth angle, and ϕ the elevation angle. It holds that:

$$\begin{gathered}\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \bar{R} \begin{bmatrix} \cos\phi \cos\alpha \\ \cos\phi \sin\alpha \\ -\sin\phi \end{bmatrix} \\ \begin{bmatrix} \bar{R} \\ \alpha \\ \phi \end{bmatrix} = \begin{bmatrix} \sqrt{X^2 + Y^2 + Z^2} \\ \tan^{-1}(Y/X) \\ \tan^{-1}\left(-Z/\sqrt{X^2+Y^2}\right) \end{bmatrix}.\end{gathered} \tag{3.3}$$

FLS devices natively use the spherical system in the imaging process: for each beam that composes the FoV, at every range interval, the average power of the reflected waves is measured and used to produce the corresponding pixel intensity in the image. However, the 3D to 2D image formation process leads to a loss of the information about the elevation angle ϕ [90]. In fact, as depicted in Fig.3.2, the 3D point ${}^F\boldsymbol{P}(\bar{R}, \alpha, \phi) \in \mathbb{R}^3$ is projected on the FLS image plane, depicted in red in Fig. 3.2 and denoted in the following as $< F_I >$, in a point $\boldsymbol{p}$ along the arc defined by the elevation angle ϕ [91]. Hence, given an FLS image, only the azimuth angle α and the range $\bar{R}$ of point can be computed. FLSs typically have limited vertical beamwidth ϕ [92], [91]. Besides, vehicles such as FeelHippo AUV, considered in this work, have the roll and pitch dynamics hydrostatically stable, and seabed inspection surveys do not excite these Degrees Of Freedom (DOFs). Hence, the AUV navigates with roll and pitch angles almost zero with negligible variations. As a consequence of these considerations, a point ${}^F\boldsymbol{P}$ can be localized through its projection $\boldsymbol{p}$ in the FLS image plane. Thus, the approximation ${}^F\hat{\boldsymbol{P}}$ of ${}^F\boldsymbol{P}$ can be computed as:

$$
{}^F\hat{\boldsymbol{P}} = \begin{bmatrix} \hat{X} \\ \hat{Y} \\ \hat{Z} \end{bmatrix} = \bar{R} \begin{bmatrix} \cos\alpha \\ \sin\alpha \\ 0 \end{bmatrix}. \tag{3.4}
$$

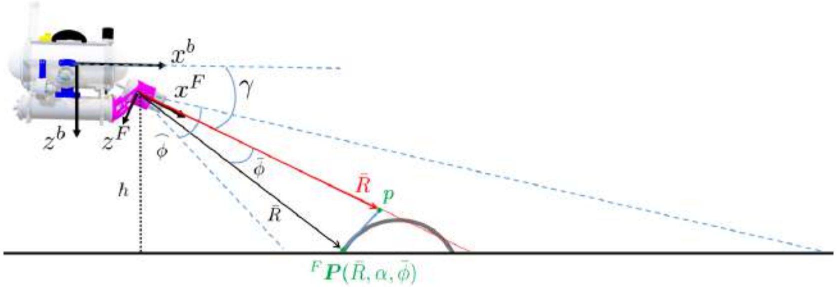


Figure 3.2: The FLS imaging process: given a range $\bar{R}$, points on the arc defined by the angle $\widehat{\phi}$ are projected in the FLS image plane $< F_I >$ (in red) in the point $\boldsymbol{p}$.

As described in Section 3.2, the proposed ATR solution uses image-based DNN architectures to identify OPIs in FLS images. Such models give as output the predicted classes, with the computed confidence, and the object bounding boxes. In particular, the bounding boxes are provided as the top-left and the bottom-right corners in the image reference frame $< I >$, which uses pixels as the measurement unit and whose center is in the image top-left corner and the x- and y-axis along the image width and height, respectively. To correctly localize the ATR findings, they shall be projected into an inertial reference frame $< N >$. To this end, the relation between the $< I >$ frame and the $< F_I >$ frame is known as depicted in Fig.3.3. Thus, for each OPI identified by the trained neural network, the bounding box can be projected from the $< I >$ frame to the $< F_I >$ frame; then the target can be localized: by using Eq.3.4 its position is estimated in the $< F >$ frame, and since the transformation $\boldsymbol{T_F^N} = \boldsymbol{T_b^N} \boldsymbol{T_F^b}$ is known, it is localized in the inertial reference frame $< N >$, and consequently, in the World Geodetic System (WGS84), which uses latitude, longitude and altitude as coordinates (see [84] for more details).

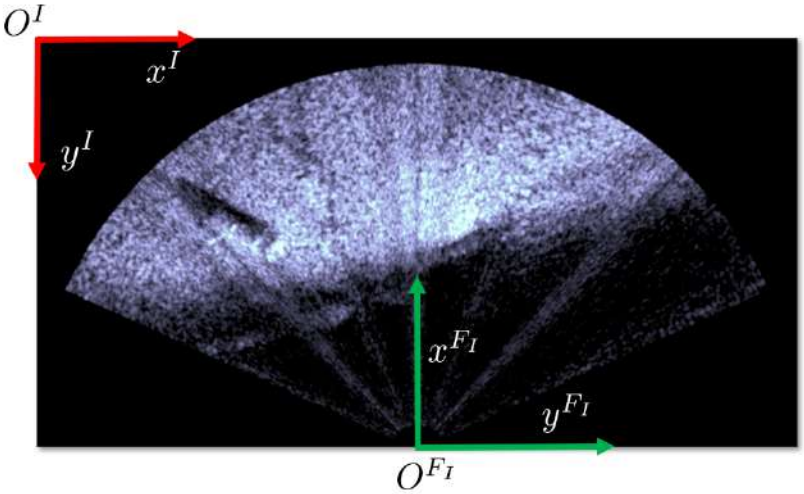


Figure 3.3: The 2D FLS frame with respect to the image reference system $< I >$.

For what concerns this research work, the considered OPIs to be identified and localized lie on the seabed, and by assuming that the sea bottom imaged within a frame is dominantly flat, which means that the DNN detected bounding boxes lie on the seafloor at an altitude h from the AUV, see Fig.3.2, a more accurate model than the one presented in Eq.3.4 for localizing the OPIs can be determined. That is, the elevation of the point ${}^F\boldsymbol{P}$ can be retrieved using altimeter data. In fact, according to the local flat seafloor hypothesis, for a point ${}^F\boldsymbol{P}$ on the sea bed, it holds:

$$\bar{R} \sin(\gamma + \bar{\phi}) = h, \tag{3.5}$$

where γ denotes the FLS tilt angle w.r.t. the horizontal plane. Eq.3.5 allows to calculate the elevation angle $\bar{\phi}$, and thus the point ${}^F\boldsymbol{P}$ can be localized accurately:

$$
{}^{F}\boldsymbol{P} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \bar{R} \begin{bmatrix} \cos\alpha\cos\bar{\phi} \\ \sin\alpha\cos\bar{\phi} \\ \sin\bar{\phi} \end{bmatrix}. \tag{3.6}
$$

In conclusion, Eq.3.5 and Eq.3.6 can be used to localize the detected bounding boxes with respect to the $< F >$ frame. Consequently, they can be localized with respect to to the $< N >$ frame by using the transformation $\boldsymbol{T_F^N} = \boldsymbol{T_b^N T_F^b}$.

3.2 Convolutional Neural Network background

This Section aims at providing the main concepts related to CNNs and their use for the object detection task (also called ATR in the marine community). It is intended to be an introduction to the use of such technologies for object recognition. In particular, the key concepts are described with the goal of providing an overview of the fundamental working aspects of CNN architectures. That is, this Section is not meant to give a theoretical dissertation about DL aspects, but it is willing to cover many practical issues beneficial for end-users when selecting and training a proper state-of-the-art CNN.

3.2.1 Machine Learning and Neural Networks

A Machine Learning (ML) algorithm is an algorithm that is able to learn from structured data. In more practical terms, for solving an image classification problem by using a ML solution, a feature extractor algorithm that detects the key points of the image is needed. Indeed, the ML model works on the outputs of the feature extractor [93]. On the other hand, DL solutions, which are a subset of ML, do not necessarily need structured data since they learn to extract the features that characterize the data by using more complex structures. Both ML and DL can be trained with two types of learning methodology: supervised and unsupervised. While the former relies on labeled data (i.e., data where the ground-truth is provided by an operator), in the latter, the training dataset is a collection of examples without a specific desired outcome [94].

A Neural Network (NN) is a DL structure based on a collection of connected nodes called artificial neurons; each node can transmit data to another one. These kinds of networks have been used on various tasks, like computer vision, speech recognition, medical diagnosis, and so on. Generally speaking, the idea of NN is to learn a model from a dataset and then use the model to evaluate samples that the model has not seen during the learning phase. In order to work properly, a NN must learn to generalize the dataset pattern; that is, it has to extract generic features that can be used to characterize new samples. To this end, a NN architecture shall be designed so that it has a sufficient learning capacity for the tackled task. However, the required learning capacity, which depends on the designed architecture and the number of parameters/weights, cannot usually be estimated a priori, and a trial-and-error approach should be followed. A NN learns during the *training* that consists of a gradient-based optimization procedure with respect to a *loss function* with the goal of finding the best set of parameter/weight values that model the training dataset features. The success of the *training* depends on several key parameters [95]:

- *Dataset*: the available set of data is of utmost importance since it determines the features that a model can learn. Generally speaking, in supervised learning, the *dataset* is composed of data and the corresponding ground truth, which is the correct value for the feature that the model shall learn. The *dataset* is usually split in three parts: the *training set*, the biggest set that is used to train the model, the *validation set*, on which the model is tested during the tuning of the *training* parameters, and the *test set* for evaluating the trained model.

- *Loss Function*: it defines the task of the ML model. It is the objective function of the optimization procedure, which is designed so that the model performance improves when it decreases. It measures the difference between the NN output and the ground truth value, and it determines the gradient used by the *optimizer* to update the value of the model weights.

- *Optimizer*: it is the optimization algorithm that minimizes the *loss function* L by updating the model weights Θ. To this end, the most common strategy is the Gradient Descent:

  $$\Theta_{n+1} = \Theta_n - \alpha \nabla L(\hat{y}, y), \tag{3.7}$$

  where y is ground truth value, $\hat{y}$ is the NN predicted output, and

$\alpha \in [0, 1]$ is the *learning rate*. Several gradient descent optimization algorithms, such as Stochastic Gradient Descent (SGD) with/without momentum [96], RMSProp [97], ADAM [98], just to mention a few, have been proposed in the last years. These algorithms aim at efficiently *training* NN and ML models by using learning rate schedules and more effective weight updating routines [96].

- *Learning Rate*: it controls the convergence rate of the *optimizer*. Large *learning rate* values could lead to an unstable optimization process since it can overshoot the minimum value of the *loss function*, which eventually makes the *training* phase fail. On the other hand, excessively small values make the *training* slow, increasing the time for developing a proper NN architecture. Typically, it is set to small values $[10^{-1}, 10^{-3}]$.

- *Number of Epochs*: it defines the length of the *training*. If a NN is trained for too long, the model could overfit the dataset. That is, it fails to generalize when tested on new data since it has learnt non-generic features or noise in the training dataset. Conversely, if the *training* is stopped too early, the model could not have converged. Predicting a proper *number of epochs* is by no means straightforward, and several attempts should be performed. A common strategy to overcome the issues related to the selection of the *number of epochs* is the early stopping criterion, which consists of monitoring the *loss function* of a validation dataset, composed of different samples than the training dataset. The *training* is stopped when the validation loss increases.

When the *training* is completed, the NN model shall be evaluated on the *test set*. To characterize the performance of a NN model, several aspects must be considered. Firstly, metrics that allow to correctly evaluate the precision of the NN predictions for the tackled task are selected. Then, if the model was designed for real-time applications, the time required for analyzing new samples (often called the *inference speed*) plays a fundamental role and must be carefully evaluated. This final step, where the network is evaluated, is crucial for comparing different NN models and select the best option for the considered application.

3.2.2 Convolutional Neural Networks and Object Detection

CNNs are NN exploiting a combination of convolutional and pooling layers. In a nutshell, convolution layers convolve the input image with a set of filters (the neurons of CNNs, also called kernels) and pass the results of this operation through a non-linear activation function. The pooling layers, instead, reduce the dimensions of data by combining the outputs of neuron clusters at one layer into a single neuron in the next layer. More details regarding CNNs can be found in [94].

CNNs have become the golden standard in many visual tasks, such as Image Classification and Object Detection [29]. Indeed, their structure is optimized for analyzing images and allows them to take advantage of some essential properties. Firstly, convolutional layers can extract relevant image features by learning the filter weights. By adding pooling layers, more complex features can be extracted. In fact, as shown in [99], the use of subsequent convolutional and pooling layers leads to a feature hierarchy, where simple features, such as edges and corners, are extracted in the first layers. Then more complex elements are identified in the last layers. Additionally, CNNs use filters that do not depend on the spatial position in the image and consider that the closer the pixels, the higher the correlation.

For what concerns the use of CNNs for Object Detection, as reported in Chapter 2, several breakthrough architectures (YOLO, SSD, and Faster R-CNN) have been proposed in the last few years. These networks use different approaches but share a common working principle: the first layers compose the backbone that extracts the relevant features (thus, the backbone is also called feature extractor), and then the final layers classify and localize possible objects of interest. Such image-based object detectors (in the following object detectors) take as input an image and give as output the predicted classes, with the computed confidence, and the object bounding boxes, provided as the top-left and the bottom-right corners in the image reference frame, whose center is in the image top-left corner and the x- and y-axis along the image width and height, respectively. For the sake of the reader, a general object detector workflow is depicted in Fig. 3.4.

To characterize the performance of an object detector and then compare different solutions, evaluation metrics must be defined. According to [48], the Microsoft Common Objects in Context (COCO) metrics [100] have been used for the investigations conducted in this work. Considered the output

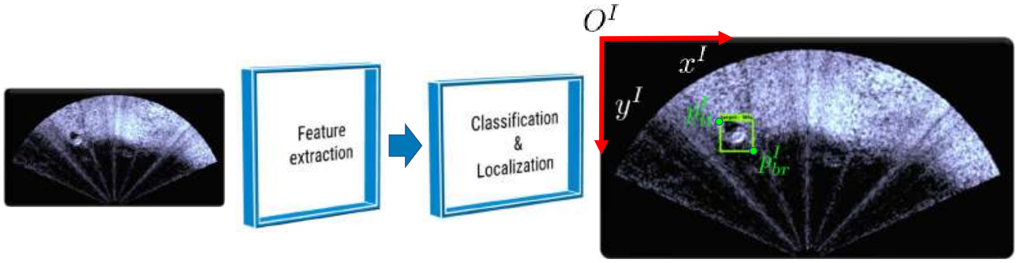


Figure 3.4: A general workflow of an object detector. Given an image as input, it outputs the predicted classes, with the computed confidence, and the object bounding boxes coordinates, expressed as the top-left (p_{tl}^I) and the bottom-right (p_{br}^I) corners in the image reference frame $\{O^I x^I y^I\}$.

of object detector algorithms, to decide whether a prediction is correct or not for an object, the Intersection over Union (IoU) is exploited. In detail, the IoU is computed as the intersection between the predicted bounding box and ground-truth bounding box divided by their union (see Fig.3.5), and a prediction is considered correct if the IoU is greater than a threshold (0.5 at least). Thus, the predictions of object detectors can be classified as follows:

- *True Positive (TP)*: A correct detection, i.e., a detection with IoU greater than a defined threshold
- *False Positive (FP)*: A wrong detection, i.e., a detection with IoU less than a defined threshold
- *False Negative (FN)*: A non detected ground-truth

To characterize the accuracy of ML and DL models, *Precision* and *Recall* are often used. The former is defined as the ratio between true positive over the sum of true positive and false positive:

$$Precision = \frac{TP}{TP + FP}, \tag{3.8}$$

while, *Recall* is the percentage of true positive detected among all relevant ground truths:

$$Recall = \frac{TP}{TP + FN}. \tag{3.9}$$

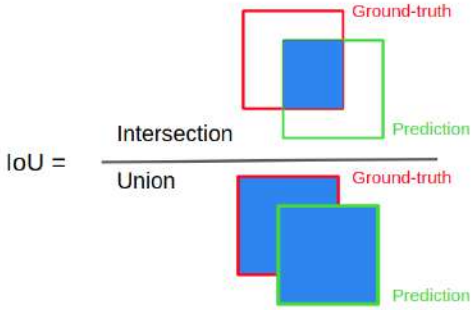


Figure 3.5: Rendering of the IoU metric.

As the selected IoU threshold increases, the detection task becomes more challenging. As a matter of fact, a good detector identifies only relevant objects (i.e., $FP = 0$) and all ground truths (i.e., $FN = 0$). The trade-off between precision and recall can be seen in the *Precision - Recall (PR)* graph (an example taken from [101] is reported in Fig. 3.6), which serves as an evaluation of the performance of an object detection model and is monotonically decreasing. An object detector is considered a good model if the precision stays high as the recall increases.

In COCO metrics, the mean Average Precision (mAP) is computed considering the *precision-recall* curve [100], and is a value that sums up the accuracy of a detector [48]. In detail, the mAP can be computed considering different IoU thresholds, typically 0.5 and 0.75 (denoted as mAP@0.5 and mAP@0.75, respectively). However, the official COCO metric uses the mAP averaged over IOU thresholds from 0.5 to 0.95 with a step size of 0.05, referred as mAP@[0.5 : 0.05 : 0.95] or mAP.

In conclusion, as this work is concerned, the precision of an object detector is evaluated by means of the mAP. However, as explained in the previous section, the accuracy of a detector is not the only parameter that shall be considered when developing an ATR solution for real-time onboard applications. In fact, the *inference speed* is of utmost importance since it determines whether an object detector can be deployed or not on the available hardware.

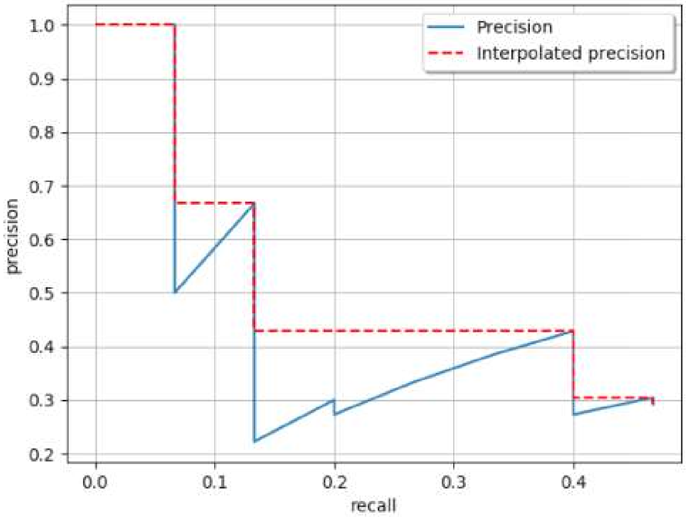


Figure 3.6: An example of the PR curve (in blue) taken from [101]. The graph reports the interpolated PR curve (in red) employed to reduce the impact of fluctuations. It is computed by considering the precision value for some recall samples, usually 11 and equally spaced.

Regarding image-based object detectors, the *inference speed* is evaluated as frames per second (fps).

3.3 Occupancy mapping fundamentals

To carry out autonomous exploration and inspection surveys, robots must create a map of the environment that reflects its structure. To this end, the occupancy grid mapping paradigm [102], which was developed as a robust representation of the surrounding, has shown noteworthy results in the last years in several robotic domains and applications, ranging from collision checking and obstacle/collision avoidance [4], to planning [5], [77], and exploration [65], [68], just to mention a few. In the context of this work, to conduct the autonomous inspections, the FLS-based mapping methodology presented in [9] was employed and extended to include the ATR findings. Since it resorts to the occupancy grid mapping paradigm, this section presents the fundamentals of such a strategy.

An occupancy map tries to infer the map m from a set of measurements

$z(\cdot)$ acquired from the time 1 up to the time t, where $t \geq 1$, and denoted as $z_{1:t}$. Each measurement is collected from a robot pose that, as far as this work is concerned, is assumed to be known, as described in Section 3.1. For the sake of brevity, the dependence on the poses will be omitted in the following.

The occupancy grid theory relies on two fundamental assumptions:

- the 2D or 3D domain is partitioned through 2D/3D grid cells m_i, such that $m = \sum_i m_i$, which are considered as independent.Therefore, the problem of estimating m is divided into independent sub-problems. Intentionally, to make problem tractable, dependencies among neighboring cells are explicitly not considered;
- the Markovian assumption $p\left(z_t \mid m_i, z_{1:t-1}\right) = p\left(z_t \mid m_i\right)$ is employed, where $p(\cdot \mid \cdot)$ represents the conditional probability.

In the Bayesian context, the update law becomes:

$$\frac{p\left(m_i \mid z_{1:t}\right)}{1 - p\left(m_i \mid z_{1:t}\right)} = \underbrace{\frac{p\left(m_i \mid z_t\right)}{1 - p\left(m_i \mid z_t\right)}}_{\text{inverse sensor model}} \underbrace{\frac{p\left(m_i \mid z_{1:t-1}\right)}{1 - p\left(m_i \mid z_{1:t-1}\right)}}_{\text{recursive term}} \underbrace{\frac{1 - p(m_i)}{p(m_i)}}_{\text{prior}} . \tag{3.10}$$

As firstly suggested by [103], given log-odds ratio $l\left(\cdot\right)$, where

$$l\left(\cdot\right) = \log \frac{p\left(\cdot\right)}{1 - p\left(\cdot\right)} , \tag{3.11}$$

an elegant and numerically efficient update formulation for the occupancy grid problem can be obtained:

$$l\left(m_i \mid z_{1:t}\right) = l\left(m_i \mid z_{1:t-1}\right) + l\left(m_i \mid z_t\right) - l\left(m_i\right) . \tag{3.12}$$

It is worth to note that the term $p\left(m_i \mid z_t\right)$ is called Inverse Sensor Model (ISM). It depends on the exploited device, and as regards as the FLS used in this work, more details can be found in [9].

The OctoMap framework [66] was utilized to develop the 3D probabilistic occupancy mapping solution presented in [9] and exploited in this work. OctoMap is a well-known and efficient open-source C++ library for probabilistic map representation; it permits to model free $\mathcal{W}_{free}$, occupied $\mathcal{W}_{obst}$, and unmapped $\mathcal{W}_{uk}$ areas, which is fundamental for planning and exploring in unknown environments. Basically, OctoMap can deal with distance-like sensor fusing measurements into a voxel representation, where the information is stored in an octree data structure.

OctoMap, in its standard version, employs the non-informative prior assumption $p(m_i) = 0.5$, which is reasonably when mapping in completely unknown environments. Moreover, the clamping update policy is used, leading to:

$$\begin{aligned} & l\left(m_i \mid z_{1:t}\right) \\ & \quad = \max\left(\min\left(l\left(m_i \mid z_{1:t-1}\right) + l\left(m_i \mid z_t\right), l_{\max}\right), l_{\min}\right) \ , \end{aligned} \tag{3.13}$$

where $l_{\min} \in \mathbb{R}^+$ and $l_{\max} \in \mathbb{R}^-$ are the lower and upper bound on the log-odds value, respectively.

3.4 View planning preliminaries

3.4.1 AUV motion modeling and feasible path planning

This Section introduces the equations used to describe the motion of an AUV moving within a fluid and discusses how such equations can be integrated into a planning algorithm in order to generate feasible paths.

Generally, the pose of an AUV in 3D is expressed by means of six variables, and its motion can be expressed through kinematic and dynamic models. According to [83] and [104], by considering the AUV as a rigid body and by using the notation and reference frames introduced in Section 3.1, the AUV pose with respect to the NED frame (<N>) is expressed with $\boldsymbol{\eta} = \left[{}^N\boldsymbol{\eta}_1^\top \ \boldsymbol{\eta}_2^\top\right]^\top \in \mathbb{R}^6$, where ${}^N\boldsymbol{\eta}_1 \in \mathbb{R}^3$ indicates the position of the vehicle with respect to the NED frame and $\boldsymbol{\eta}_2 \in \mathbb{R}^3$ its orientation. The AUV linear and angular velocities with respect to the body reference frame ($< b >$) are represented with ${}^b\boldsymbol{\nu} = \left[{}^b\boldsymbol{\nu}_1^\top \ {}^b\boldsymbol{\nu}_2^\top\right]^\top$, where ${}^b\boldsymbol{\nu}_1^\top = [u \ v \ w]$, denotes the linear velocities and ${}^b\boldsymbol{\nu}_2^\top = [p \ q \ r]$ are the angular counterparts along the axes of the body frame, namely surge, sway, and heave. Thus, the vehicle kinematic model is expressed as:

$$\begin{pmatrix} {}^N\dot{\boldsymbol{\eta}}_1 \\ \dot{\boldsymbol{\eta}}_2 \end{pmatrix} = \begin{bmatrix} J_1\left(\boldsymbol{\eta}_2\right) & 0_{3\times3} \\ 0_{3\times3} & J_2\left(\boldsymbol{\eta}_2\right) \end{bmatrix} \begin{pmatrix} {}^b\boldsymbol{\nu}_1 \\ {}^b\boldsymbol{\nu}_2 \end{pmatrix}, \tag{3.14}$$

where

$$J_1(\boldsymbol{\eta}_2) = {}^N R_b^N = R_z(\psi) R_y(\theta) R_x(\phi) = \\ = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi c_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix}, \tag{3.15}$$

and

$$J_2(\boldsymbol{\eta}_2) = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & \frac{s_\phi}{c_\theta} & \frac{c_\phi}{c_\theta} \end{bmatrix}. \tag{3.16}$$

In a compact notation, Eq. 3.14 can be expressed as:

$$\dot{\boldsymbol{\eta}} = J(\boldsymbol{\eta})\boldsymbol{\nu}, \tag{3.17}$$

where

$$J(\boldsymbol{\eta}) = \begin{bmatrix} J_1(\boldsymbol{\eta}_2) & 0_{3\times3} \\ 0_{3\times3} & J_2(\boldsymbol{\eta}_2) \end{bmatrix}. \tag{3.18}$$

In several applications, AUVs usually conduct surveys at constant depth or altitude. As described in Section 1.2, vehicles such as FeelHippo AUV have the roll and pitch dynamics hydrostatically stable, and since seabed inspection surveys, considered in this work, do not excite these DOFs, the AUV navigates with roll and pitch angles almost zero with negligible variations. For the sake of completeness, the roll motion is not controlled. Additionally, to reduce the energy consumption, FeelHippo AUV usually moves about the surge axis and only rotates about the heave axis, i.e., it only changes its orientation. As a consequence of these statements, the kinematic motion on a horizontal plane of FeelHippo AUV can be represented as a simple car-like vehicle:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} u\cos(\psi) \\ u\sin(\psi) \\ r \end{bmatrix}. \tag{3.19}$$

While kinematic models only use geometric equations that relate the vehicle positions and velocities, dynamic models consider the force and torques that create the motion. As a consequence, this latter class of models describes the AUV motion constraints more accurately. Further information about these models can be found in [83]. As far as this work is concerned, since the view planning problem was tackled, only the kinematic model was considered for the following reasons. Firstly, dynamic models require a higher computational load and more time than the kinematic counterpart to evolve the system [77]. That is, fewer new configurations can be generated given a computational time, and thus worse solutions are found. As discussed in the following Chapters, evaluating the visibility gain of a configuration is the most demanding operation in terms of time and computational load that limits the number of configurations that can be generated. Therefore, to generate as many configurations as possible within a given computation time slot (in order to compute good inspection paths), a kinematic model has been preferred.

