FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Feature-based underwater localization using an imaging sonar

António José Ventura de Oliveira

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Bruno M. Ferreira PhD

Supervisor: Nuno A. Cruz PhD

July 29, 2020

Resumo

A capacidade de um veículo autónomo submarino (AUV) se localizar no ambiente, bem como de extrair características ambientais de relevo, é de grande importância para o sucesso da navegação. No entanto, esta tarefa é particularmente desafiante em ambientes subaquáticos devido à rápida atenuação sofrida pelos sinais de sistemas de posicionamento global ou outros sinais de radiofrequência.

Os sonares são um dos dispositivos de sensorização mais comuns para localização e mapeamento subaquático, sendo usados para detectar