Additionally, and most importantly, inspired by the noteworthy RRT* [58] algorithm, the use of a rewiring strategy, which consists of a routine that checks whether a new node could improve the cost of neighbour nodes, was investigated in this work. To this end, a steering function, which returns the optimum path between two states, is needed. When using motion constraints, computing a steering function means addressing a two-point Boundary Value Problem (BVP). That is, it corresponds to solving a differential equation under certain boundary conditions [105], which is generally a difficult problem.

Describing the AUV motion with kinematic model of Eq. 3.19, the configuration space ($\mathcal{C}$) is $\mathcal{C} = SE(2)$, i.e., a configuration $\xi \in SE(2)$. Assuming that the vehicle navigates with a constant surge speed u, and by considering that it has a maximum turning rate r_{max}, which defines a minimum turning radius R_{min}, the AUV kinematic motion can be described with the Dubins vehicle model [106]. In particular, by using this model, the shortest path between two configurations consists of circular arcs of maximum curvature and straight line segments. Thus, the shortest path can be obtained by combining three possible maneuvers: straight (S), right turn (R), and left turn (L). Thus, the shortest path will always be at least one of the six combinations: RSR, RSL, LSR, LSL, RLR, LRL. The Dubins vehicle model is of particular interest in planning tasks since it can be used for both generating new configurations, which in RRT-based solutions corresponds to expanding the tree, and works as a steering function for the rewiring routine, see [58] and [77] for more details.

In conclusion, as a consequence of these considerations, for planning feasible paths for the AUV, its motion was modeled by using the Dubins vehicle model. Finally, it is worth highlighting that the AUV model selection has by no means been influenced by the considerations and assumptions regarding the sea bottom profile discussed in Section 3.1.

3.4.2 View planning and information gain metrics

In this work, the AUV is supposed to perform a constant altitude mission; thus, the coverage algorithm was designed to compute the next waypoint constituted of a 2D position (x^N, y^N) in the NED frame and orientation (ψ), which represents the vehicle heading angle. From these considerations, a robot viewpoint (or configuration) is defined as $\xi \in \Xi$, with $\xi = (x^N, y^N, \psi)$, and the considered problem is the following: computing online the best path according to a cost function, that let the AUV map the workspace $\mathcal{W}$, that is initially unknown $\mathcal{W}_{uk} = \mathcal{W}$, and classify it as free $\mathcal{W}_{free}$ or occupied $\mathcal{W}_{obs}$. The occupancy map paradigm, described in Section 3.3, is employed to determine the NBV and monitor the coverage progresses. Consequently, the coverage algorithm solves the problem of computing the waypoints that allow to estimate $m = \sum_i m_i$.

Using the occupancy mapping strategy (Section 3.3), the visibility of a viewpoint ξ is defined as the set of visible voxels that are computed through a ray casting process. The sensor characteristics define the set of rays $\mathcal{R}_\xi$ for every viewpoint; each ray $\mathfrak{r}$ ends when it reaches the maximum sensor range or the limit of the map or it hits an occupied voxel. Traversing the map, a ray visits a set of voxels $\mathcal{X}_\mathfrak{r}$. The obtainable Information Gain (IG) from a viewpoint, denoted as $\mathcal{G}_\xi$ is estimated as [62]:

$$\mathcal{G}_\xi = \sum_{\forall \mathfrak{r} \in \mathcal{R}_\xi} \sum_{\forall x \in \mathcal{X}_\mathfrak{r}} I(x) \, , \tag{3.20}$$

where $I(x)$ denotes the Volumetric Information (VI) contained in the voxel x.

In this paper, a comparison of information gain metrics for seabed inspections is reported. In particular, two VI formulations were considered: the first one, denoted as volume-based VI in the following, considers the volume of unmapped cells, while the second one, called entropy-based VI, uses the concept of entropy to take into account the map uncertainty. Similarly to [65], when exploiting the volume-based VI, viewpoints' IG is computed by considering only the volume of unmapped cells, yielding to:

$$I_v(x) = \begin{cases} \mu(x) & \text{if } x \in \mathcal{W}_{uk} \\ 0 & \text{otherwise ,} \end{cases} \tag{3.21}$$

where $\mu(x)$ denotes the volume of the voxel. OctoMap was used to develop the FLS-based 3D probabilistic occupancy mapping solution described in [9] and utilized in this work. Since it employs the non-informative prior assumption (Section 3.3), the VI can be written as:

$$I_v(x) = \begin{cases} \mu(x) & \text{if } 0.5 - \delta \leq P_o(x) \leq 0.5 + \delta \\ 0 & \text{otherwise ,} \end{cases} \tag{3.22}$$

where $P_o(x)$ is the voxel probability of being occupied, and δ is a parameter that depends on the mapping sensor accuracy.

Moreover, the occupancy grid mapping paradigm allows considering the map uncertainty easily. Assumed $\overline{P}_o(x) = 1 - P_o(x)$, an entropy-based VI formulation can be defined:

$$I_e(x) = -P_o(x)ln(P_o(x)) - \overline{P}_o(x)ln(\overline{P}_o(x)) \ . \tag{3.23}$$

Utilizing this VI formulation, the highest uncertainty matches unknown voxels, that have the occupancy probability of $P_o(x) = 0.5$.

Chapter 4

CNN-based Automatic Target Recognition

This chapter reports the development of the proposed methodology for making FeelHippo AUV able to effectively detecting OPIs autonomously during an inspection survey. Firstly, the speed/accuracy trade-off of modern CNN image-based object detectors is presented, and important desired characteristics of the developed ATR methodology for FLS imagery are discussed. On all these considerations, which are of utmost importance for selecting a proper architecture, the developed CNN-based ATR solution that is then presented is based. Then, the networks training and validation details are discussed, and their performance is evaluated. Finally, the experimental results of the conducted real sea trials are reported.

4.1 Proposed ATR methodology

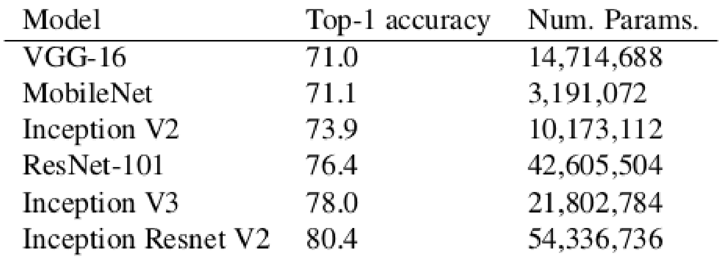
This Section investigates the development of a DL ATR strategy for FLS imagery that can run real-time on compact AUVs with limited hardware capabilities. In particular, since the effectiveness of image-based state-of-the-art CNNs on FLS images was shown in previous works [43] [25], this research focuses on a practical application of such CNN techniques. Besides, even though gathering a large and heterogeneous underwater dataset is time and cost consuming, the aforementioned state-of-the-art DNNs allow the use of transfer learning approaches that speed up the ATR development by fine-tuning the final network layers while the backbone is not modified. That is, the use of network weights pre-trained on a common dataset, such as the COCO dataset, even if it contains images that do not present any similarities with the considered FLS images, is investigated. As a result, a network model does not require thousands of images to be trained on a custom dataset, and the learning process is speeded up and simplified.

Therefore, the desired characteristics and the selection criteria for the CNN model are firstly discussed. Then, the framework used for training the selected network and the developed ATR methodology for using the CNN onboard on FeelHippo AUV is presented.

4.1.1 Model selection

As far as the speed/accuracy trade-off of CNN detector is concerned, an exhaustive investigation was presented in [48]. In particular, the authors highlighted that some modern state-of-the-art object detectors follow very similar designs; that is, they are composed of a backbone that extracts the features and final layers in charge of localizing and classifying the objects. Thus, such structure makes these CNNs modular, which means that, as proposed in [48], they can be considered as meta-architectures where different feature extractors can be exploited. Since the accuracy, measured as the mAP, is not the only parameter that shall be considered when developing an ATR solution for real deployments, the study proposed in [48] helps in selecting and testing only the most promising CNN architectures. The authors analyzed three meta-architectures (i.e., SSD, Faster R-CNN, and Region-based Fully Convolutional Network (R-FCN) [107]) with six feature extractors (see Fig. 4.1, coming from [48]), and compared their performance in terms of accuracy, speed, and memory demand. The speed/accuracy trade-off is summarized in Fig. 4.2, where the dotted gray line represents the optimality frontier, i.e., where the best detectors can be found. According to Fig. 4.2, taken again from [48], the meta-architecture performance is strongly connected with the choice of the backbone, but in general terms, the Faster R-CNN is more accurate, while the SSD and R-FCN are faster. Nevertheless, it is worth noting that by limiting the number of regions proposed, the Faster R-CNN can be fastened.

When it comes to select the most appropriate network for the recognizing objects in FLS imagery onboard real-time, some relevant points must be considered. Firstly, the acoustic frames are provided by a Teledyne BlueView M900 2D FLS that acquired at a low frame rate (3 Hz). Therefore, an



Model	Top-1 accuracy	Num. Params.
VGG-16	71.0	14,714,688
MobileNet	71.1	3,191,072
Inception V2	73.9	10,173,112
ResNet-101	76.4	42,605,504
Inception V3	78.0	21,802,784
Inception Resnet V2	80.4	54,336,736

Figure 4.1: The properties of the feature extractors used in [48] for the speed/accuracy investigation. Top-1 accuracy is the classification accuracy metric used on ImageNet [108].

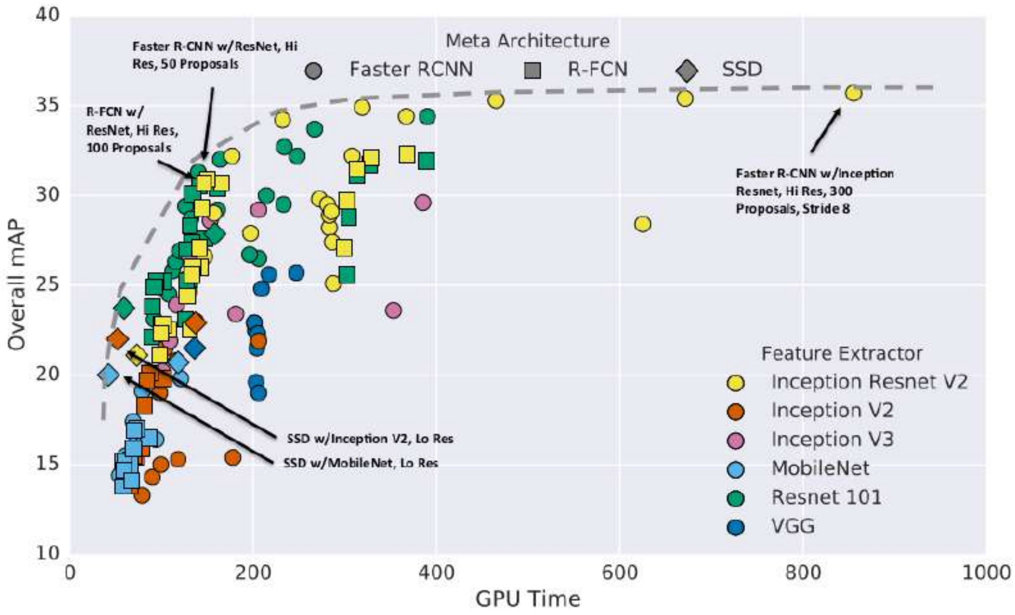


Figure 4.2: The speed/accuracy trade-off reported in [48].

extreme inference rate, such as YOLO, is not required. Moreover, the ATR solution has to provide additional geolocalization of possible seabed objects; within this context, since the target 3D positions are estimated from the 2D DNN localization in the FLS frame, minor errors in the bounding boxes at the pixel level could lead to large errors in meters in the 3D localization. Therefore, the network accuracy is of utmost importance and shall favor the inference speed as the model selection parameter. However, since the goal of this work was the development of a CNN-based ATR methodology for real-time onboard applications, the inference speed cannot be ignored; indeed, the ATR solution must analyze 3 fps, i.e., all the frames gathered by the Teledyne BlueView M900 2D FLS.

As a consequence of these statements, the Faster R-CNN represents a functional selection for the developed ATR solution as it is one of the most accurate CNN architectures and can meet real-time requirements. In detail, according to the optimilaity frontier of Fig. 4.2, the Faster R-CNN based on ResNet and a limit of 50 proposal was selected for the investigations conducted in this work. Moreover, since this research activity proposes a preliminary investigation of an ATR solution for self-contained onboard applications, the required computational resources are of utmost importance, and the efficient SSD with Mobilenet V2, designed for mobile and embedded devices, has also been tested. In fact, as shown in Fig. 4.2, SSD Mobilenet network is the fastest and computationally lightest but yet accurate solution.

Therefore, in conclusion, the following models were considered for developing the here proposed CNN-based ATR methodology for FLS images:

- Faster R-CNN with ResNet, 50 proposals;

- SSD with Mobilenet V2.

4.1.2 ATR Framework

To exploit the selected CNN models, they must be trained on a custom dataset, and they could be deployed on the dedicated AUV hardware. To this end, the open-source ML library TensorFlow [109], which has become the standard for AI applications and has been chosen by several top-level companies, has been selected. TensorFlow offers multiple levels of abstraction to easily and rapidly develop ML solutions. It provides a high-level Application Programming Interface (API) that allows to build and train custom models. Besides, a wide variety of state-of-the-art models and training dataset repositories are available. Hence, exploiting TensorFlow, it is possible to either train state-of-the-art NN models or design custom solutions by plugging together building blocks, which are network layers. Besides, TensorFlow is supported by the NVIDIA hardware, the golden standard for AI and DL developments, for both training and real deployments.

Deployment on FeelHippo AUV

As regards the real deployment of the CNN architectures, a custom ATR software was developed. It uses the TensorFlow libraries to handle the trained models, which were integrated within the ROS framework [35] to manage the input and output data.

When a new FLS image is acquired, it is analyzed by the ATR software by means of the trained CNN model. As described in Section 3.2.2, the network outputs the predicted object classes, with the computed confidence level and the object bounding boxes. Then, the bounding boxes are used to localize the detected objects in the NED frame, as explained in Section 3.1, by using the FLS mounting pose with respect to the vehicle and the AUV estimated pose. To run the developed CNN-based ATR solution, an NVIDIA Jetson Nano [110] was mounted on FeelHippo AUV (see Section 1.2). The NVIDIA Jetson Nano, depicted in Fig. 4.3, is a platform for embedded AI computing that belongs to the NVIDIA Jetson series, specifically designed to meet the power consumption and limited space requirements of autonomous machines. It acted as a dedicated payload computer that was connected to the FeelHippo AUV main computer, in charge of running onboard processing navigation and control algorithms and supervising the state of the vehicle, through an Ethernet cable. In particular, the NVIDIA Jetson Nano is the smallest hardware of the Jeston series and was favoured since it could fit into the limited available space on FeelHippo AUV.

Figure 4.3: The NVIDIA Jetson Nano mounted on FeelHippo AUV for running the developed CNN-based ATR algorithm.

4.2 Training and validation

4.2.1 Training

Since in this work the focus has shifted on fine-tuning pre-trained networks, the CNN model training procedure can be performed on a laptop. In detail, the selected SSD and Faster R-CNN models were trained on a laptop fitted with 16GB RAM, an Intel Core i7-8750H processor, and an Nvidia GeForce GTX 1050 Ti card.

For what concerns the training details, the methodology proposed by the authors of the Faster R-CNN and SSD was followed. The SSD network was trained using the optimizer RMSProp [97] and a batch size of 24. On the other hand, the Faster R-CNN model was fine-tuned by using the SGD optimizer with momentum [96] with a batch size of 1. According to [44] and [49] respectively, the learning rate schedules have been defined explicitly for each CNN architecture to accomplish optimal inference outcomes and a fast convergence timing.

Once the training has been performed, the CNN models can be saved; then, they can be used in the ATR software. A trained model consists of a configuration file describing the network architecture, and weights files, binary files that include the trained network weights. Following this lead, a general-purpose ATR software can be developed; thus, the target objects are defined by the selected trained network.

4.2.2 Dataset gathering

The training dataset was gathered with FeelHippo AUV during on-field trials, performed in May 2019, at the CSSN basin, La Spezia, Italy. The proposed ATR solution was utilized to detect OPIs on the seabed; as depicted in Fig. 4.4, target OPIs have different shapes and dimensions, and their rendering on FLS images is strictly related to the sonar viewpoint.

Among the recorded FLS images, 175 frames, in a native resolution of 894 x 477 pixels and containing one or more detectable targets, had been selected. In particular, it is worth noting that this procedure has been fulfilled in the view of collecting a diversified, heterogeneous dataset. Consequently, images with different informative regions have been taken into account. Although it may be considered an evident and negligible pattern, this design guideline plays a fundamental role in providing the CNN architecture with an optimal generalization capability. Furthermore, coherently with the aim of building a robust dataset, data augmentation options of the TensorFlow framework were used to expand the dataset and make it more heterogeneous. In particular, the dataset was augmented by randomly horizontally flipping the images and randomly varying their brightness.

In conclusion, the Faster R-CNN and the SSD models were trained using the TensorFlow framework with this dataset as explained in Section 4.1. The training was stopped for both the networks at about 10k steps, where the validation loss started to increase consistently (Section 3.2).

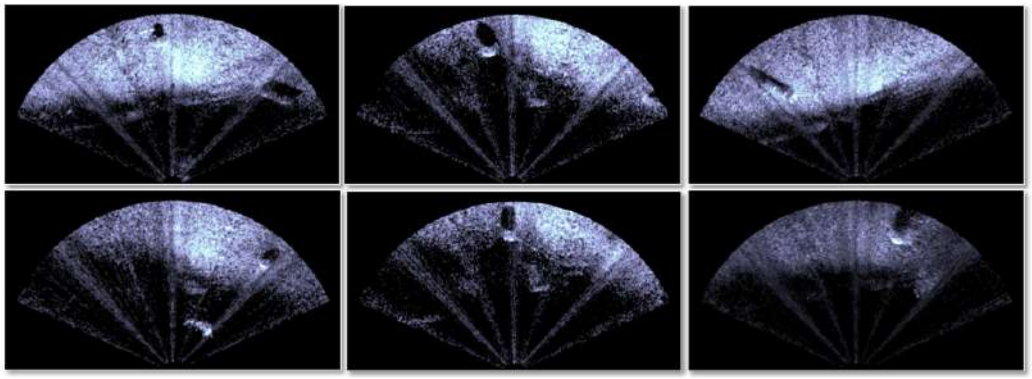


Figure 4.4: Examples of FLS images used in the training dataset. The selected CNN architectures were trained to detect and localize the depicted OPIs . It is worth noting that, in order to get a heterogeneous ATR solution, the OPI forms and sizes vary while affecting their rendering, which also depends on the FLS viewpoint.

4.2.3 Validation and performance assessment

To assess the performance of the trained networks, a validation dataset was made. In particular, the validation dataset was constituted of 200 FLS images acquired in October 2019 at the CSSN basin. During the campaign, multiple surveys with FeelHippo AUV flying at different altitudes from the seabed were conducted in order to collect a diversified set of images. Thus, the validation dataset was constituted of images depicting zero, one or more detectable targets, enlightened from several sonar viewpoints.

During the training, the CNN models were evaluated by using the validation dataset and the TensorFlow framework. The accuracy of the networks was computed through the mAP metric, and the results are reported in Fig. 4.5. Fig. 4.5 depicts the mAP considering the IoU thresholds of 0.5 (bottom-left) and 0.75 (bottom-right). It also shows the mAP averaged over IOU thresholds from 0.5 to 0.95 with a step size of 0.05, which is the metric officially used in the COCO dataset.

As expected, according to [48] (see Section 4.1), the Faster R-CNN model resulted as more accurate than the SSD. The maximum mAP value reached by the Faster R-CNN is 0.451, whereas the SSD could not overcome 0.345. In fact, as shown in Fig. 4.6 that reports two examples of the detections performed during the evaluation procedure, the predicted bounding boxes (represented in green) of the Faster R-CNN, on the left of the picture, are closer to the provided ground-truths (represented in red) than the SSD model, whose predictions are reported on the right.

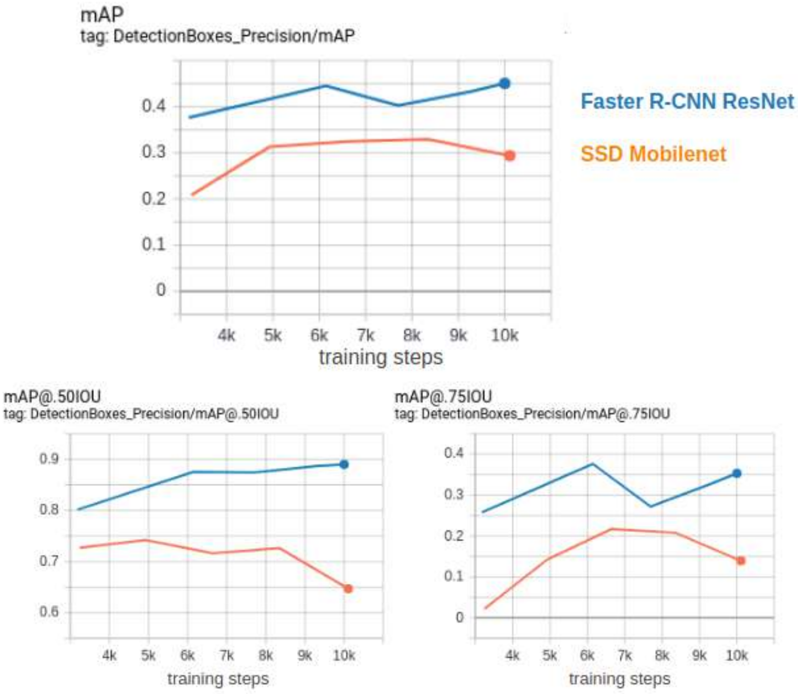


Figure 4.5: The mAP at different IoU thresholds computed during the training of the Faster R-CNN, in blue, and the SSD, in orange. The mAP was evaluated by using the validation dataset.

Nevertheless, since this research work focuses on developing a self-contained

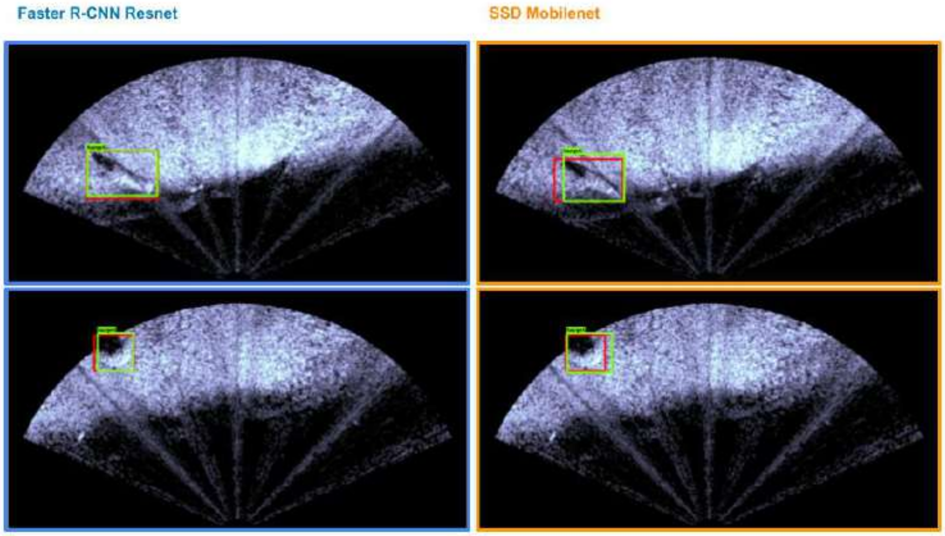


Figure 4.6: Two examples of the detections performed by the trained CNN models during the evaluation. On the left, the predictions of the Faster R-CNN, whereas, on the right, the detections of the SSD model. The green rectangles represent the computed bounding boxes. The provided ground-truths are reported in red.

ATR methodology capable of running on compact AUVs, the mAP is not the only parameter that shall be taken into account. The inference speed is of utmost importance to establish whether the ATR strategy can analyze all the acquired images in real-time, and the available hardware plays a key role. As described in Section 4.1, FeelHippo AUV acquires 3 FLS images per second by means of a Teledyne BlueView M900 2D FLS. Thus, CNN-based ATR must achieve such a working rate, which defines a real-time requirement. Therefore, the inference speed, measured as fps, of the developed ATR algorithm by utilizing both the trained CNNs was assessed. Firstly, by using pre-recorded data, the inference speed of the ATR algorithm was evaluated with both the CNNs on the laptop used for training, that is fitted with 16GB RAM, an Intel Core i7-8750H processor, and an Nvidia GeForceGTX 1050 Ti card. As reported in Table 4.1, with both the Faster R-CNN and the SSD, the ATR strategy achieved the online recognition requirements by analyzing more than 3 fps. Then, the inference speed was evaluated offline on FeelHippo AUV hardware to verify whether the developed solution could be used on hardware with limited computational resources, as the NVIDIA Jetson Nano. To this end, both the aforementioned ATR strategies were run on the NVIDIA Jetson Nano mounted on FeelHippo AUV, while its main computer was used to stream the pre-recorded data. This setup allowed to simulate real experiment working conditions. The SSD Mobilenet network, designed for mobile and embedded devices, managed to analyze up to 8 fps. On the other hand, the more accurate Faster R-CNN with ResNet has a higher computational load and resulted slower than the SSD model, reaching just 0.95-1 fps. Since, as explained above, the acoustic frames were captured by a Teledyne Blueview M900 2D FLS at 3 Hz, the Faster R-CNN network could not fulfill the online recognition requirements. Given that the OPIs rendering on FLS images strictly depends on the insonification position and angle, to the best of the author's knowledge, it is of utmost importance to analyze all the collected images to avoid false negatives. As a consequence of this analysis, only the SSD Mobilenet network, which can run at a satisfying frame rate, was considered for the sea trials described in the next Section.

Table 4.1: ATR performance evaluation

		Inference speed	
		Laptop Dell G5	**FeelHippo AUV**
CNN Model	**mAP**	**NVIDIA 1050 Ti (4 Gb GPU)**	**NVIDIA Jetson Nano**
Faster R-CNN ResNet	0.451	8–10 fps	0.95-1.0 fps
SSD Mobilenet	0.345	25-30 fps	6-8 fps

4.3 Experimental results

To validate the developed ATR methodology, an experimental campaign was conducted in October 2020 at the CSSN, La Spezia, Italy. According to the performance evaluation reported in the previous Section, since the Faster R-CNN could not reach a satisfying frame rate in the self-contained ATR solution, only the trained SSD Mobilenet network was tested during the experimental campaign. It is worth highlighting that the network was trained with the dataset acquired in May 2019, and to provide the whole experiment with an improved generalizing context, when the experimental campaign was conducted in October 2020, the OPIs were replaced and relocated. Since the testing site was an unknown environment and the OPIs ground-truth localization was not provided, the accuracy of the ATR methodology was assessed. To this end, an OPI (resembling a truncated cone) was deployed in a known position in the CSSN basin. FeelHippo AUV was used to enlighten the OPI with the Teledyne Blueview M900 2D FLS, and the ATR solution was run with the localization technique detailed in Section 3.2. In particular, the OPI position was estimated by using both the FLS standard 2D approximation (Eq. 3.4) and the 3D version of Eq. 3.6, which makes use of additional assumptions that hold for the testing site (see Section 3.2). While with the former approach, the ATR solution achieved a localization error of about 2.5 m (computed considering a single view), the latter resulted more accurate with an error of less than 2 m (about 1.3 m). It is worth noting that considering the underwater navigation system accuracy (in position and orientation, in particular, the heading angle), localization errors below 2 m can be considered accurate.

Thus, FeelHippo AUV was used to inspect the unknown region with the ATR strategy. It conducted several autonomous surveys following lawnmower paths at constant altitudes. An example is shown in Fig. 4.7. As shown in Fig. 4.8, during the surveys, the ATR solution with the SSD Mobilenet network, which was running on the NVIDIA Jetson nano, managed to detect and localize several OPIs of various forms and sizes online. Turning to quantitative analysis, during the survey, the ATR methodology provided 61 detections. Afterward, a human operator analyzed the ATR outputs in a post-processing stage that allowed to classify the detections as true positives and false positives. In detail, 59 detections were true positives, while only two images were misidentified and were classified by the operator as false positives. Besides, the human operator, by analyzing the ATR detections, managed to identify 9 different OPIs. However, as stated above, since a precise OPI map of the underwater scenario has not been provided, a strict quantitative evaluation of the OPI localization metrics cannot be reported. Moreover, the actual OPIs number was not available, and thus it is not possible to verify whether all OPIs were detected or not.

This experimental campaign demonstrated that thanks to the developed ATR methodology, FeelHippo AUV is able to identify and localize OPIs online onboard during the survey. That is the first fundamental step for making a compact AUV, such as FeelHippo AUV, able to effectively understanding and gathering knowledge of the environment and thus, making it fully autonomous and able to accomplish interactive tasks. Finally, it is worth mentioning that although the OPIs utilized in the testing site for the experimental campaign were replaced and relocated with respect to the ones used for training the ATR network, objects of comparable size but different shapes have been employed. For the sake of clarity, the here presented ATR solution was utilized for detecting and localizing possible OPIs in FLS images, i.e., for determining the presence and the position of the objects, and not for classifying the different object types.

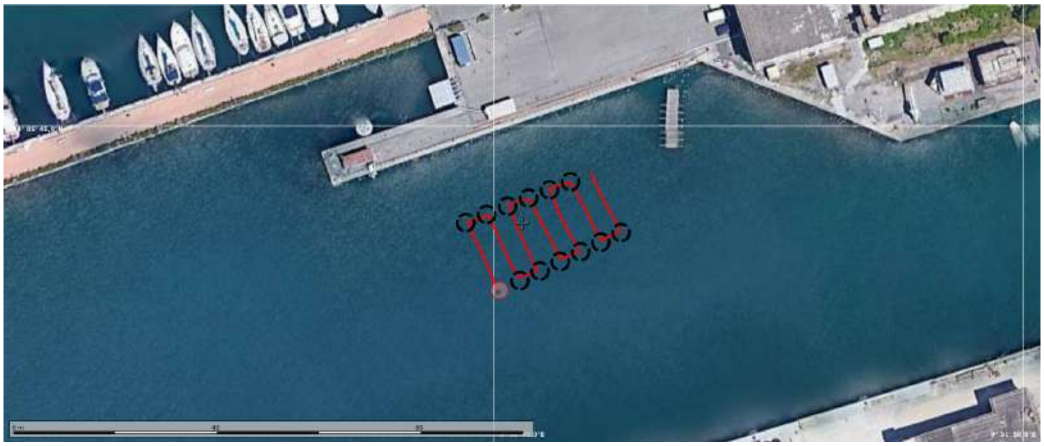


Figure 4.7: One of the lawnmower surveys performed by FeelHippo AUV during the experimental campaign in October 2020 at the CSSN, La Spezia, Italy. In a red line, the estimated FeelHippo AUV traveled path.

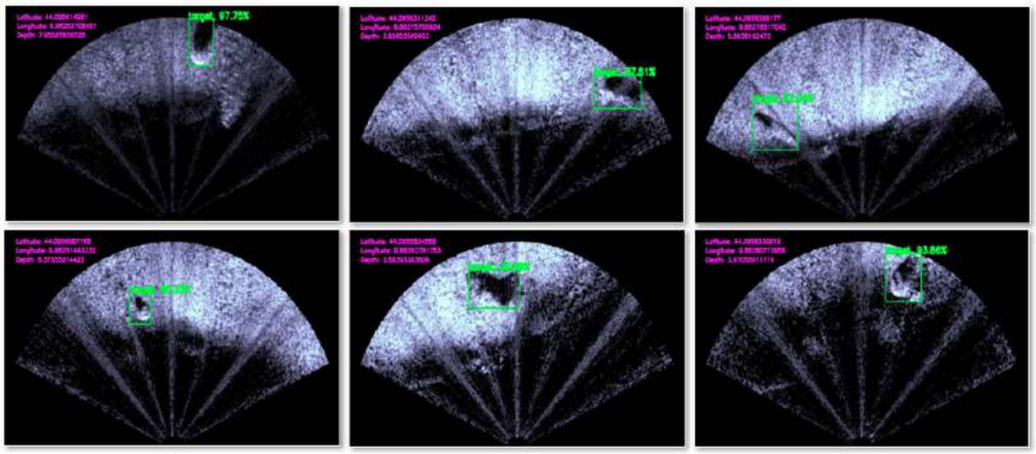


Figure 4.8: Examples of the online detected and localized OPIs by means of the developed self-contained ATR with the SSD network during the experimental campaign conducted in October 2020 at the CSSN, La Spezia, Italy. In purple the localization outcomes in a Latitude-Longitude-Depth representation whilst in green the bounding box traced around the OPIs as well as the class label and the detection probability.

4.4 Main contribution

This Chapter presented the developed CNN-based ATR architecture for compact AUVs. The solution was designed to make a small and compact AUV, such as FeelHippo AUV, able to effectively detecting OPIs autonomously in FLS images during an inspection survey.

Firstly, important desired characteristics that have driven the development of the ATR methodology were presented, and the speed/accuracy tradeoff of modern CNN image-based object detectors was discussed. Such aspects are of utmost importance for selecting the proper CNN architecture and defining the requirements of an ATR solution. As a consequence of these considerations, two state-of-the-art ATR architectures were selected for detecting OPIs in FLS images: the Faster R-CNN with ResNEt and the SSD with Mobilenet. Thus, the developed architecture for training and deploying a proper network was detailed.

Regarding the training process, a dataset constituted of 175 raw FLS images was collected in May 2019 at the CSSN basin in La Spezia, Italy. To validate and assess the performance of the trained networks, a new validation dataset was constituted of 200 FLS images acquired in October 2019 on the same site. Thus, during the training, the CNN models were evaluated. The accuracy of the networks was computed through the mAP metric, while the inference speed was evaluated by testing the trained networks on an NVIDIA Jetson Nano mounted on FeelHippo AUV. The Faster R-CNN model resulted more accurate than the SSD. However, only by using the SSD network, the developed ATR solution managed to achieve the online recognition requirements, i.e., analyzing more than 3 fps.

Finally, in October 2020, the proposed ATR methodology was validated in real tests at sea conducted at the CSSN basin, La Spezia, Italy. To provide the whole experiment with an improved generalizing context, when the experimental campaign was conducted, the OPIs were replaced and relocated. It is worth highlighting that since the Faster R-CNN could not reach a satisfying frame rate in the self-contained ATR solution, only the trained SSD Mobilenet network was tested at field. Thanks to the developed ATR methodology, FeelHippo AUV was able to identify and localize online onboard during the survey the present OPIs. In conclusion, the achieved results have highlighted the capability of the proposed architecture to autonomously inspect an unknown underwater scenario by effectively detect and localize targets of potential interest.

Chapter 5

Sensor-driven RRT-based receding-horizon coverage approach for autonomous inspections

The developed sensor-driven RHCA for autonomous inspections is described in this section. Firstly, the framework structure and implementation details are reported. Then, the developed random tree-based coverage algorithm is described and analyzed. The effect of the different VI formulations described in Section 3.4.2 is investigated through a theoretical analysis which is then validated by the obtained results. In detail, simulations and experimental campaigns were conducted to validate and assess the performance of the developed coverage strategy using different VI formulations. The main findings reported in this Chapter were published in [111].

5.1 Sensor-driven receding-horizon framework

The here proposed coverage solution, depicted in Fig.5.1, is composed of four elements: the Mission Manager, the Mapping module, the High-level planner, and the Motion planner. These modules run onboard on the AUV and were integrated within the ROS framework [35]. Additionally, a Graphical User Interface (GUI) that runs on a laptop computer was developed. It enables an operator or a science user to interact with the AUV. As shown in Fig.5.2, the GUI allows to easily define the inspection area, which is sent to the AUV through a radio or WiFi link.

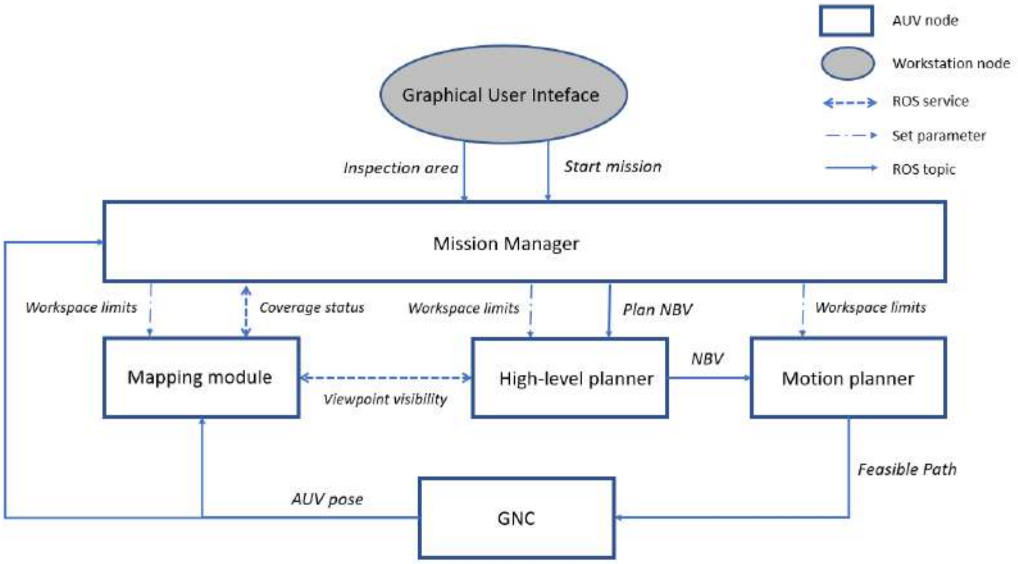


Figure 5.1: The developed inspections framework. In the context of this research, the framework was used to perform FLS seabed inspections. The operator can define the inspection area through a dedicated GUI. The Mission Manager works as an interface with the GUI, sets the workspace limits, and monitors the mission progresses. The Mapping module creates and updates the map using an exteroceptive sensor. The Mission Manager triggers the High-level planner to plan the NBV to accomplish the task. Then, the selected NBV is set as the goal configuration for the motion planner, in charge of computing a feasible path for the AUV. Finally, the GNC block includes the Guidance, Navigation, and Control strategies.

The Mission Manager uses the selected inspection area to limit the workspace of other modules. It triggers the High-level planner to compute the NBV when it receives the start command from the GUI, or the vehicle has reached the previous viewpoint. It also monitors the coverage progresses.

The Mapping module creates and updates an occupancy map. It takes as input the AUV estimated position and orientation and the gathered FLS images. A detailed description of the exploited 3D probabilistic occupancy mapping system for FLS reconstructions can be found in [9]. Briefly, when

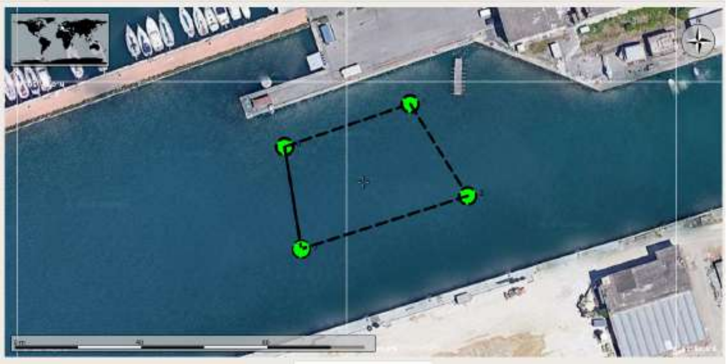


Figure 5.2: The developed GUI used for selecting the inspection area, limited by the four green points and the black dashed lines.

a new image is acquired, the module utilizes a front-end based on a segmentation techniques on acoustic images to generate a 3D point cloud of the seafloor and protruding visible objects, overcoming the information loss that occurs during the 3D to 2D image projection. Then, in the back-end the occupancy map is updated. It is worth highlighting that this mapping module was developed to provide feedback to the planning module on the area covered by the utilized imaging SONAR, i.e., the enlightened seabed, and not for creating an accurate reconstruction.

The High-level planner uses the view planning algorithm detailed in Section 5.2. Starting from the AUV position, the view planner builds a random tree to determine viewpoint candidates. For each viewpoint candidate, its visibility is evaluated according to the updated map with a ray casting strategy. To this end, the planner uses a ROS service to request the Mapping module the viewpoint visibility, which replies with the list of visible voxels and their occupancy probability. Thus, the expected information gain along the branch can be evaluated by calculating the voxels discovered along a tree branch and using one of the metrics proposed in Section 3.4.2. The algorithm selects the branch that is expected to collect the highest gain. The first node is extracted as the NBV, while the rest of the branch is stored, and it is used to initialize the tree when the Mission Manager requests a new NBV.

The computed NBV becomes the goal configuration of the Motion planner. In order to realize a multi-purpose framework that could be used for different tasks, the role of this module is twofold. In the context of this research, it is used to generate the Dubins path that connects the AUV po-

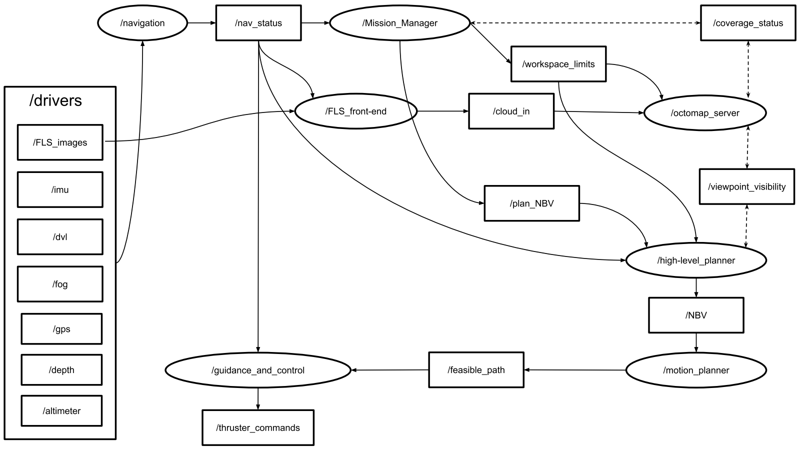


Figure 5.3: A visualization of the ROS computation graph of the developed inspection framework. Oval frames represent the ROS nodes, while squared frames depict the topics, and the continuous one-way arrows visualize the flow of information in the topics. Finally, the bidirectional dashed arrows represent the implemented ROS services.

sition to the target NBV. In addition, it could be employed to calculate the best feasible path according to the vehicle motion constraints and a cost function. For instance, in future developments, a planning algorithm to find a path that minimizes the expected localization and mapping uncertainty as proposed in [64] could be exploited.

Finally, the GNC module includes the guidance, navigation, and control strategies that the AUV uses to estimate its position and track the planned path. This module is reported for showing how the developed framework is connected to the software architecture of an AUV, but it is not in the scope of this paper. Further information regarding the exploited GNC solutions used by FeelHippo AUV, selected as the testing platform, can be found in [112] and [87]. For the sake of completeness, the ROS computation graph of the developed inspection framework is reported in Fig.5.3.

This simple structure gives the High-level planner continuous feedback of the mapped environment and enables the use of a receding-horizon approach that allows using this solution in unknown environments and helps to reduce the coverage errors. Therefore, it realizes a receding-horizon sampling-based sensor-driven view planning strategy for autonomous inspections.

5.2 RRT-based view planning

As described in Section 5.1, the High-level planning module computes online the path to accomplish the coverage survey. In the framework proposed here, whenever the Mission Manager requests to plan the next waypoint, the High-level planner computes the NBV resorting to a RRT inspired coverage algorithm. The developed algorithm, summarized in Algorithm 1, takes as input the AUV actual configuration. A new tree $\mathcal{T}$ containing the AUV configuration is initialized (line 3). If the algorithm had already been activated, the remainder of the previous call solution, i.e., the remainder of the best branch, is added to the tree, and its expected gain is evaluated according to the new updated map (lines 5-7). Then, until the terminal condition, which is the maximum planning time t_{max} is reached, the algorithm expands the tree $\mathcal{T}$: it randomly samples a new configuration ξ_s in the workspace with a uniform distribution, the nearest node in the tree is retrieved ξ_n, and a new configuration ξ_{new} is computed by propagating ξ_n along the direction to ξ_s with a random step (lines 8-10). If the generated new node ξ_{new} is inside the inspection area and the motion $\xi_n \rightarrow \xi_{new}$ is valid (line 11), ξ_{new} is added to the tree $\mathcal{T}$ (line 12). The algorithm requests to the Mapping module the visibility of ξ_{new}, which responds with the observable voxels computed by using a ray casting procedure (line 13). It is worth noting that the visibility of a viewpoint does not depend on the parent of the node; only its gain is affected by the choice of the parent. In fact, to correctly evaluate the expected viewpoint gain, the voxels already seen along the branch shall not be considered. This simple consideration constitutes the rewiring strategy. Once the viewpoint visibility has been computed, the algorithm looks for the best parent, i.e., the parent that maximizes the gain, among the nearest nodes in the tree. Therefore, the list of the k-nearest nodes of ξ_{new} is retrieved (line 14), and for each node in the list ξ_p, if the motion $\xi_p \rightarrow \xi_{new}$ is valid (line 16), the branch gain assuming ξ_p as the parent is computed (line 17). To calculate the branch gain, the viewpoint IG of Eq. 3.20 was extended to define the Branch Information Gain (BIG) $G_{\mathcal{B}}$ as follows:

$$G_{\mathcal{B}} = \sum_{\forall x \in \mathcal{X}_{\mathcal{B}}} I(x) \; , \tag{5.1}$$

where $\mathcal{X}_{\mathcal{B}}$ denotes the observable voxels along the branch $\mathcal{B}$, and $I(x)$ is the voxel VI, which is calculated according to Eq.3.22 or Eq.3.23.

Then, the branch gain is computed combining the BIG with penalizing factors. Since this work focuses on FLS seabed inspections, long and curvy paths were penalized:

$$\mathcal{G}_{\mathcal{B}} = G_{\mathcal{B}} e^{\lambda_{\psi}(\Delta\psi(\xi_0,\xi_k))} e^{\lambda_d(distance(\xi_0,\xi_k))} , \tag{5.2}$$

where $\lambda_{\psi} \in R$ penalizes curvy paths by considering the heading changes, and $\lambda_d \in R$ penalizes long paths.

Finally, the branch that is expected to collect the highest information gain is selected as the solution (lines 18-21) and stored to initialize the next call (line 22). The first node becomes the NBV that is sent to the Motion planner (line 23).

The presented coverage algorithm was developed using the Open Motion Planning Library (OMPL) [113], and was implemented using the C++ programming language and integrated within the ROS framework.

Algorithm 1 Coverage planner algorithm

```
Input: AUV configuration $\xi_t$
Output: Next best view configuration $\xi_{t+1}$
Iteration:
Initialize a new tree $\mathcal{T}$ with $\xi_0 = \xi_t$
if first call then
    $g^* \leftarrow 0$
else
    addBranch($\mathcal{T}$, $\mathcal{B}_{t-1}$)
    $g^* \leftarrow$ updateBranchGain($\mathcal{B}_{t-1}$)
    $\xi^* \leftarrow$ getPreviousSolution()
end
while $time < time_{max}$ do
    $\xi_s \leftarrow$ sampleNewConfiguration()
    $\xi_n \leftarrow$ getNearest($\mathcal{T}$, $\xi_s$ )
    $\xi_{new} \leftarrow$ randomPropagation($\xi_s$, $\xi_n$ )
    if (not isStateInArea($\xi_{new}$) or not isMotionValid($\xi_{new}$, $\xi_n$)) then
        continue
    end
    addNode( T, $\xi_{new}$)
    callVisibilityService($\xi_{new}$)
    $\mathrm{V}_{parents} \leftarrow$ getNeighbors($\mathcal{T}$,$\xi_{new}$)
    for $\xi_p : \mathcal{V}_{parents}$ do
        if (not isMotionValid($\xi_{new}$, $\xi_p$)) then
            continue
        end
        $g \leftarrow$ computeBranchGain($\xi_p$ , $\xi_{new}$)
        if isGainBetterThan($g$, $g^*$) then
            setParent($\xi_{new}$, $\xi_p$ )
            $g^* \leftarrow$g
            $\xi^* \leftarrow \xi_{new}$
        end
    end
end
Save best branch $\mathcal{B}_t$
$\xi_{t+1} \leftarrow$ getNBV($\mathcal{B}_t$ )
Delete $\mathcal{T}$
Return: $\xi_{t+1}$
```

5.3 Influence of the VI formulations

A theoretical analysis of the proposed RRT-based receding-horizon coverage solution highlights essential properties. Generally speaking, as in [65] and [5], each time a new viewpoint is requested, the High-level planner tries to solve an optimization problem. Since the environment is unknown, the best approach is to compute the next moves according to the available data, and then, when new measurements have been acquired, repeat the process. Considering the presented RRT-based solution, the receding-horizon problem can be formulated as:

$$\begin{array}{rl} \max_{\mathcal{B}} & \mathcal{G}_{\mathcal{B}} \\ \text{s.t.} & \mathcal{B} = \{\xi_i\}_{i=0}^{N_p} \\ & \xi_i = f(\xi_{i-1}) \\ & \xi \in \mathcal{W}\ , \end{array} \tag{5.3}$$

where $\mathcal{B}$ is a branch that is composed of a sequence of random length N_p of configurations (viewpoints) $\{\xi_i\}$, and ξ_0 denotes the AUV actual configuration. The function f is the relation between two consecutive vehicle configurations ξ_i and ξ_{i-1}, that, in the context of this research, was represented by the Dubin kinematic constraints.

The problem cannot be solved using deterministic optimization algorithms. In fact, a deterministic relation between the gain function (BIG) and the workspace $\mathcal{W}$, where the configurations $\{\xi_i\}$ are sampled, i.e., the inspected area, is not available (the environment is not known a priori). De facto, the only way to know the BIG associated with a configuration is to test the configuration, i.e., compute the visible voxels given the available map; thus, the gain shall be considered a "black-box" function. Stochastic optimization strategies were developed to handle such problems [114]. Random search algorithms use a probabilistic approach by repeatedly sampling the feasible region, typically according to a uniform sampling distribution. They are proven to converge probabilistically to the global optimum with probability one, but the expected number of cost function evaluations grows exponentially with the feasible space dimension [115]. Thus, they were applied to many "black-box" global optimization problems to rapidly find a sub-optimal solution. Exploiting random trees to solve the optimization problem in Eq.5.3, as in this research or in [65], correspond to use a Random Search (RS) strategy [116]: RRT algorithms compute random trees by randomly sampling new configurations typically with a uniform distribution over the workspace, which leads to a Voronoi-biased expansion strategy to explore the workspace efficiently. Hence, the theory of the RS algorithm can be used to analyze the coverage/exploration problem. Firstly, as explained above, the gain function is a "black-box" whose value cannot be foretold. However, it is worth to note that it depends on two factors: the number of visible voxels and the IG formulation used. Regarding the former, the number of visible voxels along a branch $\mathcal{B}$ depends on the configurations that constitute the branch, which are randomly sampled, and on the sensor characteristics (FoV and range). The IG depends on the occupancy probability of visible voxels and the exploited VI formulation, such as the volume and entropy-based proposed in Section 3.4.2. Therefore, given a branch $\mathcal{B}$, i.e., a sequence of configurations (viewpoints), and the sensor used to accomplish the task, the set of visible voxels is defined, and the gain function of the optimization problem depends only on the VI formulation.

Since the high-level planner has a limited computing time to solve the optimization problem, according to the RS theory, it looks for a sub-optimal solution. Defined the optimum branch $\mathcal{B}^\star$ at each call of the algorithm and its gain $g^\star$ as:

$$\begin{aligned} \mathcal{B}^\star &= \arg\max_{\mathcal{B}} \mathcal{G}_{\mathcal{B}} \\ \mathcal{G}^\star &= \mathcal{G}_{\mathcal{B}^\star} \ , \end{aligned} \tag{5.4}$$

the algorithm tries to compute a sub-optimal solution with a gain $\mathcal{G}^\star - \epsilon$. According to [116], the probability of a RS algorithm to generate a sample with gain better than a value $c \in R$ can be defined:

$$p(c) = \nu(S(c))/\nu(\mathcal{W}) \ , \tag{5.5}$$

where $\nu(\cdot)$ denotes the Lebesgue measure, $\mathcal{W}$ is the workspace (where the samples are generated), and $S(c)$ is the level set:

$$S(c) = \{\mathcal{B} = \{\xi\} : \xi \in \mathcal{W} \text{ and } \mathcal{G}_{\mathcal{B}} \geq c\} \ . \tag{5.6}$$

Therefore, at each call, there is a probability

$$p(\mathcal{G}^\star - \epsilon) = \nu(S(\mathcal{G}^\star - \epsilon))/\nu(\mathcal{W}) \ , \tag{5.7}$$

of finding a sub-optimal solution. Finally, it is possible to analyze the effect of the VI on the high-level planner. Fig.5.4 shows the VI formulations (Section

3.4.2) normalized with respect to the their maximum value. As discussed above, given a set of visible voxels, the gain function of the optimization problem in Eq.5.3 depends only on the voxels' occupancy probability and the VI used. Thus, as shown in Fig.5.4, according to Eq.5.7, since the entropy VI formulation has a larger sub-optimal level set, an algorithm using such VI formulation is more likely to find a sub-optimal solution than by exploiting the volume-based.

From the presented analysis, some important considerations can be drawn. The exploited VI formulation impacts the performance of the here proposed solution. A receding-horizon strategy is used in this work; thus, the last call's best solution is kept to initialize the algorithm. However, this solution was the best according to the information available in the previous step. The VI based on the entropy enhances the algorithm probability of finding a sub-optimal solution within the given computation time, meaning that the algorithm is more likely to compute a new best solution. In conclusion, the entropy VI formulation leads to a more well-posed optimization problem and could enhance the performance of the coverage algorithm. This analysis is validated with the results presented in Section 5.4 and Section 5.5, where a comparison of the two VI formulations is proposed.

5.4 Validation and quantitative analysis

The proposed RHCA has firstly been validated with an extensive comparison of realistic simulations made by means of the UUV Simulator. FeelHippo AUV, detailed in Section 1.2, was selected as the testing platform. Since conducting experimental sea trials is time and cost expansive, this validation step made through realistic simulations, has been of utmost importance. In fact, the results reported in this Section aimed to validate and tune the developed coverage framework and the proposed analysis in light of the sea trials described in the next Section 5.5. In fact, during the experimental campaign, the coverage solution was tested with the parameters found in the here reported validation process.

The realistic simulations were based on the dynamic model of FeelHippo AUV that was implemented in the UUV Simulator. Coverage surveys at a constant altitude of 2 meters from the seabed to inspect an area of 36x34 m were simulated. To make these simulations as realistic as possible, the BlueView M900 2D FLS, which is mounted on FeelHippo AUV, was simulated

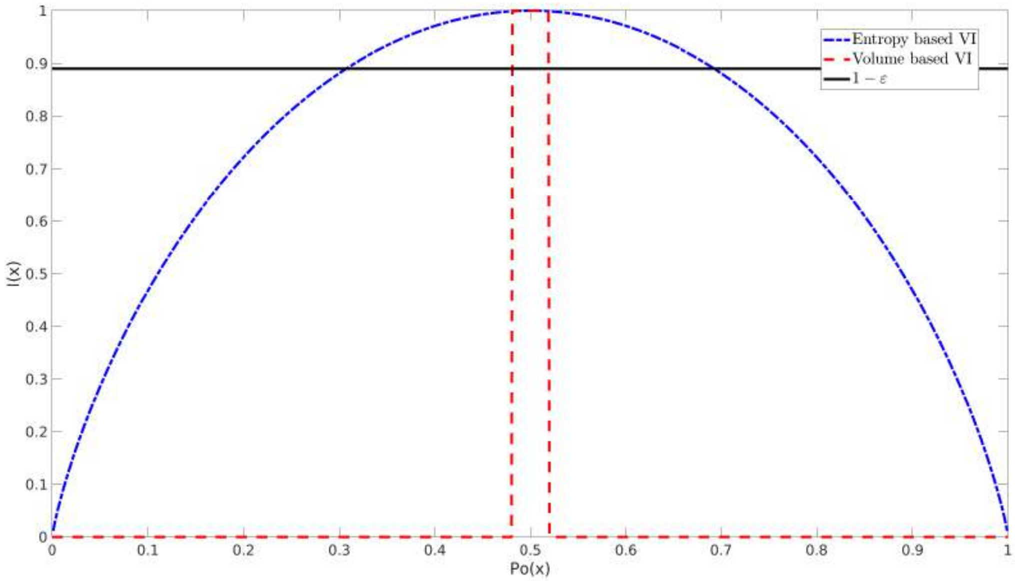


Figure 5.4: Comparison of the VI formulations normalized with respect to their maximum value. Given a sub-optimal value $(1-\varepsilon)$, depicted with the continuous black line, the entropy-based VI formulation (blue dashed-dotted line) has a larger sub-optimal level set than the volume-based formulation (red dashed line).

through the UUV Simulator. The sensor has a horizontal FoV (hFOV) of 130° and a vertical FoV (vFOV) of 20° and the range was set to 10 m. The FLS was mounted in front of the vehicle with a tilt angle of 30° w.r.t. the horizontal plane. The Motion planner and High-level planner modules used the Dubins curves with a turning radius of 3 m to model the AUV kinematics constraints. Finally, the High-level planner maximum computing time was set to 2 s.

Firstly, two lawnmower paths were designed, considering the characteristics of the sensor (FoV and range) and the target area dimensions. In particular, the lawnmower patterns were designed by two different operators to ensure the coverage of the 90% of the inspected area with the AUV flying at a constant altitude of 2 meters from the seabed, equipped with the considered FLS. As shown in Table 5.1, the length of the paths varies. The coverage solution was validated by performing 10 simulations with and without the rewiring strategy for both the proposed VI formulations (a total of 40 runs were performed). For each simulation, the AUV position and heading were randomly initialized. Since the developed coverage algorithm is based on a probabilistic approach, Table 5.1 reports the path length mean and standard deviation of the performed experiments. It is worth highlighting that the developed RHCA does not make use of the inspection area shape; thus, to the best of the author's knowledge, the results reported here have general validity. *RHCA volume* denotes the proposed receding-horizon coverage approach with the volume-based VI, while *RHCA entropy* stands for the proposed solution using the VI formulation based on the entropy function. When the rewiring strategy is used, the algorithm tests the closest $k = 10$ neighbor nodes. The rewiring procedure is computationally expensive; this parameter was selected heuristically as a trade-off to let the algorithm evaluate a sufficient number of possible parents and generate an adequate number of samples during the given planning time.

Table 5.1: Outcomes of the coverage simulations

	Coverage level 80%		***Coverage level*** 90%	
Method	***Path length mean [m]***	***Path length standard deviation [m]***	***Path length mean [m]***	***Path length standard deviation [m]***
Lawnmower 1	102.94	-	121.31	-
Lawnmower 2	87.76	-	106.47	-
RHCA volume	98.87	9.47	126.65	16.01
RHCA entropy	89.63	9.75	123.16	13.14
RHCA volume - rewiring	95.59	7.35	124.17	13.09
RHCA entropy - rewiring	96.13	4.44	123.21	12.35

Table 5.1 highlights the effectiveness of the here proposed coverage solution and points out some key aspects. The proposed RHCA solution is based on a random optimization process that varies the performance in each trial. As shown by the path length mean and standard deviation, the RHCA outperformed the lawnmower paths in some trials, but the planned path is longer in other experiments. However, lawnmower paths are usually developed by skilled operators, taking into account both the environmental characteristics and the robots' payloads; the here proposed strategy could represent a solution for science-users that do not require such field experts. Besides, the classic surveys may require multiple attempts. In fact, the AUV acquires data passively, without considering the quality and quantity of the gathered data. As these simulations are concerned, they were conducted in a smooth sea

bottom scenario without significant shadowing phenomena. By knowing the geometric properties of the scene and the sensor operational settings in advance, the lawnmower patterns were easily designed. The perfect knowledge of the environment allowed the lawnmower surveys to achieve the requested coverage levels (without requiring multiple attempts) and outperform the proposed RHCA. However, such conditions do not hold when exploring and gathering data of new areas is demanded. In the rapidly changing underwater environment, AUVs do not always deal with smooth bottom scenarios, and no knowledge is available a priori to operators to design the missions. The proposed strategy instead ensures an adequate coverage level of the inspection area by actively monitoring the gathering process. It permits to obtain reasonable performance, close to the lawnmower missions, without knowing the underwater scenario in advance. Moreover, the RHCA eliminates the pre-mission time: the AUV is deployed, and the mission can start avoiding an operator to design a suitable path that might take several minutes.

Deepening the analysis, the outcomes validate the theoretical investigation proposed in Section 5.3. The entropy-based VI formulation led to shorter paths for achieving both the 80% and the 90% of the coverage. Regarding the former goal, the *RHCA entropy* reached a mean path length of 89.63 m, which is close to the inspection conducted following the Lawnmower 2; while the *RHCA volume*, which accomplished a mean path length of 98.87 m, outperformed the survey conducted with the Lawnmower 2. The importance of the theoretical analysis of Section 5.3 emerges from comparing the RHCA results with the two VI formulations. The entropy VI formulation enhances the coverage algorithm's success probabilities of computing a solution better than the one found at the previous call. Hence, it is more likely to update the survey path than the volume-based VI. Consequently, it improves the performance of the proposed methodology.

Achieving a 90% of coverage is a more challenging task. The more the AUV covers the area, the more difficult the optimization problem of Eq.5.3 becomes. It means that the algorithm is less likely to change the computed old best path, leading to worse performance, i.e., longer surveys. A longer planning time could lead to better performance of the RHCA, and better highlight the effect of VI formulation. Nevertheless, the planning time is a trade-off between the proposed framework's performance and the online computation constraints.

The rewiring procedure reduces the variances of the computed paths. In fact, by testing different possible parents of the newly generated node, the optimization is guided toward better solutions. However, this mechanism is computationally expensive, reducing the number of generated samples within the given time. While it enhances the RHCA volume performance, it deteriorates the RHCA entropy outcomes. By generating fewer samples, it reduces the probabilities of computing new good paths, but it enables the algorithm to perform better updates, i.e., the most promising branches are expanded. Although the rewiring strategy deteriorates the mean values of the RHCA entropy, it reduces the standard deviation.

5.5 Experimental results

The proposed framework was tested in real sea trials in shallow waters, performed in October 2020 with FeelHippo AUV at CSSN basin in La Spezia, Italy. The experimental campaign aimed to validate the coverage framework in an unknown real environment. Besides, the results validated the theoretical and quantitative analyses discussed above. An inspection area as large as the one used during the simulations (Section 5.4), i.e., 36x34 m, was selected. FeelHippo AUV performed several surveys at constant altitude (2 meters) to inspect the seabed using the BlueView M900 2D FLS. The FLS was mounted in front of the vehicle with a tilt angle of 30° w.r.t. the horizontal plane (see Fig.5.5). The SONAR range was set to 10 m. The Mapping module aimed to create an occupancy grid map of the covered area that is used as an active feedback for the High-level planning module. Thus, since the goal was not to create a detailed reconstruction of the environment, the map resolution was set to 0.5m. Both the Motion planner and High-level planner modules modeled again the AUV kinematics constraints with Dubins curves with a turning radius of 3 m. The High-level planner had 2 s to compute the NBV.

First of all, in order to have a benchmark for evaluating the proposed methodology, a lawnmower survey at a constant altitude (2 meters) over the selected area of interest was performed. The executed lawnmower pattern was designed considering the SONAR characteristics and its mounting pose with respect to the AUV. By following such a path, the AUV managed to cover 80% and 90% of the target area in 65.34 m and 72.37 m, respectively.

Regarding the proposed RHCA, the experiments showed that the developed approach led the AUV toward the inspection surveys. Fig.5.6 reports a sequence of snapshots of the planning process. The High-level planner used the algorithm described in Section 5.2 to grow a random tree from the AUV initial position. The best branch (whose nodes are reported in yellow in Fig.5.6), i.e., the branch that is expected to acquire more information, was selected, and the first node became the NBV. Then, FeelHippo AUV followed the path computed by the Motion planner (depicted in green). When the NBV was reached, the High-level planner grew a new random tree that was initialized using the previous best solution. As reported by Fig.5.6, the RHCA solution has the desired behavior: while performing the survey, the AUV can replan the mission considering the Mapping module feedback.

To highlight this sought behavior, Fig.5.7 reports a visualization of the data collected during an inspection survey planned by the developed RHCA during the experimental campaign. While the AUV was following the inspection path, the Mapping module updated the occupancy map and allowed to monitor the covered area. The map was updated using the mapping strategy described in [9] that made use of the gathered FLS images. The map's use is twofold: it is used to monitor the progress of the coverage survey and plan the NBV. In fact, as shown in Fig.5.7, the RHCA considers the map to move the AUV toward non-enlightened regions. The inspection path planned by the proposed High-level planner is depicted with the blue line, while the purple spheres represent the path to lead the AUV to the NBV, calculated by the Motion planner. In particular, it is worth noting that since the initially planned survey (Fig.5.7a) managed to guide the vehicle through non-enlightened regions of the area of interest, it was not modified. Each time the AUV reached a viewpoint (the blue spheres) of the inspection path, the view planning algorithm was executed, but within the given computational time, it did not find a better path (Fig.5.7b and Fig.5.7c). Then, as depicted in Fig.5.7d, the inspection path was updated to complete the coverage of the area. Finally, Fig.5.8 shows the map created by means of the developed RHCA when the 90% of the inspection area was covered. Therefore, thanks to the developed solution, the vehicle inspected the seabed actively by monitoring the quantity and quality of the acquired data during the survey and using the data to replan the mission. Then, an adequate coverage level can be ensured, avoiding multiple attempts.

Turning to quantitative analysis, the RHCA was tested at field both with the volume and the entropy VI formulations. To assess the developed RHCA solution's performance in a real scenario, four trials for each VI formulation were performed: two with and two without the rewiring procedure. Table 5.2 reports the obtained results using the volume-based VI. The vehicle managed to inspect the area with satisfying results. In accordance with the valida-

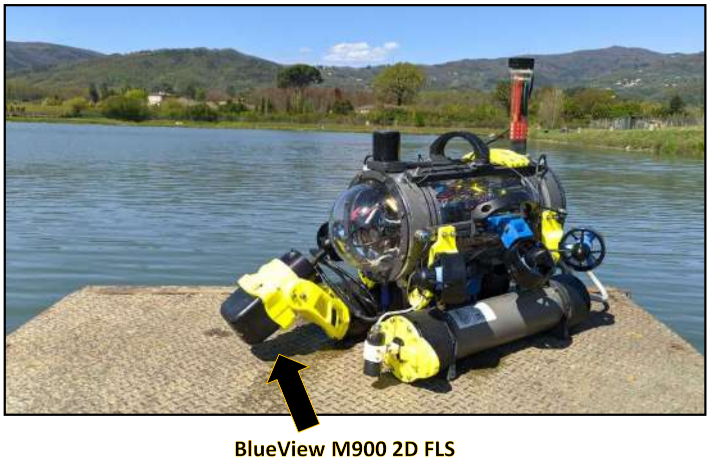


Figure 5.5: FeelHippo AUV endowed with the BlueView M900 2D FLS used to conduct the inspection surveys during the experimental campaign.

Table 5.2: RHCA volume-based VI sea trials results

Method		***Coverage level*** 80%	***Coverage level*** 90%
RHCA volume	***rewiring***	***Path length [m]***	***Path length [m]***
Trial 1	no	88.87	118.91
Trial 2	no	91.34	114.74
Trial 3	yes	87.82	104.53
Trial 4	yes	83.79	110.15

tion experiments' outcomes, the rewiring procedure reduced the path length of the *RHCA volume* method. In fact, by testing possible parents of new nodes, the rewiring procedure helps the algorithm finding most promising solutions. In Table 5.3 the results of the trials performed using the *RHCA entropy* algorithm are reported. Together with the realistic simulations, these experimental results proved the validity of the theoretical analysis of the developed coverage methodology. The entropy VI formulation leads to a more well-posed optimization problem, and by enhancing the algorithm probability of finding a new best solution at each call, it improves the performance

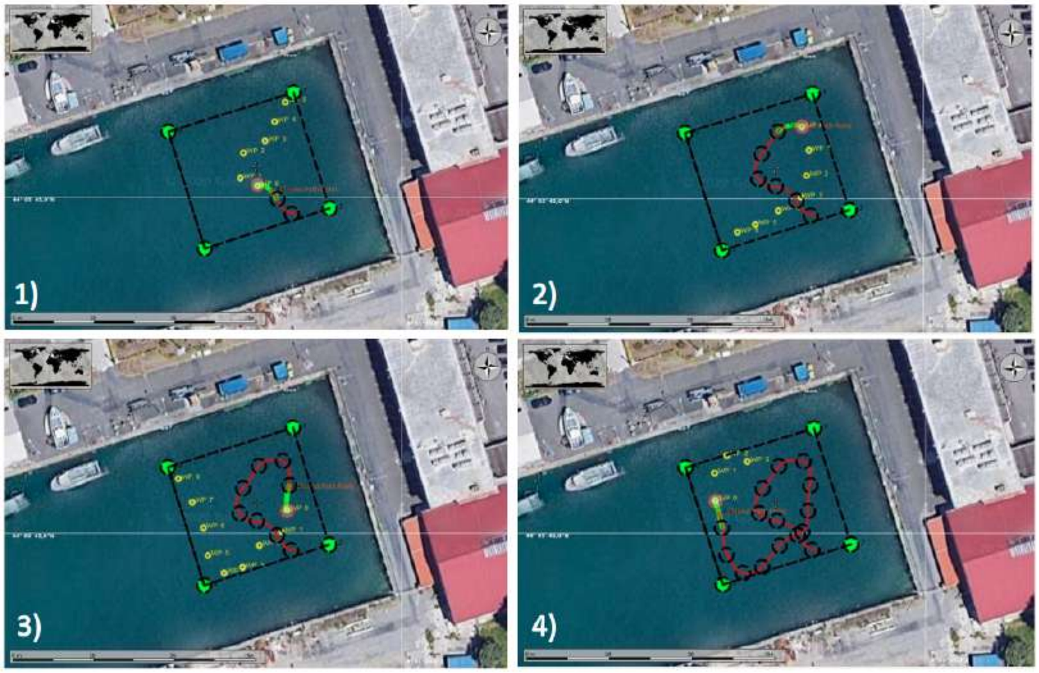


Figure 5.6: Different snapshots of one of the autonomous surveys conducted by FeelHippo AUV during the experimental campaign, La Spezia (Italy). Using the developed coverage framework, the vehicle managed to inspect the seabed of the target area, defined by the four green points and the four black dashed lines. The yellow points are the nodes of the best branch computed by the High-level planner. The path to lead the AUV to the NBV is depicted in green, while the estimated AUV tracked path is reported in red.

Table 5.3: RHCA entropy-based VI sea trials results

Method		***Coverage level*** 80%	***Coverage level*** 90%
RHCA entropy	***rewiring***	***Path length*** *[m]*	***Path length*** *[m]*
Trial 1	no	75.39	95.85
Trial 2	no	76.92	88.87
Trial 3	yes	86.41	105.74
Trial 4	yes	80.66	121.61

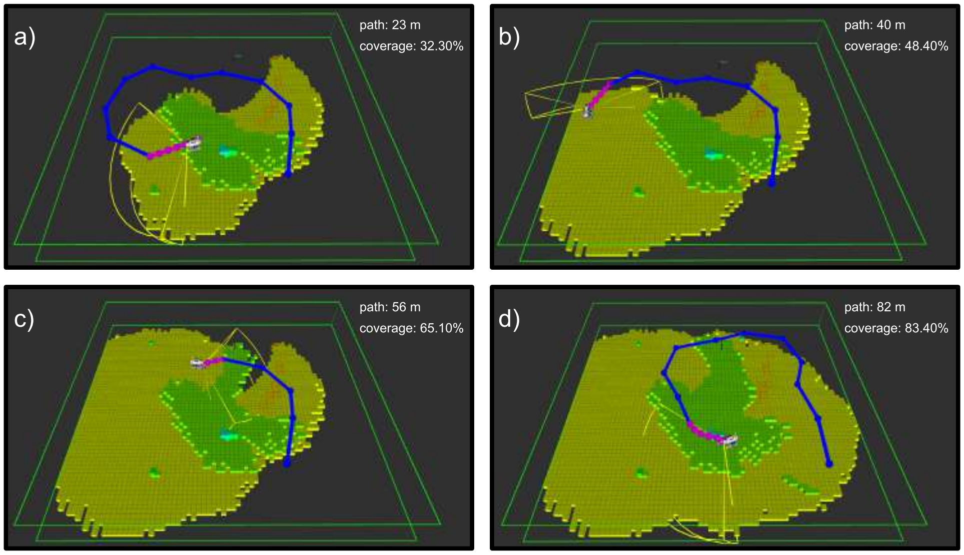


Figure 5.7: Four snapshots of an inspection survey performed by FeelHippo AUV exploiting the developed RHCA. Two green polygons delimit the inspection area at depth 0 m and the maximum depth. The Mapping module uses the data gathered with the FLS, whose FoV is represented with the yellow lines, during the survey to update the map of the covered area. The inspection path generated by the High-level planner is shown with the blue line, while the purple spheres depict the path created by the Motion planner, which the AUV tracks to reach the computed NBV. For the sake of completeness, the snapshots report, on top on the right side, the traveled distance and the reached coverage level.

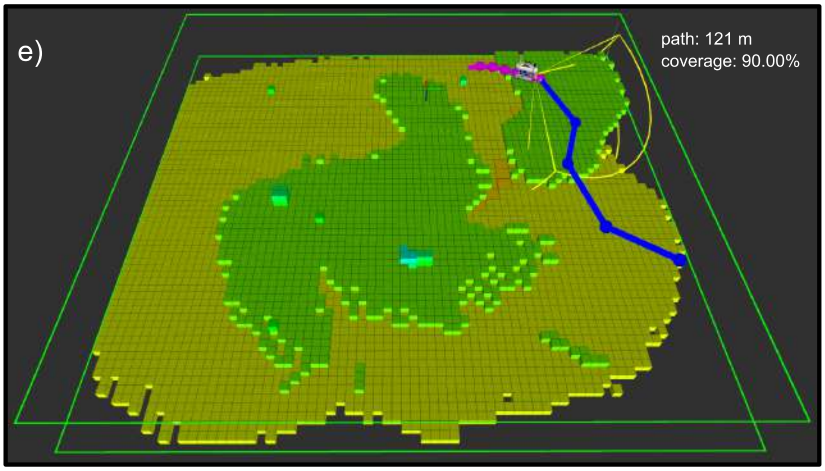


Figure 5.8: The final map of the inspected area. FeelHippo AUV conducted the inspection survey autonomously by using the developed coverage solution.

of the inspection framework. As shown in Section 5.4, the rewiring strategy deteriorates the *RHCA entropy* algorithm's performance. It is a computationally expensive process that reduces the number of generated samples. Consequently, it reduces the chances of computing new sub-optimal solutions.

In conclusion, the RHCA strategy using the entropy-based VI formulation without the rewiring procedure led to better results in both simulations and real sea trials. The outcomes are in line with the theoretical discussion (Section 5.2). Besides, the experimental campaign demonstrated that the developed framework can guide the AUV toward active inspection surveys in a real unknown environment and can guarantee adequate levels of coverage of the target area.

Finally, to evaluate the developed framework, the achieved results shall be compared with the ad-hoc pre-planned lawnmower pattern. By using both the VI formulations, the proposed methodology led to longer coverage paths than the state-of-the-art lawnmower. Nevertheless, the lawnmower path requires the perfect knowledge of the scenario and the FLS operational settings in advance, and during the mission, the AUV passively stores the acquired data. Moreover, it was calculated by a skilled operator and required a non-negligible pre-mission time. On the other hand, the hereby sensor-driven RHCA, being fully probabilistic, permits to obtain reasonable performance (close to the lawnmower mission) without knowing the underwater scenario in advance. In addition, it eliminates the path designing time, and by monitoring the gathered data through the created map, ensures the insonification of a required portion of the seafloor. Therefore, the developed RHCA could represent a solution for AUV end-users by simplifying the data acquisition process.

5.6 Main contribution

A framework to enable an AUV to perform FLS seabed inspections autonomously, but suitable for any acoustic or optical sensor, was presented in this Chapter.

The framework is composed of a Mapping module, a High-level planner, and a Motion planner. In detail, the latter generates the Dubins path that connects the AUV position to the target NBV, while the Mapping module exploits the probabilistic 3D occupancy mapping system for FLS based reconstructions proposed in [9]. The focus of this work has shifted to the High-level for which a sensor-driven coverage path planning algorithm was developed. The algorithm utilizes the Mapping module updated map to expand random trees and find the NBV. The visibility of each node of the tree is evaluated on the map through a ray casting process that takes into account the FLS range and FoV. The AUV tracks the path computed by the Motion planner to reach the NBV; then, the process is repeated in a receding-horizon paradigm: the previous best branch is evaluated on the updated map, and it is used to initialize the new tree. The usage of such a RRT-based methodology, inspired by the aerial domain, has never been investigated, to the author's knowledge, in the underwater domain.

Moreover, several metrics for estimating the discovered map from a viewpoint have been proposed in the last years, but their influence on the considered problem has only been evaluated heuristically. Here, a mathematical formulation of the developed coverage algorithm and a theoretical investigation of the effects of the exploited VI formulation were provided. Besides, a comparison of two VI formulations was reported. To the author's best knowledge, the theoretical analysis of the VI formulations is novel. This research work also presents the first investigation of information gain metrics for seabed inspections.

The proposed framework was also validated with realistic simulations and then through a dedicated experimental campaign at sea. The developed solution endowed FeelHippo AUV with the ability to autonomously inspecting the seabed in a target area. By actively monitoring the data gathering process, it ensures adequate coverage levels and avoids multiple attempts. In addition, the outcomes are in accordance with the theoretical analysis; the entropy-based VI enhances the algorithm chances of computing a better solution w.r.t. the previous call and leads to better results, i.e., shorter paths.

Chapter 6

Advances in RRT-based view planning for autonomous inspections

Chapter 5 presented the designed RHCA for conducting autonomous inspections and detailed the view planning algorithm. In this Chapter, advances for enhancing the performance of the developed algorithm are proposed.

Firstly, due to the time constraints, the algorithm detailed in Chapter 5 managed to test only ten parent candidates for each new viewpoint and did not evaluate the effect of new nodes in the tree. In fact, performing a complete rewiring procedure would have reduced the number of generated viewpoints within the given computational time. Thus, to overcome this limitation, the RRT-based planning algorithm has been rewritten in a parallel fashion. This new implementation allows testing possible rewiring actions while a new node is generated.

In addition, an innovative sampling methodology for guiding the expansion of the tree towards unexplored areas is presented. This novel informed tree expansion strategy that combines the KDE technique and a rejection sampling method represents the major contribution of this Chapter. The performance of the RHCA using this novel strategy based on estimating the probability density function of the discovered map by using the KDE technique is compared with the state-of-the-art RRT sampling policy based on a uniform distribution.

6.1 Parallel implementation of the view planning algorithm

The performance of the developed RHCA solution strictly depends on the view planning algorithm. As described in Section 5.4 and Section 5.5, during the conducted campaigns, only ten parent candidates were considered during the rewiring routine. This parameter was selected as a trade-off to allow Algorithm 1 to test a sufficient number of possible parents and generate an adequate number of configurations within a given computational time. Evaluating more parent candidates could enhance the performance achieved by the RHCA framework, but it would reduce the number of generated viewpoints. Besides, the algorithm presented in Chapter 5 looked for only the best parent and did not evaluate whether the new node improves the expected gain of other branches. Thus, the view planning algorithm has been rewritten in a parallel manner to perform a complete rewiring routine without reducing the generated viewpoints.

First of all, by analyzing Algorithm 1, two distinct phases can be identified: a generation step, where a new node is created and added to the tree (lines 8-13), and then a rewiring step, where possible connections are tested (lines 14-21). In detail, in the former step, a configuration is sampled using a uniform distribution over the workspace, and a new node is computed by expanding the closest tree node toward the direction of the sampled configuration. It is worth noting that by using this strategy, tree nodes that correspond to larger Voronoi regions are more likely to be selected for the expansion, i.e., it creates a Voronoi-biased expansion that guarantees that the tree explores the workspace rapidly. Once the pose of the viewpoint has been calculated, its visibility can be evaluated, which in the context of this work, it was done by the Mapping module (see Fig. 5.1) that acted as a server for the High-level planner. As stated in Section 5.2, it is worth highlighting that the visibility of a node depends only on its position and orientation. On the other hand, in the rewiring step, the best parent of the new viewpoint is searched by calculating the expected BIG assuming near tree nodes as parents.

In Algorithm 1, the generation and the rewiring phases are sequentially performed. Deepening the analysis, it emerges that for performing a step of the algorithm, the former takes about 75% of time, while the latter the remainder 25%. In particular, during the generation phase, most of the time is spent waiting for the Mapping module computing the visibility. Therefore, by the time the generation step is computed, the rewiring procedure could be performed. However, the standard RRT implementation on which Algorithm 1 is based does not allow such operation. In fact, until the node visibility is not calculated, it is not possible to test other tree nodes as a parent. As a consequence, a novel implementation of the RRT-based algorithm, Parallel Rapidly-exploring Random Tree (pRRT) in the following, has been designed. As detailed in Algorithm 2, it allows executing the view planning algorithm in a parallel fashion. To this end, at each step, Algorithm 2 generates a new node ξ_n and calls the service for evaluating its visibility (Algorithm 3), while it runs the rewiring routine (Algorithm 4) for finding the best connections for the node created at previous iteration ξ_p. As stated above, this simple structure allows the use of a parallel implementation. In fact, the functions **generateNode()** and **rewireNode(ξ)**, in lines 33 and 34 respectively, can be run concurrently by using two distinct threads. Therefore, by the time the generation phase is performed, the rewiring routine can be executed without increasing the time required for a step of the algorithm. That is, pRRT only leverages the advantages of the rewiring process.

Thanks to the developed pRRT, the rewiring procedure, described in Algorithm 4, can both look for the best parent of the new node (lines 52-58) and checks whether the new node enhances the tree by evaluating if it can improve the expected gain of other viewpoints (lines 59-66).

6.2 Informed expansion of the tree

In the proposed RHCA solution, the RRT algorithm was used to find the NBV. As a matter of fact, the RRT can rapidly explore the workspace by exploiting a Voronoi-biased expansion strategy. That is, at each iteration, by sampling a random configuration with a uniform distribution (line 42 of Algorithm 3) and expanding the closest node in the tree (lines 43-44), nodes that correspond to larger Voronoi regions are more likely to be selected. As shown in Figure 6.1, published in the remarkable work presented in [117], such a strategy ensures that the generated tree well explores the workspace, better than a naive solution where nodes are randomly selected for the expansion. Therefore, the sampling policy of the expansion phase plays a key role.

For what concerns the motion planning task, several efforts have been made to develop innovative strategies for sampling new configurations to en-

Algorithm 2 Coverage planner algorithm parallel implementation

```
Input: AUV configuration $\xi_t$
Output: Next best view configuration $\xi_{t+1}$
Iteration:
Initialize a new tree $\mathcal{T}$ with $\xi_0 = \xi_t$
Initialize $\xi_p = Null$
if first call then
    $g^* \leftarrow 0$;
else
    addBranch($\mathcal{T}$, $\mathcal{B}_{t-1}$)
    $g^* \leftarrow$ updateBranchGain($\mathcal{B}_{t-1}$)
    $\xi^* \leftarrow$ getPreviousSolution()
end
while $time < time_{max}$ do
    if $\xi_p = Null$ then
        $\xi_n \leftarrow$ generateNode()
    else
        $\xi_n \leftarrow$ generateNode()
        rewireNode($\xi_p$)
    end
    if nodeIsValid($\xi_p$) then
        addNode( T, $\xi_p$)
    end
    $\xi_p \leftarrow \xi_n$
end
$\mathcal{B}_t \leftarrow$ getBestBranch($\mathcal{B}_t$ )
Save best branch $\mathcal{B}_t$
$\xi_{t+1} \leftarrow$ getNBV($\mathcal{B}_t$ )
Delete $\mathcal{T}$
Return: $\xi_{t+1}$
```

hance the performance of RRT algorithms. For instance, a common solution for steering the tree toward the solution is to randomly sample the goal configuration. Thus, a trade-off between the exploration of the workspace and the rapidity of obtaining a solution to the task can be achieved. Another approach, among the others, was presented in [118], where the subset of states that can improve a solution was defined and used to sample the new

Algorithm 3 generateNode()

```
Input:
Output: New configuration $\xi_{new}$
Iteration:
$\xi_s \leftarrow$ sampleNewConfiguration()
$\xi_n \leftarrow$ getNearest($\mathcal{T}$, $\xi_s$ )
$\xi_{new} \leftarrow$ randomPropagation($\xi_s$, $\xi_n$ )
if ( isStateInArea($\xi_{new}$) and isMotionValid($\xi_{new}$, $\xi_n$)) then
    callVisibilityService($\xi_{new}$)
    setNodeValidity($\xi_{new}$, True)
    Return $\xi_{new}$
else
    generateNode()
end
```

configurations.

When it comes to view planning applications, the RRT exploration properties are of utmost importance since evaluating the visibility of a viewpoint through the ray casting routine is time expensive, and thus a limited number of configurations can be generated within a given planning time. As a consequence, for finding good inspection paths, the view tree has to contain well-spawned nodes. However, while the Voronoi-based expansion is an efficient solution at the beginning of the survey (i.e., when most of the area is unknown), it loses its effectiveness as the area is explored. That is, to generate a branch that leads the robot toward unexplored regions, more viewpoints shall be generated since the optimization becomes more challenging.

In this Section, a novel methodology for biasing the tree expansion to unexplored areas is introduced. By making use of the KDE technique, the discovered occupancy map density function can be estimated and used to sample new configurations. Therefore, while Algorithm 1 can find a path by chance (as in [64] and [65]), with the hereby proposed sampling methodology a map-aware expansion is achieved. Simulation and experimental results demonstrate that the proposed strategy enhances the performance of the RHCA.

```
Algorithm 4 rewireNode($\xi$)
Input: $\xi$
Output:
Iteration:
$g_b \leftarrow$ getGain($\xi$)
$V_p \leftarrow$ getNeighbors($\mathcal{T}$,$\xi$)
for $\xi_p : \mathcal{V}_p$ do
    if (not isMotionValid( $\xi_p$, $\xi$)) then
        continue
    end
    $g \leftarrow$ computeBranchGain($\xi_p$ , $\xi$)
    if isGainBetterThan($g$, $g_b$) then
        setParent($\xi$, $\xi_p$ )
        setGain($\xi$, $g$ )
        $g_b \leftarrow$g
    end
end
for $\xi_c : \mathcal{V}_p$ do
    if (not isMotionValid($\xi$, $\xi_c$)) then
        continue
    end
    $g_c \leftarrow$ getGain($\xi_c$)
    $g \leftarrow$ computeBranchGain($\xi$, $\xi_c$)
    if isGainBetterThan($g$, $g_c$) then
        setParent($\xi_c$, $\xi$ )
        setGain($\xi_c$, $g$ )
        updateChildrenGain($\xi_c$ )
    end
end
Return
```

6.2.1 KDE for estimating the discovered map distribution

The developed methodology for biasing the expansion of the tree makes use of KDE technique [119] that estimates the density function of the discovered map. The KDE is a non-parametric method to estimate the probability

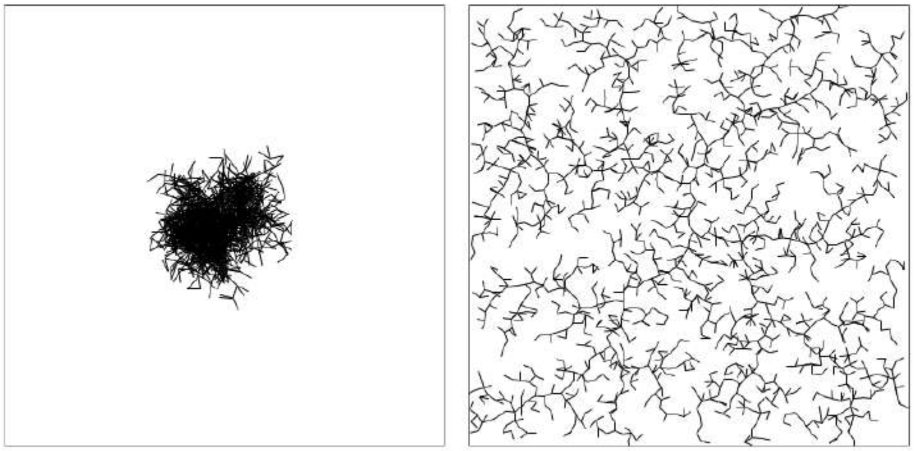


Figure 6.1: Comparison of the expansion of a Naive Random Tree (on the left side) with an RRT (right side), both composed of 2000 nodes [117].

density function of independent and identically distributed samples that has found a broad use in statistics and ML.

In particular, let $(s_1, s_2, \ldots, s_n)$ be independent and identically distributed samples drawn from some distribution with an unknown density function f at any given point s. The KDE estimator allows to estimate the value of $f(s)$ as:

$$\widehat{f}_h(s) = \frac{1}{n}\sum_{i=1}^{n} K_h\left(s - s_i\right) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{s - s_i}{h}\right), \tag{6.1}$$

where K is the kernel, which in the context of this research a Gaussian Kernel was considered, and h is the bandwidth, a tuning parameter. Considering the occupancy mapping paradigm used in this research [9], by utilizing as the samples the instantiated map cells $\{m_i\}_{i=1}^{n}$, i.e. the discovered map, the value of density function of a point p can be estimated:

$$\widehat{f}_h(p) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{p - m_i}{h}\right). \tag{6.2}$$

Eq.6.2 can be used for various purposes. As this research work is concerned, it was employed to generate samples to expand the tree toward the unexplored regions, as detailed in the next Section. In particular, the

RHCA framework was developed with the goal of making an AUV able to autonomously inspect the seabed of an area of interest using an FLS. Therefore, the reconstructed seabed made through the solution presented in [9] was considered as the discovered map, and since constant altitude surveys were assumed, only the 2D position of the cells was utilized. It is worth highlighting that the method could be employed for exploration tasks and can be extended to 3D applications. In fact, by using such a solution, a robot learns the distribution of the discovered map and can use it to guide the exploration.

6.2.2 Expanding the tree toward non-explored regions

The novel methodology based on the KDE technique for estimating the distribution of the map provides the opportunity to develop an informed map-aware tree expansion strategy. As a consequence, the RHCA is more likely to find paths that steer the robot toward non-explored regions than by using the standard Voronoi bias.

Firstly, a sampling strategy shall be designed. Given a distribution, samples are typically drawn by using different techniques so that they are distributed accordingly. As this work is concerned, the goal is to generate samples that do not belong to the given distribution. In particular, since Eq.6.2 estimates the distribution of the discovered map, a sampling policy that aims at generating observations in the non-explored regions is required.

To this end, the estimated probability density function given by Eq.6.2 was employed for implementing a rejection sampling algorithm [120]. The reject method is a basic sampling approach that allows drawing random numbers from various distributions. It was used since Eq.6.2 only estimates $\widehat{f}_h(p)$, and an explicit form of f is not available, and thus, approaches such as the inverse transform sampling cannot be exploited.

Briefly, the designed sampling algorithm utilizes a simple workflow. It samples a point p with a uniform distribution over the workspace, and a value $u \in \mathcal{U}[0, \max\{\widehat{f}_h(m_i)\}_{i=1}^{n_m}]$, where $\{(m_i)\}_{i=1}^{n_m}$ denote the map discovered cells and $\mathcal{U}$ denotes the uniform distribution. Then, $\widehat{f}_h(p)$ is computed, and if $u > \widehat{f}_h(p)$, which means that the point p does not belong to the distribution f, i.e., it is in an unexplored region, the point p is accepted, otherwise it is rejected and process is repeated until a good point is found.

To validate the proposed methodology, known seabed reconstructions with a resolution of 0.5 m were simulated. Three examples are depicted on the left side of Fig. 6.2, whose estimated distributions are reported in the middle column. Regarding the utilized KDE estimator, the mlpack library was used [121]. In detail, a Gaussian kernel was employed, and the bandwidth was set to 0.5, corresponding to the map resolution. Finally, the histogram of 5k samples generated by using the above-described rejection method is reported on the right side of Fig.6.2.

As described in the following, the developed KDE-based sampling methodology was run online on the main computer of FeelHippo AUV and used during the conducted experimental campaign. Such a solution allows the robot to learn the distribution of the discovered map, and thus it learns where to sample the most promising configurations for guiding the view planning algorithm.

Therefore, it is possible to bias the expansion of the tree by utilizing the samples drawn in the non-explored areas during the generation step of Algorithm 2. To this end, the function **sampleNewConfiguration()** (line 42, Algorithm 3) has been modified as follows. At each call, the function samples a value $p_s \sim \mathcal{U}[0,1]$, where $\mathcal{U}$ denotes the uniform distribution. Then, if p_s is smaller than a pre-defined threshold t_s, ξ_s is generated using the KDE-based methodology so that it biases the tree toward the unknown region. Otherwise, it is randomly sampled with a uniform distribution over the workspace. It is worth highlighting that the threshold t_s allows modifying the behaviour of the expansion procedure. While small values promote the use of the standard uniform distribution, which leads to the Voronoi-bias and thus a rapid exploration of the workspace, with a threshold close to 1, the here proposed informed expansion is more likely to be used, and the tree is steered toward the most promising areas. By selecting intermediate values, a trade-off between the two behaviour can be achieved.

As reported in the next Sections, in this work three values of t_s were investigated: $t_s = 0$, i.e., the standard expansion routine, $t_s = 1$ for utilizing only samples generated with the proposed KDE methodology, and $t_s = 0.5$ that balances the Voronoi-bias and the informed expansion.

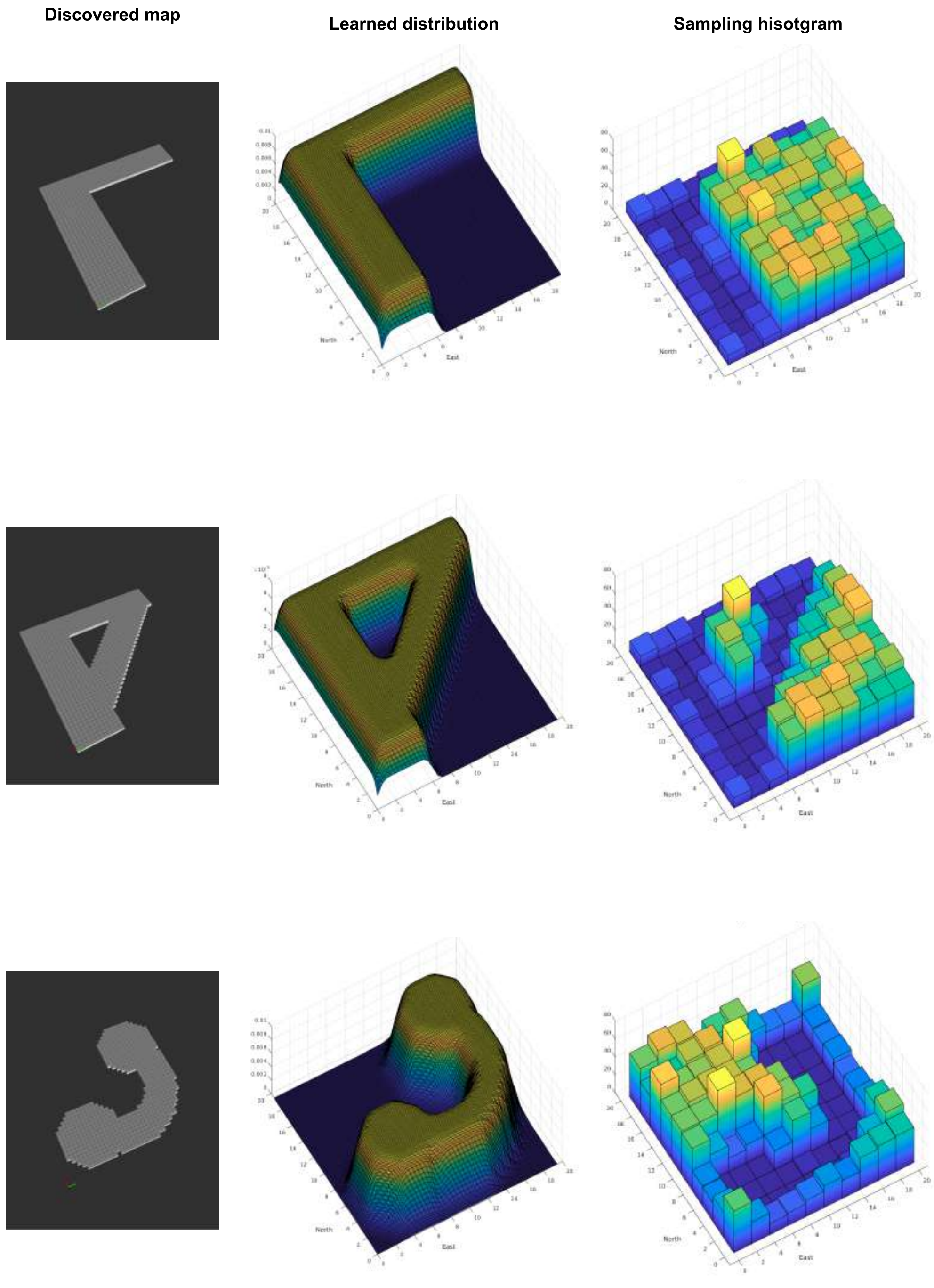


Figure 6.2: On the left side, the occupancy map of the simulated seabeds, whose distributions were estimated with the KDE approach (middle column). On the right side, the histogram of 5k samples drawn in the unexplored region.

6.3 Validation

To test and validate the advances proposed for the view planning algorithm utilized in the RHCA framework, realistic simulations were firstly performed exploiting the UUV Simulator. As in Chapter 5, a dynamic model of Feel-Hippo AUV was used, and the BlueView M900 2D FLS, which has an hFOV of 130° and a vFoV of 20°, was simulated. The FLS range was set to 10 m. To have results that are consistent with those previously obtained, the same conditions detailed in Chapter 5 (Section 5.4 and Section 5.5) were considered. In particular, FLS-based coverage surveys at a constant altitude (2 meters from the seabed) of an area of 36x34 m were performed. The High-level planner running the view planning algorithm had 2 s of computing time.

Figure 6.3 highlights the effectiveness of the developed learning-sampling methodology for realizing the informed tree expansion. On the top, a snap of one of the performed surveys using the informed expansion strategy. In detail, the green lines delimit the inspection area. The gray voxels depict the discovered sea bottom, the yellow lines represent the generated tree, and the blue line highlights the selected best path. On the left-bottom side, a representation of the discovered map distribution learned online by the robot using the KDE technique. Such a distribution is employed by the sampling algorithm to generate samples in unexplored regions. For the sake of clarity, the histogram of 1k samples generated accordingly is reported on the right-bottom side. The learning-sampling method was used to expand the tree, reported in yellow on the top side of the image, which is thus steered toward the most promising areas.

Three configurations of the algorithm were tested. As described above, the Voronoi-based expansion (the standard RRT strategy) was compared with the hereby proposed informed expansion that utilizes the KDE methodology for estimating the discovered map. In addition, a balanced solution that makes use of both the Voronoi and the informed expansions was investigated. With regard to the sampling strategy described in Section 6.2.2, such algorithm configurations are achieved by considering sampling thresholds of $t_s = 0$, $t_s = 1$, and $t_s = 0.5$, respectively. The results reported in the following aim at demonstrating the effectiveness of the novel informed expansion strategy. Thus, since it is a preliminary validation, only the 50-50 sampling ratio for the balanced configuration is investigated. However, different sampling ratios, e.g., 25-75 and 75-25, could be adopted. Please note that the

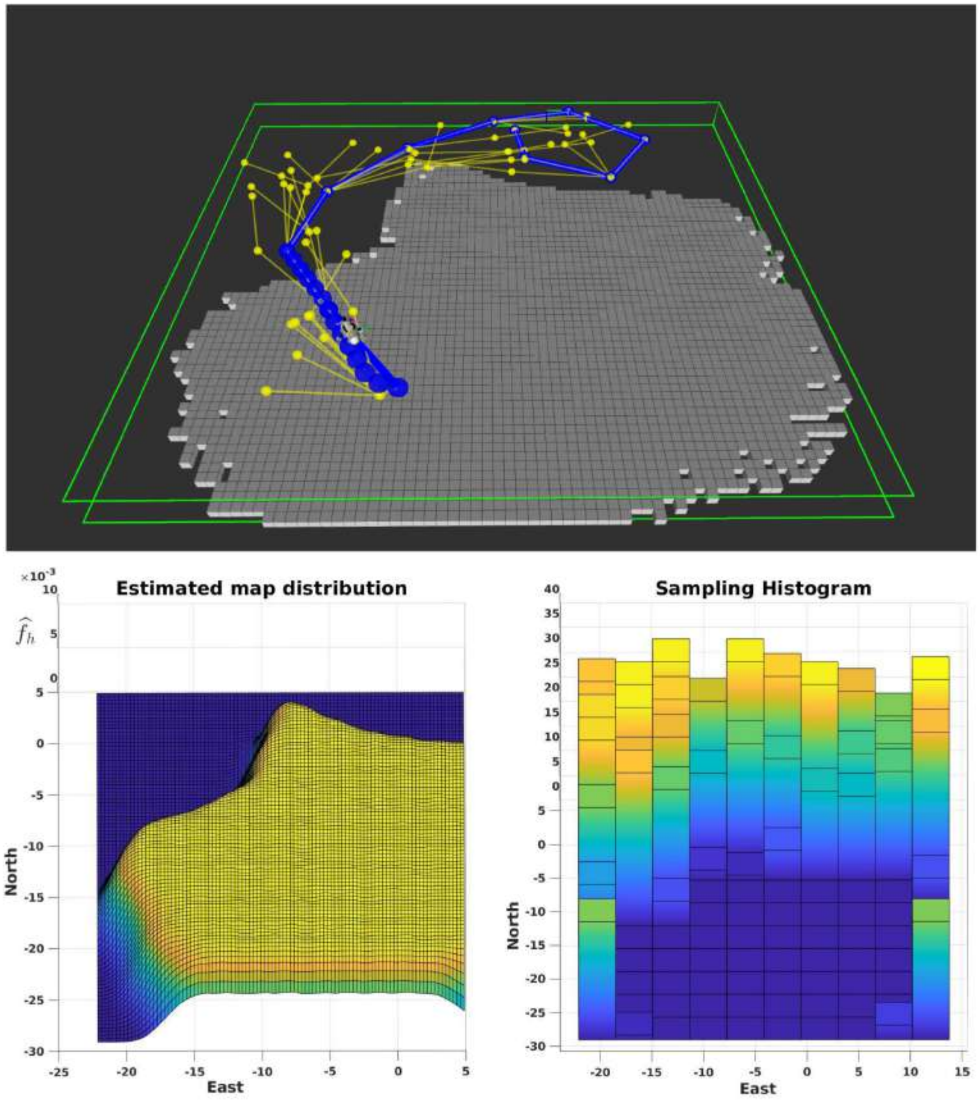


Figure 6.3: The proposed learning-sampling methodology utilized to develop the informed expansion. On the top, a snap collected during one of the surveys depicting the inspection area (delimited by the green lines) and the discovered map, reported in gray. The map distribution learned online by the robot with the KDE strategy is shown on the left-bottom side. On the right, the histogram of 1k samples generated in the non-explored area thanks to the learning-sampling methodology is reported. The tree informed expansion is realized by means of this learning-sampling strategy, which allows steering the tree (shown in yellow) towards the most promising regions.

choice of the 50-50 sampling ratio does not depend on the inspection area shape.

Regarding the exploited VI formulation, it is worth highlighting that only the entropy-based formulation (Eq.3.23) has been tested. In fact, as in Chapter 5, with such formulation the RHCA achieved better results.

Being the sensor-driven RHCA fully probabilistic, for each algorithm configuration, denoted as *RHCA - Voronoi* ($t_s = 0$), *RHCA - Informed* ($t_s = 1$), and *RHCA - Balanced*($t_s = 0.5$), 10 simulations with random starting AUV positions were performed, and the performance was evaluated in terms of the path length required to achieved the 80%, 90%, and 95% of coverage of the area. The mean values of the 10 runs were considered. The results are summarized in Table 6.1.

The proposed informed expansion enhances the performance of the RHCA. The considered coverage levels are achieved with shorter paths when the KDE methodology is used for generating the samples for expanding the tree toward unexplored regions than exploiting the state-of-the-art Voronoi bias. The higher the target coverage level, the more the RHCA benefits from the developed Informed expansion solution. In fact, when the 95% of coverage is demanded, which is the most challenging task, the KDE-based sampling strategy leads to a mean path length reduction of about 20 m (15% of the total mean path length). Besides, even for reaching the 80% and 90% of coverage, the mean length is significantly reduced.

Table 6.1 shows that the best solution is using a balanced expansion strategy. That is, at each iteration of the view planning algorithm, there is a 50-50 chance of using either the Voronoi-biased or the Informed expansion. Thus, a trade-off between the exploration of the workspace and a guided expansion toward the goal, i.e., unexplored regions, is exploited. Using only the here proposed KDE-based methodology for sampling new configuration improves the performance compared to the Voronoi bias. However, only focusing on the expansion toward new areas prevents the tree from exploring the space. Thus, it limits the view planning algorithm performance.

For the sake of completeness, for each view planning algorithm configuration, an example of the tree expansion is reported. In detail, Figure 6.4 reports the use of the Voronoi-biased expansion, while Figure 6.5 shows the effect of the proposed Informed strategy. The former ensures that the tree rapidly covers the area of interest, but it does not consider the already discovered map, and thus, it could lead to longer paths. The latter steers the tree toward new areas, focusing the view planning algorithm on looking in the most promising areas. Finally, Fig. 6.6 depicts the balanced approach. This latter solution combines the benefits of the aforementioned strategies. It drives the tree toward the unexplored regions while maintaining a fast exploration of the workspace.

Table 6.1: Simulation outcomes

	Coverage level 80%	
Method	***Path length mean [m]***	***Path length standard deviation [m]***
RHCA - Voronoi	85.69	12.88
RHCA - Informed	79.63	10.23
RHCA - Balanced	75.15	11.99

(a) Results for 80% of coverage

	Coverage level 90%	
Method	***Path length mean [m]***	***Path length standard deviation [m]***
RHCA - Voronoi	116.98	21.04
RHCA - Informed	99.18	10.68
RHCA - Balanced	100.34	7.67

(b) Results for 90% of coverage

	Coverage level 95%	
Method	***Path length mean [m]***	***Path length standard deviation [m]***
RHCA - Voronoi	140.00	26.93
RHCA - Informed	120.71	20.31
RHCA - Balanced	118.54	12.58

(c) Results for 95% of coverage

6.4 Experimental results

An experimental campaign for testing and validating the proposed advances for the view planning algorithm was conducted in July 2021 at the CSSN

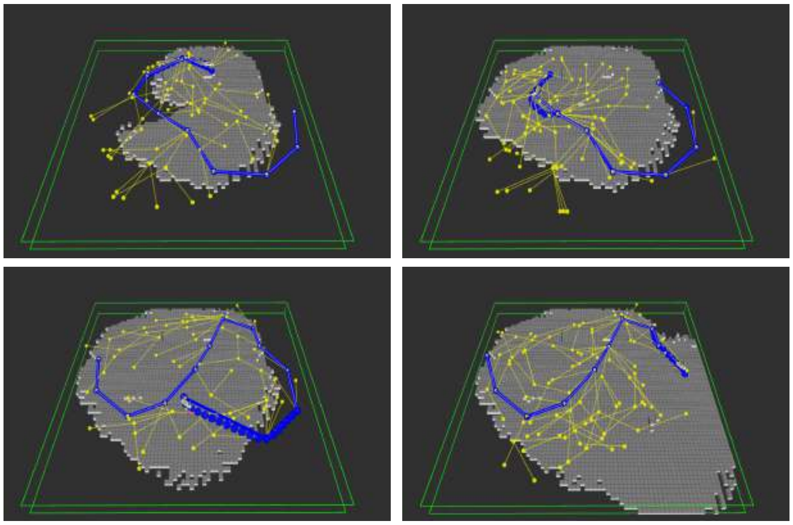


Figure 6.4: Examples of the generated inspection tree (in yellow) by using the Voronoi-bias during the expansion. Each yellow point represents an AUV configuration. The green lines delimit the inspection area, while the blue one depicts the selected inspection path, i.e., the best branch. The Voronoi bias ensures that the tree rapidly explores the workspace.

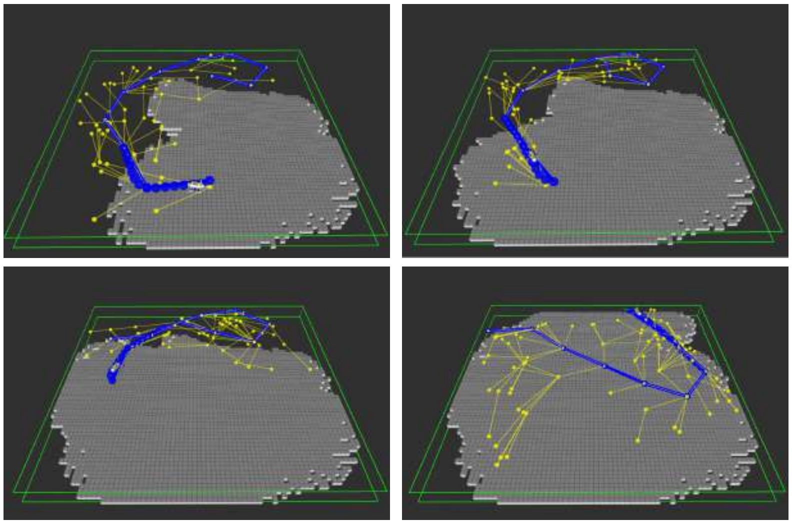


Figure 6.5: The tree expansion when only the KDE-based Informed strategy is exploited. The tree is biased through non-explored regions.

basin in La Spezia, Italy. The same framework shown in Chapter 5 was utilized. The improvements proposed in this Chapter have been implemented using C++ language, and the OMPL library [113]. Regarding the KDE-based map density estimation technique, it was realized exploiting the mlpack, a flexible C++ machine learning library [121]. The advanced view planning algorithm was run online on the main computer of FeelHippo AUV (Section 1.2).

Since conducting experimental sea trials is time and cost expansive, two configurations of the view planning algorithm were tested during the experimental campaign. The *RHCA - Balanced*, that fuses the Voronoi-bias with the developed informed strategy and emerged as the best solution during the validation process, was compared with the *RHCA - Voronoi* that utilizes the state-of-the-art Voronoi-based expansion.

Thus, 5 coverage surveys of an area of 36x34 m for each configuration were conducted. The surveys were performed autonomously by FeelHippo AUV utilizing the BlueView M900 2D FLS, whose range was set to 10 m. As for the simulations, the surveys were evaluated in terms of path length required to achieved the 80%, 90%, and 95% of coverage of the area, Table

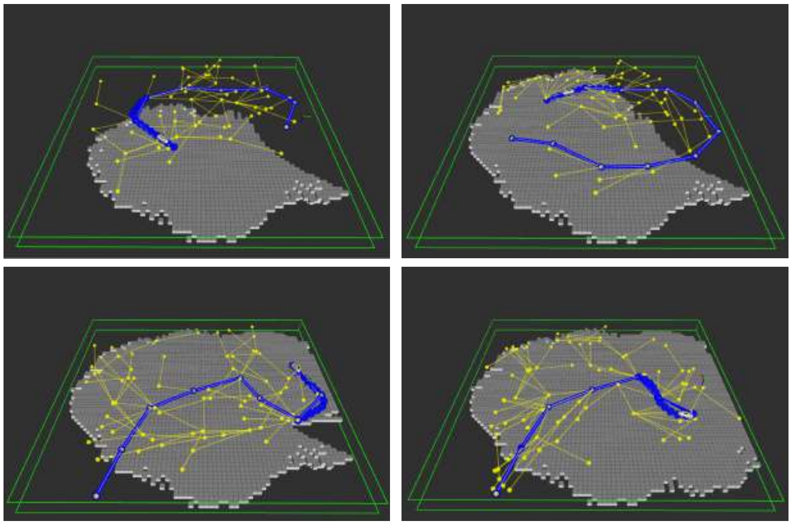


Figure 6.6: Four snapshots depicting the tree generated using the balanced expansion approach. By randomly exploiting both the Voronoi and the Informed bias, it enhanced the performance of conducted autonomous coverage surveys.

6.2 reports the mean value as well as the standard deviation of the 5 trials.

As expected from the simulations, the *RHCA - Balanced* outperformed the state-of-the-art *RHCA - Voronoi* in all the benchmarks. By mixing the uniform distribution and the map-aware sampling policy, the balanced configuration guides the expansion of the tree toward the most promising areas while guarantying it well-explores the workspace. The *RHCA - Balanced* reached a mean path length reduction of 18.38 m, 31.80 m, and 18.40 m (21.5%, 25.7%, and 12.7% of the total mean path length) for the 80%, 90%, and 95% of coverage, respectively.

To benchmark the achieved experimental results, Table 6.2 reports the outcomes of the lawnmower survey described in Section 5.5. The lawnmower pattern was designed considering the FLS characteristics by a skilled operator. In the smooth scenario where the real tests were conducted, the lawnmower outperformed the developed solution in the best configuration, i.e., *RHCA - Balanced*. However, the same considerations stated in the previous Chapter hold. For designing the lawnmower survey, a perfect knowledge of the scenario and a non- negligible pre-mission time is needed. In addition, the AUV passively stores the acquired data, and thus multiple attempts should be performed. By using the proposed sensor-driven RHCA strategy, the survey designing time (i.e., the pre-mission time) is eliminated. Besides, the AUV monitors the acquired data and can ensure the insonification of a required portion of the seafloor, avoiding multiple attempts. Being fully probabilistic, the results can vary but are reasonable compared with the state-of-the-art lawnmower pattern. Besides, thanks to the proposed KDE-based methodology, the vehicle learns where to sample new configurations for steering the tree toward unexplored regions of the workspace and enhances the strategy performance getting closer to the lawnmower.

It is worth highlighting that the proposed map-aware sampling strategy, based on learning the discovered map distribution by means of the KDE technique, was run online on the main computer mounted FeelHippo AUV. Thus, it does not require dedicated hardware and can be used even on compact and lightweight robots.

6.5 Main contribution

This Chapter proposed advances for enhancing the view planning algorithm utilized in the RHCA framework.

Table 6.2: Experimental results

	Coverage level 80%	
	Path length	***Path length***
Method	***mean [m]***	***standard deviation [m]***
Lawnmower	65.34	-
RHCA - Voronoi	85.49	5.91
RHCA - Balanced	67.12	11.58

(a) Results for 80% of coverage

	Coverage level 90%	
	Path length	***Path length***
Method	***mean [m]***	***standard deviation [m]***
Lawnmower	72.37	-
RHCA - Voronoi	123.56	21.63
RHCA - Balanced	91.76	14.84

(b) Results for 90% of coverage

	Coverage level 95%	
	Path length	***Path length***
Method	***mean [m]***	***standard deviation [m]***
Lawnmower	76.39	-
RHCA - Voronoi	145.08	5.93
RHCA - Balanced	126.68	11.00

(c) Results for 95% of coverage

For enabling the use of a complete rewiring routine, the RRT-based view planning algorithm has been redesigned. It uses two parallel threads for speeding up the tree construction. By the time a new node is generated and its visibility is computed through a ray-casting procedure, the rewiring routine is executed for the node created at the previous iteration. The scope of this latter procedure is twofold: it finds the best parent and checks if the new node can improve other branches.

The developed informed expansion strategy represents the major contribution. For the first time to the best of the author's knowledge, the KDE technique, which has found a large use in statistics and ML, is exploited to estimate the distribution density function of the discovered occupancy map. Combined with a rejection sampling algorithm, the estimated density function is used to draw sample configurations in the unexplored areas of the workspace. Such samples can be used to biasing the expansion of the tree toward the most promising region. That is, the robot learns where to sample the configurations for guiding the planning algorithm.

Simulations as well as real experimental results, collected during sea trials conducted in July 2021 at the CSSN basin in La Spezia, Italy, show that the informed expansion strategy improves the performance of the RHCA framework. In particular, a comparison of the three expansion strategies was provided. The results highlight that the state-of-the-art Vornoi-based expansion was outperformed by the informed approach. However, mixing the two strategies emerged as the best configuration. In fact, using both the uniform distribution and the map-aware sampling policy offers a good trade-off between a goal-guided expansion and the exploration of the workspace.

Chapter 7

Randomized MPC for view planning in AUV seabed inspections

Chapter 5 presented the development and testing of an RRT-based view planning algorithm utilized for autonomous FLS seabed inspections. Then, in Chapter 6 advances for improving the performance of the algorithm were proposed and validated through both realistic simulations and an experimental campaign. In particular, the state-of-the-art Voronoi biased expansion was compared with an innovative informed strategy based on estimating the distribution of the discovered map.

This Chapter aims at investigating a different approach based on the use of an RMPC [122] strategy for generating viewpoints. The main findings reported in this Chapter were published in [123]. As showed in Chapter 6, the performance of RRT-based solutions is affected by the sampling strategy utilized for expanding the tree. In addition, in order to grow the tree, the motion between a tree's node and a newly generated node must comply with the robot motion constraints. As this work is concerned, since kinematic constraints are considered, the motion feasibility check can be computed in a reasonable amount of time. Still, in light of expanding the view planning methodologies to dynamic constraints, a BVP, which is generally a difficult and computationally expensive problem, shall be solved. The solution investigated in this Chapter, by directly applying feasible control inputs to the robot motion equations, realizes a forward propagation that generates proper vehicle paths, avoiding the motion checking routine.

The same framework presented in Chapter 5 is exploited. The RMPC strategy is used as the view planing algorithm in the High-level planner instead of the previously discussed RRT-based solution. By selecting random controls feasible for the considered vehicle, the algorithm randomly propagates the AUV state and thus generates viewpoint candidates. Each viewpoint is evaluated according to the discovered map. The map that can be discovered from a viewpoint is generally evaluated through a ray casting process. However, sensors such as the FLS considered in this work has wide FoV and long ranges, making the ray casting evaluation slow and computationally expansive. As a consequence, the number of candidates that can be generated in a given computational time is limited and degrades the performance of the view planning algorithms. To overcome this limitation, the KDE methodology described in Chapter 6 is utilized for estimating the density of the discovered map and rapidly evaluating the viewpoint candidates.

7.1 Randomized MPC for View Planning

While Chapter 5 proposed the use of RRT-based view planning solution for seabed inspections, here, an MPC approach is investigated. The developed view planner algorithm utilizes an MPC strategy for generating an inspection path that steers the AUV toward unexplored regions. In particular, the view planner selects among the predicted paths the best path that guides the AUV toward unexplored regions. As described in Chapter 5, when the robot reaches the first viewpoint of the selected path, the process is repeated realizing a receding-horizon approach. In fact, the inspection framework detailed in Chapter 5 was utilized. Following the idea proposed in [124], a predicted inspection path is a sequence of sub-paths, where each one starts from an AUV configuration and consists of a control input $\mathbf{u} = [u, r]^\top$, where u is the surge velocity and r the yaw rate, which is used to evolve the AUV state. Thus, the simulated paths can be represented by using a tree structure, initialized at the AUV current location, where the nodes are the AUV states and the edges denote the sub-paths.

Therefore the view planner solves the following optimization problem:

$$\begin{array}{rl} \max_{\xi_0,\{\mathbf{u}\}} & \mathcal{G}_{\{\xi\}_i} \\ \text{s.t.} & \xi_i = f(\xi_{i-1}, \mathbf{u}_i) \\ & \mathbf{u} \in \mathcal{U} \\ & \xi \in \mathcal{W}\ , \end{array} \tag{7.1}$$

where $\mathcal{G}_{\{\xi\}_i}$ is the optimization function that corresponds to the expected discovered map along the path constituted by the predicted configurations $\{\xi\}_i$, ξ_0 is the current AUV pose, and $\mathcal{U}$ denotes the space of feasible control inputs for the AUV.

As discussed in Section 5.2, since the optimization function $\mathcal{G}_{\{\xi\}_i}$ is a black-box function, i.e., the relation between the function and the AUV configurations is unknown, it is not possible to compute the optimal control sequence $\{\mathbf{u}\}$ by means of a deterministic optimization algorithm [125], [116]. Therefore, the optimization problem in Eq.7.1 shall be solved by directly evaluating the expected gain of the predicted paths, which means computing the IG of the associated configurations and selecting the one with the highest gain. To generate an appropriate set of possible paths that well explore the workspace, which is of utmost importance for finding a near-optimal solution for the optimization problem, a control input selection strategy must be adopted. In the context of this work, to avoid the use of a manually defined set of controls, a set of random control inputs is selected for each AUV configuration. However, to guarantee the exploration of the workspace, the control inputs shall be well-distributed. Thus, a set of desired controls $\mathcal{U}_d = \{(u_d, r_{d,k})\}_{k=0,\dots,Nc}$ was defined. Such set of controls assumes a constant desired surge velocity u_d, and, given the yaw maximum changing rate r_{max}, uniformly samples the range $[-r_{max}, r_{max}]$. Finally, for each tuple of the desired control set $(u_d, r_{d,k})$, a control input (u, r) in sampled with a Gaussian distribution. The idea is summarized in Fig.7.1.

7.2 Fast evaluation of the optimization function via KDE

As discussed above, solving the optimization problem of Eq.7.1 means finding among the generated paths, the one that has the highest gain, i.e., is expected to discover more new regions. In fact, since the optimization function $\mathcal{G}_{\{\xi\}_i}$ is a black-box function, to known the gain of a path, it must be evaluated. That

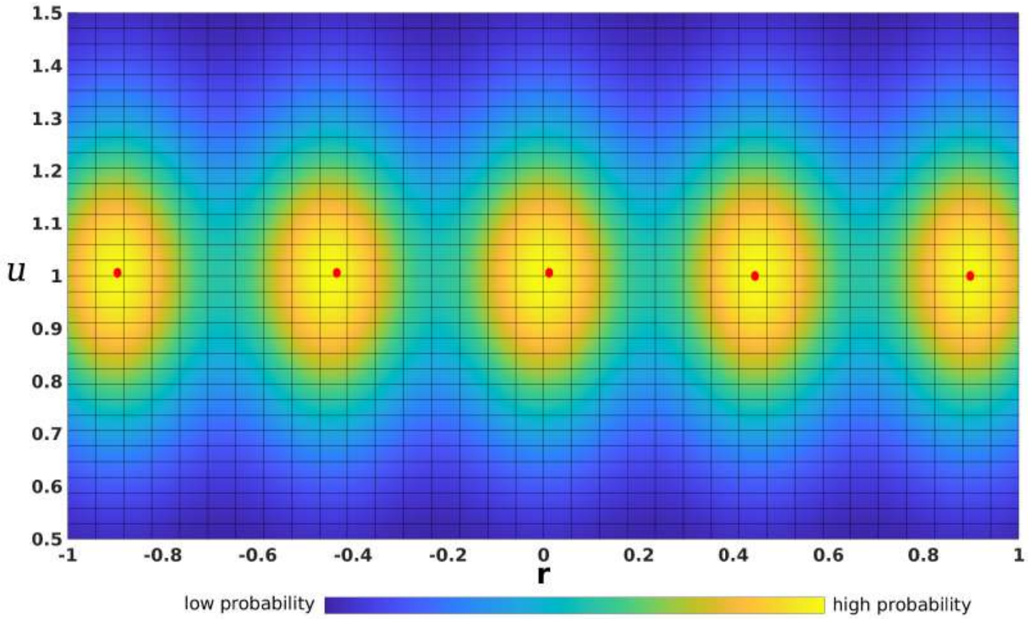


Figure 7.1: The control input selection strategy utilized in this work. The red points depict the desired control set, which assumes a constant surge velocity u_d and uniformly samples the set $[-r_{max}, r_{max}]$. To predict the AUV paths, for each configuration, a set of random controls is sampled around the desired set with a Gaussian distribution. In particular, for each tuple of the desired control set $(u_d, r_{d,k})$, a control input (u, r) is randomly selected. The figure shows the sampling probability of the set of possible control inputs. Points in the yellow regions are more likely to be sampled than the ones in the blue. It is worth mentioning that the x- and y-axis report the yaw turning rate and the surge velocity normalized with respect to the maximum rate r_{max} and desired velocity u_d, respectively.

is, for each configuration that composes a path, Eq.3.20 must be computed by using a ray casting procedure. Nevertheless, this procedure for sensors with wide FoV and long ranges, such as the FLS considered in this work, is computationally expansive and requires a lot of time, reducing the number of candidates that can be evaluated in a given computational time, and thus degrading the performance of the view planning algorithm. RRT-based solutions, as proposed in Chapter 5, make use of a Voronoi-bias expansion (or the informed sampling strategy described in Chapter 6) strategy that ensures that the generated tree, even if it does not contain a plethora of nodes, well explores the workspace. Thus, the optimization function $\mathcal{G}_{\{\xi\}_i}$ can be considered as the sum of the VI of all the voxels visible from a path, which are computed using the classic ray casting procedure for each configuration that composes the path.

For what concerns the RMPC approach proposed here, the ray casting procedure cannot be employed. In fact, by using such strategy for evaluating the optimization function, either a small number of paths can be generated or only short paths (i.e., a short prediction horizon for the MPC problem that leads to poor results) can be evaluated. To overcome this limitations, a novel method for rapidly evaluating the viewpoint candidates based on the KDE methodology for estimating the distribution density function of the discovered map has been utilized (see Section 6.2.2). To evaluate whether a viewpoint candidate ξ is leading the AUV toward non explored regions, the KDE estimator can be exploited as measure of the configuration gain:

$$\widehat{f}_h(\xi_{(x,y)}) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{\xi_{(x,y)} - m_i}{h}\right) \tag{7.2}$$

Computing $\widehat{f}_h(\xi_{(x,y)})$, denoted as $\widehat{f}_h(\xi)$ in the following for the sake of clarity, is greatly faster than performing the ray casting procedure. Thus, more paths can be evaluated. It is worth highlighting that $\widehat{f}_h(\xi)$ acts as a surrogate model for the optimization problem. In particular, by using the KDE approach, the visibility of a configuration is not computed, but $\widehat{f}_h(\xi)$ gives as a measurement that allows to quantify the quality of a configuration, i.e., how it leads the vehicle to unexplored areas.

Finally, it is possible to define the optimization function $\mathcal{G}_{\{\xi\}_i}$ as:

$$\mathcal{G}_{\{\xi\}_i} = \mathcal{G}_{\{\xi\}_{i-1}} + (\max\{\widehat{f}_h(m_i)\}_{i=1}^{n} - \widehat{f}_h(\xi))e^{\gamma_\psi \Delta_\psi(\xi_0,\xi_i)}e^{\lambda_d(distance(\xi_0,\xi_k))} \tag{7.3}$$

where $\Delta_\psi(\xi_0, \xi_i)$ computes the heading changes along the path, $\gamma_\psi \in R$ is a penalization factor for curvy paths, and $\lambda_d \in R$ penalizes long paths.

7.3 Validation of the RMPC view planning algorithm for autonomous inspections

To validate the developed view planning RMPC algorithm with the fast evaluation method, realistic simulations were performed by means of the UUV Simulator. As done for the validating the RRT-based solution (Chapter 5 and Chapter 6), FLS-based inspection surveys at a constant altitude (2 meters from the seabed) of an area of 36x34 m were performed. The Teledyne BlueView M900 2D FLS was simulated with a range of 10 m.

The RMPC view planning solution has been developed using the OMPL library [113]. It was implemented using the C++ programming language and integrated within the ROS framework, which the FeelHippo AUV software architecture is based on.

The set of desired control inputs was defined by considering the desired velocity u_d as 0.4 m/s and uniformly sampling nine values of angular rate in the range [-0.5, 0.5] rad/s. Therefore, for each AUV configuration, nine control inputs were randomly sampled by using the strategy presented in Section 7.1 (Fig.7.1). The process was repeated to generate as many configurations as possible within the given computational time, set to 2s. For the sake of clarity, a visualization of an example of the generated paths is depicted in Fig.7.2. Please note that when the AUV reached the first configuration of the selected path, the RMPC was run again, and by using the previously computed solution to initialize the algorithm, a receding-horizon fashion was achieved.

To assess the performance of the developed RMPC strategy, since it is a probabilistic method, the same approach utilized for the RRT-based algorithm was used. Ten simulations with random starting AUV positions were performed. The inspection surveys were evaluated in terms of the mean path length required to achieve the 80%, 90%, and 95% of coverage of the area. Besides, to have a benchmark to compare the RMPC method with a state-of-the-art solution, the experimental results achieved with the lawnmower survey described in Section 5.5 were considered, see Table 7.1.

Firstly, it is worth analyzing the benefits of using the KDE evaluation strategy (Eq.7.2) compared with the classic ray casting procedure. Using the latter, the RMPC managed to generate a mean of 70-120 configurations within the given computational time (2s), which was not enough for finding good solutions. In fact, it resulted in a short prediction horizon, and the

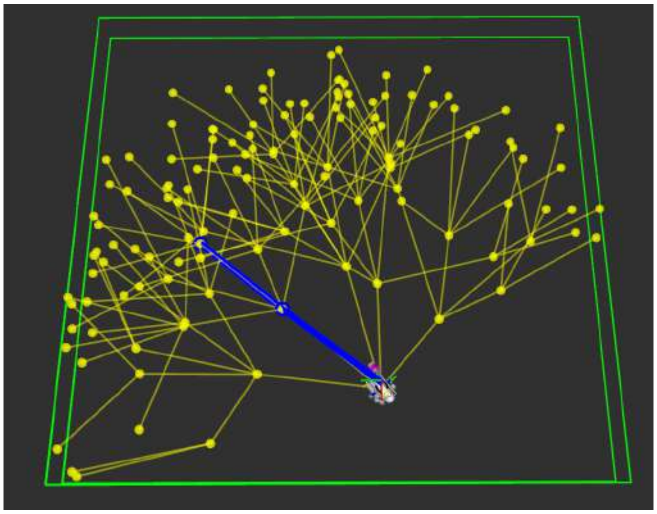


Figure 7.2: Example of the generated paths (in yellow) by the developed RMPC. In the depicted case, five control for each configuration were randomly sampled. The green lines delimit the inspection area. Each yellow point represents an AUV configuration. The blue line depicts the selected inspection path.

AUV was not able to achieve the target coverage levels. The developed KDE evaluation strategy speeded up the process. It allowed to generate a mean of 12k configurations for each run of the RMPC algorithm, and thus it increased the performance of the conducted coverage surveys.

Table 7.1 reports the outcomes of the inspection mission autonomously conducted by the AUV using the developed RMPC view planning algorithm with the KDE evaluation strategy.

Although the mean values of the RMPC strategy are greater than the ad-hoc designed lawnmower path, it could represent an alternative since it eliminates the pre-mission time. Indeed, it does not require a skilled operator for designing the survey and thus could easily be used by non-specialized end-users.

It is worth noting that the tests were conducted in smooth simulated sea bottom scenarios. When the outcomes of the FLS measurements are not predictable, the RMPC solution, as the RRT-based algorithm, could avoid repeated attempts.

The results of the RMPC should be compared with the RRT solution (Section 6.1). The RMPC algorithm, based on a different approach, achieved similar results to the RRT-based solution when the Voronoi-biased expansion is utilized. However, the RRT view planning algorithm utilizing the informed and the balanced expansion represent the best solutions. These latter methodologies, by guiding the tree expansion toward the unexplored areas, outperformed the RMPC algorithm.

Finally, it is worth highlighting that the fast evaluation methodology allowed to generated a wide set of viewpoint candidates, but it does not consider the effect of possible protruding objects and only evaluates whether a viewpoint steers the robot toward unexplored areas. On the other hand, the ray casting routine, by computing the visibility of a viewpoint, can estimate the effect of protruding objects and deal with more complex environments. However, it is time expansive and limits the number of generated viewpoints. Thus, a combined strategy that uses the fast evaluation methodology for generating a plethora of paths, and then the ray casting routine executed only for the most promising directions could be a solution for enhancing the performance of both the RMPC and RRT methodologies.

To enhance the performance of the RMPC solution, the use of the estimated map distribution for generating the control inputs could be investigated.

Table 7.1: RMPC outcomes

	Coverage level 80%	
Method	***Path length mean [m]***	***Path length standard deviation [m]***
Lawnmower	65.34	-
RHCA - RMPC	92.76	15.40
RHCA - RMPC - best run	66.88	-

(a) Results for 80% of coverage

	Coverage level 90%	
Method	***Path length mean [m]***	***Path length standard deviation [m]***
Lawnmower	72.37	-
RHCA - RMPC	118.65	24.65
RHCA - RMPC - best run	75.89	-

(b) Results for 90% of coverage

	Coverage level 95%	
Method	***Path length mean [m]***	***Path length standard deviation [m]***
Lawnmower	76.39	-
RHCA - RMPC	137.07	22.42
RHCA - RMPC - best run	98.70	-

(c) Results for 95% of coverage

7.4 Major contribution

The use of an RMPC algorithm for conducting autonomous seabed inspections exploiting an FLS was investigated in this Chapter. Firstly, a sampling strategy for randomly generating possible inspection paths that well explore the workspace was developed. Then, the KDE solution for estimating the density of the discovered map has been used for creating a fast methodology for evaluating the generated paths. Thanks to this evaluation strategy, the RMPC algorithm managed to generate several more configurations than by using a ray casting procedure and achieved outstanding results. The developed view planning algorithm was validated with realistic simulations made through the UUV Simulator and the inspection framework presented in Chapter 5. In particular, the RMPC algorithm was used as the High-level planner, instead of the RRT-based solution.

The solution presented in this Chapter will be tested in real sea trials as soon as possible. A combined solution that fuses the advantages of the designed KDE evaluation methodology and the ray casting procedure will be investigated. Besides, future works could also involve using a Reinforcement Learning technique to make the robot able to learn the best control inputs.

Chapter 8

Sensor-driven target-aware autonomous inspections

This research work has been driven by two aspects that have become of utmost importance in the field of modern robotics: meaningfully perceiving and modeling the surroundings and autonomously conducting the assigned task.

Regarding the former, i.e., understanding the environment, a CNN-based ATR strategy for recognizing OPIs was presented in Chapter 4. Such a solution enables an AUV to meaningfully gather knowledge of the surroundings. However, it does not fulfil all the required tasks. In particular, it does not create a model of the environment that can be used to achieve the assigned task and allows to define the areas containing the localized OPIs. This could be pivotal for various applications ranging from defining high-risk areas in MCM applications (and thus guiding the interventions or designing safe routes), to guide divers for archaeological missions, or planning more accurate inspections paths using different sensors, to mention a few.

The latter aspect was tackled in Chapter 5 (and then in Chapter 6 and Chapter 7), where the development of a framework for sensor-driven coverage surveys was reported. The framework was used for autonomously conducting FLS seabed inspections. To guarantee an adequate coverage of the area of interest, the framework, based on a receding horizon approach, utilizes an occupancy map that modelled the discovered regions for computing the inspection survey. The map provided feedback of the already enlightened seabed to the planning algorithm that could thus guide the AUV towards complete coverage of the area of interest. Besides, the reconstructed map could be employed to predict possible collisions of the vehicle and thus plan feasible paths online. The mapping solution also provides as output a reconstruction of the seabed. Nevertheless, it does not consider the presence of OPIs. That is, a semantically annotated occupancy map is not realized ([67], [68]), limiting the information contained in the created model of the world. Such knowledge could help human operators in analyzing the collected data and could be employed by a robot to achieve complex tasks, e.g., exploring unknown environments and simultaneously searching OPIs.

As a consequence of these considerations, a semantically annotated occupancy map could represent a solution for the highlighted deficiencies of the two aspects discussed above. In particular, by considering the ATR findings into the mapping strategy, the regions that are most likely to contain OPIs can be identified and used to design more detailed inspections or delineate potentially hazardous areas. Therefore, in this Chapter a framework for creating a semantically annotated map that fuses the ATR methodology detailed in Chapter 4 and the sensor-driven inspection framework reported in Chapter 5 is investigated. The proposed strategy is validated with realistic simulations made by means of the UUV Simulator. The results reported in Section 8.3 show that the developed solution allows the AUV to cover the inspection area and recognize and localize the OPIs.

8.1 Semantically annotated occupancy map

To realize the semantic occupancy mapping strategy, the OctoMap framework [66], which constitutes the backbone of the solution presented in [9] and used in this work (see Chapter 5), has been extended. In its standard version, as described in Section 3.3, OctoMap allows to represent free and occupied areas using nodes, usually known as voxels, which represent the space contained within a cubic volume. OctoMap utilizes the octree data structure to efficiently update the voxels probability of being occupied by using a Bayesian approach. Thus, each voxel has a probability P_o of being occupied. The semantic occupancy map was obtained by adding the semantic data to the voxels. In detail, each voxel was extended to include the probability P_a of belonging to an OPI. Therefore, each voxel holds:

- the probability P_o of being occupied;
- the probability P_a of containing an OPI.

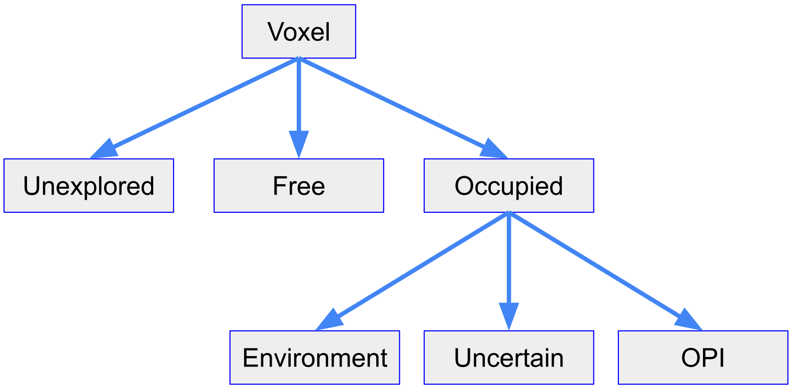


Figure 8.1: The voxel classification policy used to create a semantically annotated map. It extends the standard OctoMap label map by adding to each occupied voxel the probability of belonging to an OPI. This probability is updated by using the outputs of the ATR system detailed in Chapter 4. The developed semantic mapping strategy allows creating a meaningful model of the area of interest that can be used to localize the OPIs and for planning and executing object-aware tasks.

The here proposed mapping strategy was developed to create a semantic model of the world that could be used for identifying the regions that are most likely to contain OPIs or exploited on-board by a fully autonomous AUV. Therefore, the goal is to classify the voxels of the area of interest as free or occupied and determine which occupied voxel belongs to an OPI. As a consequence, each voxel can be labeled as *unexplored*, *free*, *occupied-environment*, *occupied-uncertain*, and *occupied-OPI*. In detail, occupied voxels are classified considering P_a as follows. While small values of P_a determine *occupied-environment* voxels, P_a close to one determines *occupied-OPI* voxels. Intermediate values are used to define *occupied-uncertain*, which are voxels with a non-negligible probability of containing an OPI but are not well classified yet. Fig. 8.1 summarizes the utilized voxel policy.

While the voxels' occupancy is updated by the mapping solution presented in [9], P_a is based on the results of the ATR solution. To this end, an update procedure shall be defined. Firstly, it is worth to note that by utilizing the fundamental assumptions on which the occupancy grid theory relies on, discussed in Section 3.3, P_a can be updated following the Bayesian law of Eq.3.10. Then, given the definition of the log-odds ratio, the elegant and numerically efficient update formulation of Eq.3.12 can be used. Thus, by applying the clamping policy, P_a can be updated by means of Eq.3.13. Since the mapping strategy was developed for mapping completely unknown environments, the non-informative prior assumption is employed for initializing P_a, i.e., $p_a(mi) = 0.5$.

8.1.1 Generating ATR-based point clouds

Regarding the measurement generation process, the ATR findings were used for computing the point clouds. The ATR system makes use of CNNs to detect and localize OPIs and it outputs the predicted object bounding boxes with the computed confidence level (see Section 3.2.2). As described in Section 3.1, assuming that the sea bottom imaged within a frame is dominantly flat, each bounding box can be accurately localized with respect to the FLS reference frame ($< F >$) (see Section 3.1), and thus by knowing the sensor mounting pose, with respect to the body frame ($< b >$). Then, for each ATR finding a point cloud can be generated by dividing the bounding box into cells and considering the center of each cell. For the sake of clarity, each cell center that represents the position of an element of the point cloud is expressed in spherical coordinates. Finally, the ISM for updating P_a is calculated by associating at each point the probability of belonging to an OPI, which is computed considering a Gaussian distribution as follows:

$$p_a\left(m_i \mid z_t^{ATR}\right) \approx p_a\left(m_i, c_{bb}\right) = p_c \; e^{-\frac{1}{2}\left(\frac{|m_i - c_{bb}|}{\sigma_c}\right)^2}, \tag{8.1}$$

where c_{bb} denotes the center of the detected OPI's bounding box, and p_c the computed confidence level. For the sake of simplicity, a constant variance σ_c is considered, and its value was selected so that p_c is distributed over the entire bounding box area.

It is worth noting that the noise and the lack of details in FLS images makes the OPIs recognition task challenging. Besides, the OPIs's rendering depends on the FLS characteristics and the insonification angle, i.e., the viewpoint. In fact, during an inspection survey, the AUV could enlighten the same OPI from different viewpoints. All these patterns can lead to non detected objects or wrong detections. Therefore, for each FLS image for which the ATR system does not find any targets, the enlightened voxels are updated with $p_a \left(m_i \mid z_t^{ATR}\right) = 0.45$ that, in the Bayesian context, reduces the probability of belonging to an OPI. This strategy was adopted to reduce the effects of wrong detections on the reconstructed semantic map and increase the OPIs localization accuracy.

8.2 Object uncertainty-aware view planning for coverage

The modified mapping solution was employed with the autonomous inspection framework (see Chapter 5). The view planning algorithms were designed to cover the area of interest with a given sensor, which in the context of this research an FLS was considered. As detailed above, the developed algorithms aim at optimizing the inspection survey by using the reconstructed map as feedback. That is, they realized a sensor-driven solution that ensured adequate coverage levels. In particular, in FLS inspections, it corresponds to ensuring that the seabed is enlightened. Therefore, if during the mission the ATR solution is exploited, the target of interest can be recognized and localized. Besides, the semantically-annotated mapping strategy allows determining the regions that are most likely to contain OPIs. However, as presented in the previous Chapters, the view planners cannot consider the knowledge provided by the advanced mapping solution and only tried to compute the shortest path for covering the area. Thus, they did not consider the goal of reconstructing a semantic map of the environment actively. With regard to the mapping solution described above, this corresponds to classify the voxels as *free*, *occupied-environment*, and *occupied-OPI* and reducing as much as possible the voxels labelled as *occupied-uncertain*.

To enable the view planner to consider the object uncertainty into the map, the optimization function was modified. The RRT-based algorithm with the balanced KDE expansion strategy (see Chapter 6) was considered. Therefore, Eq.5.2 has been modified. By using the concepts introduced in Section 3.4.2, the voxel uncertainty of containing an OPI can be expressed by its entropy:

$$I_a(x) = -P_a(x)ln(P_a(x)) - \overline{P}_a(x)ln(\overline{P}_a(x)) \ , \tag{8.2}$$

where $\overline{P}_a(x)$ is defined as $\overline{P}_a(x) = 1 - P_a(x)$.

Thus, the uncertainty observed along a branch can be estimated as:

$$G_{\mathcal{B}a} = \sum_{\forall x \in \mathcal{X}_{\mathcal{B}}} I_a(x) \ , \tag{8.3}$$

where $\mathcal{X}_{\mathcal{B}}$ denotes the observable voxels along the branch $\mathcal{B}$ (as in Section 5.2).

This target uncertainty-based component can be combined with the BIG in Eq.5.1 for defining an appropriate branch gain model. Regarding this research work, $G_{\mathcal{B}a}$ was utilized to promote branches that lead the AUV toward unexplored regions and allow to reduce the target uncertainty. Thus, the branch gain was designed as follows:

$$\mathcal{L}_{\mathcal{B}} = \mathcal{G}_{\mathcal{B}} e^{\lambda_a G_{\mathcal{B}a}} \ . \tag{8.4}$$

$\mathcal{G}_{\mathcal{B}}$ denotes the gain function defined in Eq.5.2 and used for carrying the autonomous coverage surveys; $\lambda_a > 0 \in R$ is a tuning parameter weighing the target uncertainty.

While by using $\mathcal{G}_{\mathcal{B}}$ the algorithm selects the shortest path that guides the vehicle toward unexplored regions, $\mathcal{L}_{\mathcal{B}}$ promotes solutions that allow exploring the area and simultaneously reduces the OPIs localization uncertainty. Thus, it is worth highlighting that $\mathcal{L}_{\mathcal{B}}$ could lead to longer inspection surveys. In fact, enhancing the localization of the OPIs could require the vehicle to re-observe already explored areas.

8.3 Autonomous inspections for localizing targets

The proposed methodology for autonomous underwater inspections and simultaneously looking for object of interest has been validated with realistic simulations made by means of the UUV Simulator, where a dynamic model of FeelHippo AUV was implemented. The RHCA framework detailed in Section 5.1 was used together with the ATR strategy presented in Chapter 4. In detail, the developed semantically annotated occupancy mapping solution utilized the seabed reconstruction strategy detailed in [9] and the outputs of the ATR system. The Faster R-CNN architecture was selected for recognizing the OPIs in the FLS images. Regarding the view planning strategy, the RRT-basd algorithm made used of the object uncertainty-aware branch gain function described in Eq.8.4. The KDE balanced informed expansion methodology (see Section 6.2.2) was selected since it resulted as the best coverage solution (Section 6.3 and Section 6.4).

To create an appropriate simulation environment, three objects resembling a truncated cone and whose dimensions are 0.30 m of minor radius (top), 0.49 m of major radius (bottom), and 0.44 m height were placed on the sea bottom. It is worth highlighting that the selected object shape is a rendering of the OPI utilized in Section 4.3 to quantify the localization accuracy of the ATR system. To validate and asses the performance of the developed methodology, four inspections were conducted with a fixed AUV starting position and the three OPI placed in known positions, whose coordinates in the NED reference frame are reported in Table 8.1. Autonomous coverage surveys at a constant altitude (2 meters from the seabed) of an area of 36x34 m were performed. The BlueView M900 2D FLS with a range of 10 m was considered.

Table 8.1: OPI positions in the NED frame

	Position [m]
OPI 1	(10, -5, 4)
OPI 2	(-10, -4, 4)
OPI 3	(-2, 6, 4)

To quantitatively evaluate the proposed strategy, the travelled path length to achieve the 90% and 95% of the coverage were considered. In addition, to assess the OPIs localization precision, when the AUV reached the latter threshold, euclidean clusters were extracted from the voxels with $P_a > 0.7$. The centroid of each cluster has been considered as the closest OPI estimated position. Then, in order to define the regions containing the OPIs, for each cluster, the distance along the North and East directions of the cluster farthest point from the centroid has been calculated. Therefore, at the end of the survey, the regions that are most likely to contain the OPIs can be identified. Table 8.2 summarizes the results of the simulations.

As the coverage task is concerned, although the uncertainty-aware gain function lead the RRT view planning solution to slightly longer paths compared with the outcomes shown in Chapter 6, the AUV managed to ensure the coverage of the area of interest. During the autonomous surveys, the three OPIs were detected by the ATR system and the semantically annotated map was updated. Regarding the localization accuracy, all the targets of interest were localized with a maximum error of about 2 m. Besides, the identified object areas ensures that the OPIs are well localized. In fact, given the target dimensions and the computed maximum distance of the cluster points from the centroid, the OPIs were contained within the defined regions. This is of utmost importance in several applications, e.g, MCM operations for defining safe routes and plan intervention operations, archaeological campaigns for designing more accurate inspections, and so on.

In Fig. 8.2, the semantic map reconstructed during a survey is shown. Occupied voxels belonging to the environment are reported in grey, while coloured cubes depict voxels with a probability of containing an OPI greater than 0.7. In particular, green turns to red as the probability increases. Such a semantic map can be helpful for operators for the aforementioned tasks, or it can be used online by the robot. In particular, in this research work, the target uncertainty was used by the AUV to plan the inspection path. As discussed above, the exploitation of the gain function $\mathcal{L}_{\mathcal{B}}$ (Eq.8.4) promotes branches that lead the AUV toward unexplored regions and allow to reduce the target localization uncertainty. Fig. 8.3 and Fig. 8.4 highlight this behavior. In fact, instead of selecting the shortest path to complete the coverage task, the view planning algorithm preferred a slightly longer pattern (in blue) that permits observing the uncertain areas. That is, the vehicle selected a path that enables it to first re-observe the targets in order to reduce their localization uncertainty and then complete the coverage task.

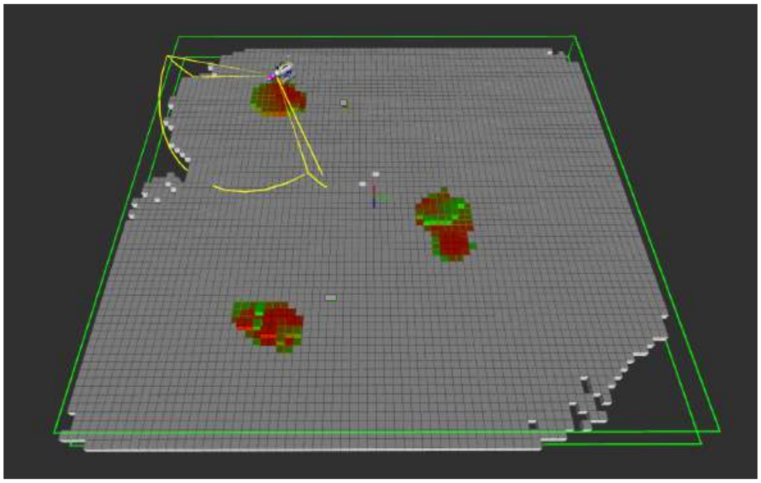


Figure 8.2: The semantically annotated map created during an autonomous coverage survey. Grey voxels depict occupied voxels belonging to the environment (seafloor). Coloured voxels delimit the regions with a high probability of containing an OPI. Green turns to red as the probability increases.

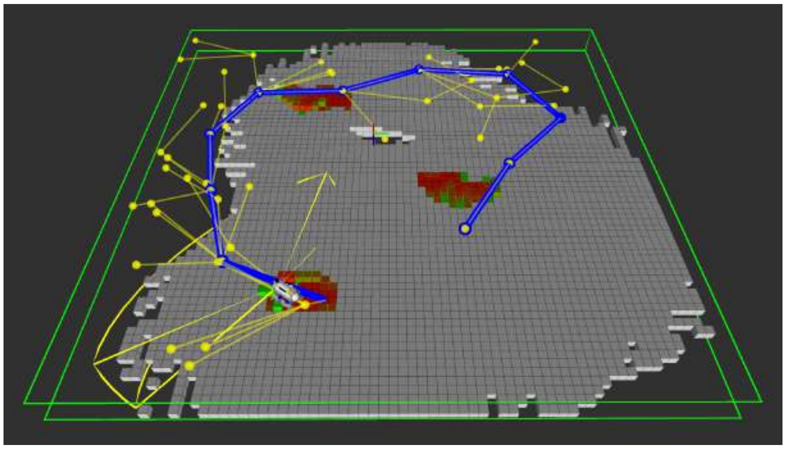


Figure 8.3: The semantic map is used online by the AUV. The knowledge about the OPIs is exploited for planning the survey. By using the target uncertainty-aware gain function, the planner selected a path (in blue) that both steers AUV toward the coverage task and allows to re-observe the most uncertain areas. For the sake of completeness, the green lines delimit the inspection area, while in yellow the generated view planning tree is reported.

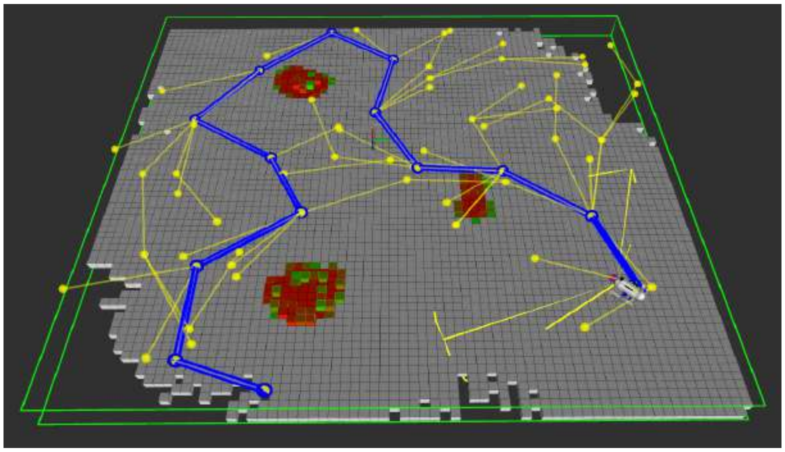


Figure 8.4: The snapshot highlights the effect of the proposed gain function. Although among the generated possible paths, shown in yellow, there are a few that can briefly guide the vehicle to complete the coverage task, a solution that also considers the map uncertainty is selected (depicted in blue).

8.4 Contribution

The solution presented in this Chapter represents the junction between the ATR and RHCA methodologies investigated during the Ph.D. period. The OctoMap-based mapping system was extended for considering the ATR findings and thus, creating a semantic model of the environment. To this end, the standard OctoMap node was modified to include the probability of containing an OPI. The reconstructed semantically annotated map can be used online by a robot to achieve complex tasks. For what this research is concerned, the probabilistic semantic map was utilized for realizing an object uncertainty-aware view planning algorithm for coverage surveys. In particular, the knowledge about the OPIs was used to design a gain function to promote branches that steer the robot toward unexplored regions and simultaneously allow to correctly localize the targets. In fact, the proposed gain function makes the view planning algorithm select branches that reduce the map uncertainty due to both the unexplored areas and OPIs localization. That is, it enables the robot to achieve the goal of reconstructing a semantic map of the environment, i.e., covering the inspected area and well-determining regions that are most likely to contain the targets of interest.

Realistic simulations validated the proposed solution. The results show that the planning solution allowed the AUV to both cover with the FLS the 90% and the 95% of the inspection area, and recognize and localize the OPIs.

In conclusion, by fusing the proposed ATR methodology and the RHCA strategy, a solution for conducting autonomous coverage surveys and simultaneously looking for targets of interest was developed. Therefore, it endows a robot with the ability to meaningfully perceive and model the surroundings and autonomously conduct the assigned task.

Table 8.2: Simulation results

Test 1		
Coverage level 90%	*Path length [m]*	93.37
Coverage level 95%	*Path length [m]*	138.92
OPI 1	*Centroid*	(9.64, -5.42)
	Distance [m]	(2.11, 1.67)
OPI 2	*Centroid*	(-9.52, -2.79)
	Distance [m]	(1.74, 2.04)
OPI 3	*Centroid*	(-2.54, 5.29)
	Distance [m]	(1.29, 1.54)

(a) Outcomes of Test 1

Test 2		
Coverage level 90%	*Path length [m]*	119.59
Coverage level 95%	*Path length [m]*	147.62
OPI 1	*Centroid*	(10.03, -5.36)
	Distance [m]	(1.75, 1.89)
OPI 2	*Centroid*	(-9.95, -6.58)
	Distance [m]	(1.30, 1.33)
OPI 3	*Centroid*	(-1.54, 6.31)
	Distance [m]	(1.79, 1.56)

(b) Outcomes of Test 2

Test 3		
Coverage level 90%	*Path length [m]*	132.47
Coverage level 95%	*Path length [m]*	141.81
OPI 1	*Centroid*	(10.65, -4.99)
	Distance [m]	(1.58, 1.79)
OPI 2	*Centroid*	(-9.98, -3.60)
	Distance [m]	(2.23, 2.15)
OPI 3	*Centroid*	(-2.59, 5.59)
	Distance [m]	(1.34, 1.65)

(c) Outcomes of Test 3

Test 4		
Coverage level 90%	*Path length [m]*	102.18
Coverage level 95%	*Path length [m]*	112.73
OPI 1	*Centroid*	(7.98, -5.91)
	Distance [m]	(1.26, 1.34)
OPI 2	*Centroid*	(-10.28, -4.04)
	Distance [m]	(1.03, 1.79)
OPI 3	*Centroid*	(-1.62, 5.36)
	Distance [m]	(1.63, 2.39)

(d) Outcomes of Test 4

Chapter 9

Conclusions

This thesis reports the results carried out during the Ph.D. period in the years 2018-2021. The work represents the first effort of the MDM Laboratory of UNIFI DIEF to develop a fully autonomous vehicle for underwater operations. A fully autonomous vehicle requires several complex systems acting together for tackling the challenging scenarios it must face. The focus of this work has shifted on two aspects that were identified as fundamental importance as they represent the foundations of complex fully autonomous systems: meaningfully perceiving and modelling the environment, and autonomously planning and re-planning the mission for carrying the assigned tasks.

To make a mobile robot able to gather knowledge of the surroundings, an ATR strategy for onboard applications was developed. As the analysis of state of the art highlighted the supremacy of CNN architectures over other approaches, the ATR strategy, detailed in Chapter 4, was based on CNN image-based detectors. Considering the requirements of onboard online applications in terms of speed and accuracy, usually imposed by the limited hardware capabilities available on compact vehicles, such as FeelHippo AUV, selected as the testing platform, state-of-the-art architectures were selected for the ATR framework. As far as this work is concerned, the ATR methodology was utilized for detecting and localizing OPIs in FLS images onboard in real-time. In particular, various objects of comparable size but different shapes were considered, and the ATR solution was used for determining the presence and the position of the objects, and not for classifying the different object types.

After the training phase, the selected networks were first evaluated offline to assess the performance, intended as the networks precision and inference speed. Then, the ATR solution was validated with real tests at sea. In particular, during the experimental campaign, an OPI was placed in a known position to evaluate the localization accuracy. Then, FeelHippo AUV was utilized to find possible targets in an area of interest. The results achieved during the experimental campaign have highlighted the capability of the proposed architecture to autonomously inspect an unknown underwater scenario by effectively detect and localize targets of interest.

Several efforts have been made during the Ph.D. period for enabling an AUV to autonomously conduct FLS seabed inspections. A sensor-driven framework based on a receding horizon approach was realized. The framework, described in Chapter 5, is composed of a Mapping module, a Motion planner, and a High-level planner. Regarding the former module, a probabilistic 3D occupancy mapping system for FLS based reconstructions had already been realized in previous works ([9]). The Motion planner was used to generate the Dubins path that connected the AUV position to the target configuration. The main focus has shifted to view planning algorithms to be used as the High-level planner, in charge of computing and updating the survey for achieving the task. Firstly, an RRT-based solution was developed and tested in realistic simulations and real trials at sea. An analysis and a benchmark of information gain metrics for AUV coverage tasks were provided. Then, advancements for enhancing the performance of the coverage solution has been proposed in Chapter 6. In detail, the RRT algorithm has been redesigned to speed up the procedure. In addition, a novel methodology based on the KDE technique for estimating the density function of the discovered map was proposed. Such a technique was used to develop an innovative informed expansion strategy for steering the tree toward the unexplored regions. It represents an alternative to the standard Voronoi-biased expansion approach. As the results of the realistic simulations and the conducted experimental campaign have shown, the proposed advances enhanced the performance of the RHCA framework by reducing the travelled path length required to achieve the target coverage.

Moreover, a novel view planning approach was investigated during the Ph.D. period. A RMPC algorithm for FLS autonomous seabed inspections was presented in Chapter 7. To this end, a sampling strategy for randomly generating possible inspection paths that well explore the workspace was developed. The KDE solution for estimating the density of the discovered map has been used for creating a fast methodology for evaluating the generated paths. The RMPC algorithm was employed as the High-level planner in the RHCA framework, instead of the RRT-based solution, and it was validated with simulations performed by means of the UUV Simulator. It will be tested at field as soon as possible.

It is worth highlighting that all developed inspection algorithms do not rely on any assumption regarding the monitored area shape and sea bottom profile, and thus, they can potentially be used in any scenario, and the results presented in this thesis have general validity.

Finally, in Chapter 8 to fulfil the requirements of the topics identified as goals of this thesis, the proposed ATR methodology and the RHCA strategy were combined to realize a target-aware planning solution for autonomously inspecting an area of interest. The ATR findings were employed to create and update a semantic map of the environment. The probability of containing an OPI was added to the standard occupancy mapping strategy. This additional information was used by the RRT view planner to generate paths that cover the area of interest and simultaneously reduces the target localization uncertainty. Realistic simulations validated the proposed solution, and the results showed that it allowed the AUV to both covers the inspection area and recognized and localized the OPIs.

In conclusion, this thesis presented the development of strategies for making FeelHippo AUV able to meaningfully perceive and model the surroundings and autonomously conduct inspections surveys, paving the way for the development of fully autonomous AUVs.

Nonetheless, progress still needs to be made. First of all, the proposed target-aware view planning solution for coverage surveys shall be tested in a real experimental campaign conducted in a controlled environment, i.e., with the OPIs placed in known positions (in order to have a good ground truth). In addition, to create a multi-labeled semantic map of the environment, the task of classifying different objects could be tackled. Then, a methodology for accelerating the nodes' visibility computation could be investigated. In fact, it is the most computationally expensive process of view planning algorithms, which limits the number of generated viewpoints. By generating more viewpoints, the algorithm could achieve better results. Future trends could also involve the extension of the combined ATR and coverage methodologies to other exteroceptive sensors, such as SSSs and optical cameras. Besides, a multi-sensor strategy could be deepened. Finally, a multi-vehicle solution for autonomously inspecting and modeling the underwater environment could represent a coherent continuation of the research activity carried out so far.

References

[1] M. Prats, D. Ribas, N. Palomeras, J. C. García, V. Nannen, S. Wirth, J. J. Fernández, J. P. Beltrán, R. Campos, P. Ridao, *et al.*, "Reconfigurable AUV for intervention missions: a case study on underwater object recovery," *Intelligent Service Robotics*, vol. 5, no. 1, pp. 19–31, 2012.

[2] G. Ferri, F. Ferreira, and V. Djapic, "Multi-domain robotics competitions: The CMRE experience from SAUC-E to the European Robotics League Emergency Robots," in *OCEANS 2017-Aberdeen*, pp. 1–7, IEEE, 2017.

[3] M. Cashmore, M. Fox, T. Larkworthy, D. Long, and D. Magazzeni, "AUV mission control via temporal planning," in *2014 IEEE international conference on robotics and automation (ICRA)*, pp. 6535–6541, IEEE, 2014.

[4] D. Youakim, P. Cieslak, A. Dornbush, A. Palomer, P. Ridao, and M. Likhachev, "Multirepresentation, Multiheuristic A* search-based motion planning for a free-floating underwater vehicle-manipulator system in unknown environment," *Journal of Field Robotics*, vol. 37, no. 6, pp. 925–950, 2020.

[5] E. Vidal, N. Palomeras, K. Istenič, N. Gracias, and M. Carreras, "Multisensor online 3D view planning for autonomous underwater exploration," *Journal of Field Robotics*, 2020.

[6] L. Paull, S. Saeedi, M. Seto, and H. Li, "Sensor-driven online coverage planning for autonomous underwater vehicles," *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 6, pp. 1827–1838, 2012.

[7] L. Zacchini, A. Ridolfi, and B. Allotta, "Receding-horizon sampling-based sensor-driven coverage planning strategy for AUV seabed inspections," in *2020 IEEE/OES Autonomous Underwater Vehicles Symposium (AUV)*, pp. 1–6, IEEE.

[8] È. Pairet, J. D. Hernández, Y. Petillot, and M. Lahijanian, "Online mapping and motion planning under uncertainty for safe navigation in unknown environments," *arXiv preprint arXiv:2004.12317*, 2020.

[9] M. Franchi, A. Bucci, L. Zacchini, E. Topini, A. Ridolfi, and B. Allotta, "A Probabilistic 3D Map Representation for Forward-Looking SONAR Reconstructions," in *2020 IEEE/OES Autonomous Underwater Vehicle Workshop (AUV))*, pp. 1–8, IEEE, 2020.

[10] E. Simetti, R. Campos, D. Di Vito, J. Quintana, G. Antonelli, R. Garcia, and A. Turetta, "Sea Mining Exploration with an UVMS: Experimental Validation of the Control and Perception Framework," *IEEE/ASME Transactions on Mechatronics*, 2020.

[11] A. Ridolfi, N. Secciani, M. Stroobant, M. Franchi, L. Zacchini, R. Costanzi, G. Peralta, and L. E. Cipriani, "Marine Robotics for Recurrent Morphological Investigations of Micro-Tidal Marine-Coastal Environments. A Point of View," *Journal of Marine Science and Engineering*, vol. 9, no. 10, p. 1111, 2021.

[12] R. B. Wynn, V. A. Huvenne, T. P. Le Bas, B. J. Murton, D. P. Connelly, B. J. Bett, H. A. Ruhl, K. J. Morris, J. Peakall, D. R. Parsons, *et al.*, "Autonomous Underwater Vehicles (AUVs): Their past, present and future contributions to the advancement of marine geoscience," *Marine Geology*, vol. 352, pp. 451–468, 2014.

[13] C. Larroque, B. M. de Lépinay, and S. Migeon, "Morphotectonic and fault–earthquake relationships along the northern Ligurian margin (western Mediterranean) based on high resolution, multibeam bathymetry and multichannel seismic-reflection profiles," *Marine Geophysical Research*, vol. 32, no. 1-2, pp. 163–179, 2011.

[14] J. Escartín, C. Mevel, S. Petersen, D. Bonnemains, M. Cannat, M. Andreani, N. Augustin, A. Bezos, V. Chavagnac, Y. Choi, *et al.*, "Tectonic structure, evolution, and the nature of oceanic core complexes and their

detachment fault zones (13 20 N and 13 30 N, Mid Atlantic Ridge)," *Geochemistry, Geophysics, Geosystems*, vol. 18, no. 4, pp. 1451–1482, 2017.

[15] G. A. Mitchell, D. L. Orange, J. J. Gharib, and P. Kennedy, "Improved detection and mapping of deepwater hydrocarbon seeps: optimizing multibeam echosounder seafloor backscatter acquisition and processing techniques," *Marine Geophysical Research*, vol. 39, no. 1, pp. 323–347, 2018.

[16] D. W. Caress, H. Thomas, W. J. Kirkwood, R. McEwen, R. Henthorn, D. A. Clague, C. K. Paull, J. Paduan, K. L. Maier, J. Reynolds, *et al.*, "High-resolution multibeam, sidescan, and subbottom surveys using the MBARI AUV D. Allan B," *Marine habitat mapping technology for Alaska*, pp. 47–69, 2008.

[17] Y.-D. E. Lee, A. Robert, *et al.*, "High-resolution geological AUV survey results across a portion of the eastern Sigsbee Escarpment," *AAPG bulletin*, vol. 88, no. 6, pp. 747–764, 2004.

[18] J. A. Turner, R. C. Babcock, R. Hovey, and G. A. Kendrick, "AUV-based classification of benthic communities of the Ningaloo shelf and mesophotic areas," *Coral Reefs*, vol. 37, no. 3, pp. 763–778, 2018.

[19] K. Schrottke, M. Becker, A. Bartholomä, B. W. Flemming, and D. Hebbeln, "Fluid mud dynamics in the Weser estuary turbidity zone tracked by high-resolution side-scan sonar and parametric sub-bottom profiler," *Geo-Marine Letters*, vol. 26, no. 3, pp. 185–198, 2006.

[20] M. A. Moline, D. L. Woodruff, and N. R. Evans, "Optical delineation of benthic habitat using an autonomous underwater vehicle," *Journal of Field Robotics*, vol. 24, no. 6, pp. 461–471, 2007.

[21] D. Karimanzira, M. Jacobi, T. Pfützenreuter, T. Rauschenbach, M. Eichhorn, R. Taubert, and C. Ament, "First testing of an AUV mission planning and guidance system for water quality monitoring and fish behavior observation in net cage fish farming," *Information Processing in Agriculture*, vol. 1, no. 2, pp. 131–140, 2014.

[22] A. Bucci, L. Zacchini, M. Franchi, A. Ridolfi, and B. Allotta, "Comparison of feature detection and outlier removal strategies in a mono

visual odometry algorithm for underwater navigation," *Applied Ocean Research*, vol. 118, p. 102961, 2022.

[23] D. Terracciano, L. Bazzarello, A. Caiti, R. Costanzi, and V. Manzari, "Marine Robots for Underwater Surveillance," *Current Robotics Reports*, pp. 1–9, 2020.

[24] L. Jin, H. Liang, and C. Yang, "Accurate Underwater ATR in Forward-Looking Sonar Imagery Using Deep Convolutional Neural Networks," *IEEE Access*, vol. 7, pp. 125522–125531, 2019.

[25] L. Zacchini, A. Ridolfi, A. Topini, N. Secciani, A. Bucci, E. Topini, and B. Allotta, "Deep Learning for on-board AUV Automatic Target Recognition for Optical and Acoustic imagery," in *IFAC-V 2020, Germany*, IFAC, 2020.

[26] L. Zacchini, M. Franchi, V. Manzari, M. Pagliai, N. Secciani, A. Topini, M. Stifani, and A. Ridolfi, "Forward-Looking Sonar CNN-based Automatic Target Recognition: an experimental campaign with FeelHippo AUV," in *2020 IEEE/OES Autonomous Underwater Vehicles Symposium (AUV)*, pp. 1–6, IEEE, 2020.

[27] B. Allotta, R. Costanzi, A. Ridolfi, M. Reggiannini, M. Tampucci, and D. Scaradozzi, "Archaeology oriented optical acquisitions through MARTA AUV during ARROWS European project demonstration," in *OCEANS 2016 MTS/IEEE Monterey*, pp. 1–4, IEEE, 2016.

[28] M. Franchi, A. Ridolfi, and L. Zacchini, "A forward-looking sonar-based system for underwater mosaicing and acoustic odometry," in *2018 IEEE/OES Autonomous Underwater Vehicle Workshop (AUV)*, pp. 1–6, IEEE, 2018.

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[30] M. Valdenegro-Toro, "Object recognition in forward-looking sonar images with Convolutional Neural Networks," in *OCEANS 2016 MTS/IEEE Monterey*, pp. 1–6, IEEE, 2016.

[31] B. Allotta, L. Pugi, F. Bartolini, R. Constanzi, N. Monni, J. Gelli, G. Vettori, L. Gualdesi, M. Natalini, and A. Ridolfi, “The THESAURUS project, a long range AUV for extended exploration, surveillance and monitoring of archaeological sites,” in *MARINE V: proceedings of the V International Conference on Computational Methods in Marine Engineering*, pp. 760–771, CIMNE, 2013.

[32] B. Allotta, R. Costanzi, J. Gelli, L. Pugi, and A. Ridolfi, “Design of a modular propulsion system for MARTA AUV,” in *OCEANS 2015-Genova*, pp. 1–7, IEEE, 2015.

[33] G. Ferri, F. Ferreira, and V. Djapic, “Boosting the talent of new generations of marine engineers through robotics competitions in realistic environments: The SAUC-E and euRathlon experience,” in *OCEANS 2015-Genova*, pp. 1–6, IEEE, 2015.

[34] G. Ferri, F. Ferreira, and V. Djapic, “Multi-domain robotics competitions: The CMRE experience from SAUC-E to the European Robotics League Emergency Robots,” in *OCEANS 2017-Aberdeen*, pp. 1–7, IEEE, 2017.

[35] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, “ROS: an open-source Robot Operating System,” in *ICRA workshop on open source software*, vol. 3, p. 5, Kobe, Japan, 2009.

[36] V. Myers and J. Fawcett, “A template matching procedure for automatic target recognition in synthetic aperture sonar imagery,” *IEEE Signal Processing Letters*, vol. 17, no. 7, pp. 683–686, 2010.

[37] Y. Petillot, Y. Pailhas, J. Sawas, N. Valeyrie, and J. Bell, “Target recognition in synthetic aperture and high resolution side-scan sonar,” in *Proc. European Conference on Underwater Acoustics*, pp. 99–106, 2010.

[38] M. Valdenegro-Toro, “Best practices in convolutional networks for forward-looking sonar image recognition,” in *OCEANS 2017-Aberdeen*, pp. 1–9, IEEE, 2017.

[39] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: an astounding baseline for recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 806–813, 2014.

[40] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

[41] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263–7271, 2017.

[42] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[43] I. Kvasić, N. Mišković, and Z. Vukić, "Convolutional neural network architectures for sonar-based diver detection and tracking," in *OCEANS 2019-Marseille*, pp. 1–6, IEEE, 2019.

[44] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*, pp. 21–37, Springer, 2016.

[45] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[46] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

[47] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[48] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, *et al.*, "Speed/accuracy trade-offs for modern convolutional object detectors," in *Proceedings*

of the IEEE conference on computer vision and pattern recognition, pp. 7310–7311, 2017.

[49] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, pp. 91–99, 2015.

[50] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proceedings of the IEEE international conference on computer vision*, pp. 2961–2969, 2017.

[51] X. Li, M. Shang, H. Qin, and L. Chen, "Fast accurate fish detection and recognition of underwater images with Fast R-CNN," in *OCEANS 2015-MTS/IEEE Washington*, pp. 1–5, IEEE, 2015.

[52] Z. L. Cao, Y. Huang, and E. L. Hall, "Region filling operations with random obstacle avoidance for mobile robots," *Journal of Robotic systems*, vol. 5, no. 2, pp. 87–102, 1988.

[53] H. Choset, "Coverage for robotics–a survey of recent results," *Annals of mathematics and artificial intelligence*, vol. 31, no. 1-4, pp. 113–126, 2001.

[54] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous systems*, vol. 61, no. 12, pp. 1258–1276, 2013.

[55] D. P. Williams, F. Baralli, M. Micheli, and S. Vasoli, "Adaptive underwater sonar surveys in the presence of strong currents," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2604–2611, IEEE, 2016.

[56] E. Galceran, R. Campos, N. Palomeras, M. Carreras, and P. Ridao, "Coverage path planning with realtime replanning for inspection of 3D underwater structures," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6586–6591, IEEE, 2014.

[57] N. Palomeras, N. Hurtós, M. Carreras, and P. Ridao, "Autonomous mapping of underwater 3-D structures: From view planning to execution," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1965–1971, 2018.

[58] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011.

[59] M. Trummer, C. Munkelt, and J. Denzler, "Online next-best-view planning for accuracy optimization using an extended e-criterion," in *2010 20th International Conference on Pattern Recognition*, pp. 1642–1645, IEEE, 2010.

[60] S. Kriegel, C. Rink, T. Bodenmüller, and M. Suppa, "Efficient next-best-scan planning for autonomous 3D surface reconstruction of unknown objects," *Journal of Real-Time Image Processing*, vol. 10, no. 4, pp. 611–631, 2015.

[61] J. I. Vasquez-Gomez, L. E. Sucar, and R. Murrieta-Cid, "View planning for 3D object reconstruction with a mobile manipulator robot," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4227–4233, IEEE, 2014.

[62] J. Delmerico, S. Isler, R. Sabzevari, and D. Scaramuzza, "A comparison of volumetric information gain metrics for active 3D object reconstruction," *Autonomous Robots*, vol. 42, no. 2, pp. 197–208, 2018.

[63] S. Isler, R. Sabzevari, J. Delmerico, and D. Scaramuzza, "An information gain formulation for active volumetric 3D reconstruction," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3477–3484, IEEE, 2016.

[64] C. Papachristos, S. Khattak, and K. Alexis, "Uncertainty-aware receding horizon exploration and mapping using aerial robots," in *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 4568–4575, IEEE, 2017.

[65] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon path planning for 3D exploration and surface inspection," *Autonomous Robots*, vol. 42, no. 2, pp. 291–306, 2018.

[66] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, 2013.

[67] K. Liu, Z. Fan, M. Liu, and S. Zhang, "Object-aware Semantic Mapping of Indoor Scenes using Octomap," in *2019 Chinese Control Conference (CCC)*, pp. 8671–8676, IEEE, 2019.

[68] T. Dang, C. Papachristos, and K. Alexis, "Autonomous exploration and simultaneous object search using aerial robots," in *2018 IEEE Aerospace Conference*, pp. 1–7, IEEE, 2018.

[69] R. Ashour, T. Taha, J. M. M. Dias, L. Seneviratne, and N. Almoosa, "Exploration for Object Mapping Guided by Environmental Semantics using UAVs," *Remote Sensing*, vol. 12, no. 5, p. 891, 2020.

[70] G. Costante, C. Forster, J. Delmerico, P. Valigi, and D. Scaramuzza, "Perception-aware path planning," *arXiv preprint arXiv:1605.04151*, 2016.

[71] Z. Zhang and D. Scaramuzza, "Beyond point clouds: Fisher information field for active visual localization," in *2019 International Conference on Robotics and Automation (ICRA)*, pp. 5986–5992, IEEE, 2019.

[72] M. Massot-Campos and G. Oliver-Codina, "Optical sensors and methods for underwater 3D reconstruction," *Sensors*, vol. 15, no. 12, pp. 31525–31557, 2015.

[73] M. Johnson-Roberson, M. Bryson, A. Friedman, O. Pizarro, G. Troni, P. Ozog, and J. C. Henderson, "High-resolution underwater robotic vision-based mapping and three-dimensional reconstruction for archaeology," *Journal of Field Robotics*, vol. 34, no. 4, pp. 625–643, 2017.

[74] A. Palomer, P. Ridao, and D. Ribas, "Inspection of an underwater structure using point-cloud SLAM with an AUV and a laser scanner," *Journal of Field Robotics*, vol. 36, no. 8, pp. 1333–1344, 2019.

[75] E. Hernandez, P. Ridao, A. Mallios, and M. Carreras, "Occupancy grid mapping in an underwater structured environment," *IFAC Proceedings Volumes*, vol. 42, no. 18, pp. 286–291, 2009.

[76] Y. Wang, Y. Ji, H. Woo, Y. Tamura, A. Yamashita, and A. Hajime, "3D occupancy mapping framework based on acoustic camera in underwater environment," *IFAC-PapersOnLine*, vol. 51, no. 22, pp. 324–330, 2018.

[77] J. D. Hernández, E. Vidal, M. Moll, N. Palomeras, M. Carreras, and L. E. Kavraki, "Online motion planning for unexplored underwater environments using autonomous underwater vehicles," *Journal of Field Robotics*, vol. 36, no. 2, pp. 370–396, 2019.

[78] P. V. Teixeira, M. Kaess, F. S. Hover, and J. J. Leonard, "Underwater inspection using SONAR-based volumetric submaps," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4288–4295, IEEE, 2016.

[79] P. Sodhi, B.-J. Ho, and M. Kaess, "Online and consistent occupancy grid mapping for planning in unknown environments," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7879–7886, IEEE, 2019.

[80] B.-J. Ho, P. Sodhi, P. Teixeira, M. Hsiao, T. Kusnur, and M. Kaess, "Virtual occupancy grid map for submap-based pose graph SLAM and planning in 3D environments," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2175–2182, IEEE, 2018.

[81] G. R. Terrell and D. W. Scott, "Variable kernel density estimation," *The Annals of Statistics*, pp. 1236–1265, 1992.

[82] C. Yen-Chi, "Lecture 6: Density estimation: Histogram and kernel density estimator," 2018.

[83] T. I. Fossen *et al.*, *Guidance and control of ocean vehicles.* John Wiley & Sons, 1994.

[84] R. M. Rogers, *Applied mathematics in integrated navigation systems.* American Institute of Aeronautics and Astronautics, 2007.

[85] L. Zacchini, V. Calabrò, M. Candeloro, F. Fanelli, A. Ridolfi, and F. Dukan, "Novel Noncontinuous Carouseling Approaches for MEMS-Based North Seeking Using Kalman Filter: Theory, Simulations, and Preliminary Experimental Evaluation," *IEEE/ASME Transactions on Mechatronics*, vol. 25, no. 5, pp. 2437–2448, 2020.

[86] R. Costanzi, F. Fanelli, N. Monni, A. Ridolfi, and B. Allotta, "An attitude estimation algorithm for mobile robots under unknown magnetic

disturbances," *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 4, pp. 1900–1911, 2016.

[87] B. Allotta, A. Caiti, R. Costanzi, F. Fanelli, D. Fenucci, E. Meli, and A. Ridolfi, "A new AUV navigation system exploiting Unscented Kalman Filter," *Ocean Engineering*, vol. 113, pp. 121–132, 2016.

[88] M. Franchi, A. Ridolfi, and B. Allotta, "Underwater navigation with 2D forward looking SONAR: An adaptive unscented Kalman filter-based strategy for AUVs," *Journal of Field Robotics*, 2020.

[89] S. Negahdaripour, "On 3-D motion estimation from feature tracks in 2-D FS sonar video," *IEEE Transactions on Robotics*, vol. 29, no. 4, pp. 1016–1030, 2013.

[90] N. Hurtós, D. Ribas, X. Cufí, Y. Petillot, and J. Salvi, "Fourier-based Registration for Robust Forward-Looking SONAR Mosaicing in Low-visibility Underwater Environments," *Journal of Field Robotics*, vol. 32, no. 1, pp. 123–151, 2015.

[91] N. Hurtós Vilarnau *et al.*, "Forward-looking sonar mosaicing for underwater environments," 2014.

[92] F. Ferreira, V. Djapic, M. Micheli, and M. Caccia, "Forward looking sonar mosaicing for mine countermeasures," *Annual Reviews in Control*, vol. 40, pp. 212–226, 2015.

[93] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[94] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning.* MIT Press, 2016. http://www.deeplearningbook.org.

[95] M. Valdenegro-Toro, "Deep neural networks for marine debris detection in SONAR images," *arXiv preprint arXiv:1905.05241*, 2019.

[96] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[97] T. Tieleman and G. Hinton, "Neural networks for machine learning," *Coursera (Lecture 65-RMSprop)*, 2012.

[98] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[99] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*, pp. 818–833, Springer, 2014.

[100] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*, pp. 740–755, Springer, 2014.

[101] R. Padilla, S. L. Netto, and E. A. da Silva, "A survey on performance metrics for object-detection algorithms," in *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, pp. 237–242, IEEE, 2020.

[102] W. Burgard, D. Fox, and S. Thrun, "Probabilistic robotics," *The MIT Press*, 2005.

[103] H. P. Moravec, "Sensor fusion in certainty grids for mobile robots," in *Sensor devices and systems for robotics*, pp. 253–276, Springer, 1989.

[104] G. Antonelli, *Underwater robots*, vol. 3. Springer, 2014.

[105] Y. Li, Z. Littlefield, and K. E. Bekris, "Asymptotically optimal sampling-based kinodynamic planning," *The International Journal of Robotics Research*, vol. 35, no. 5, pp. 528–564, 2016.

[106] L. E. Dubins, "On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents," *American Journal of mathematics*, vol. 79, no. 3, pp. 497–516, 1957.

[107] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object Detection via Region-based Fully Convolutional Networks," *arXiv preprint arXiv:1605.06409*, 2016.

[108] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[109] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, *et al.*, “Tensorflow: A system for large-scale machine learning,” in *12th USENIX symposium on operating systems design and implementation (OSDI) 16*, pp. 265–283, 2016.

[110] NVIDIA, “NVIDIA Jetson Nano.” https://developer.nvidia.com/embedded/jetson-nano-developer-kit (last visit October 2021), 2018.

[111] L. Zacchini, M. Franchi, and A. Ridolfi, “Sensor-driven autonomous underwater inspections: A receding-horizon RRT-based view planning solution for AUVs,” *Journal of Field Robotics*, 2022.

[112] B. Allotta, A. Caiti, L. Chisci, R. Costanzi, F. Di Corato, C. Fantacci, D. Fenucci, E. Meli, and A. Ridolfi, “An unscented Kalman filter based navigation algorithm for autonomous underwater vehicles,” *Mechatronics*, vol. 39, pp. 185–195, 2016.

[113] I. A. Şucan, M. Moll, and L. E. Kavraki, “The Open Motion Planning Library,” *IEEE Robotics & Automation Magazine*, vol. 19, pp. 72–82, December 2012. https://ompl.kavrakilab.org (last visit October 2021).

[114] A. Törn and A. Žilinskas, *Global optimization*, vol. 350. Springer, 1989.

[115] F. J. Solis and R. J.-B. Wets, “Minimization by random search techniques,” *Mathematics of operations research*, vol. 6, no. 1, pp. 19–30, 1981.

[116] Z. B. Zabinsky, *Stochastic adaptive search for global optimization*, vol. 72. Springer Science & Business Media, 2013.

[117] S. M. LaValle and J. J. Kuffner Jr, “Randomized kinodynamic planning,” *The international journal of robotics research*, vol. 20, no. 5, pp. 378–400, 2001.

[118] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, “Informed RRT*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic,” in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2997–3004, IEEE, 2014.

[119] B. A. Turlach, “Bandwidth selection in kernel density estimation: A review,” in *CORE and Institut de Statistique*, Citeseer, 1993.

[120] M. T. Wells, G. Casella, and C. P. Robert, “Generalized accept-reject sampling schemes,” in *A Festschrift for Herman Rubin*, pp. 342–347, Institute of Mathematical Statistics, 2004.

[121] R. R. Curtin, M. Edel, M. Lozhnikov, Y. Mentekidis, S. Ghaisas, and S. Zhang, “mlpack 3: a fast, flexible machine learning library,” *Journal of Open Source Software*, vol. 3, no. 26, p. 726, 2018.

[122] J. L. Piovesan and H. G. Tanner, “Randomized model predictive control for robot navigation,” in *2009 IEEE International Conference on Robotics and Automation*, pp. 94–99, 2009.

[123] L. Zacchini, M. Franchi, A. Bucci, N. Secciani, and A. Ridolfi, “Randomized MPC for view planning in AUV seabed inspections,” in *OCEANS 2021: San Diego–Porto*, pp. 1–6, IEEE, 2021.

[124] B.-O. H. Eriksen, M. Breivik, E. F. Wilthil, A. L. Flåten, and E. F. Brekke, “The branching-course model predictive control algorithm for maritime collision avoidance,” *Journal of Field Robotics*, vol. 36, no. 7, pp. 1222–1249, 2019.

[125] M. Cavazzuti, *Optimization Methods.* Springer, Berlin, Heidelberg, 2013.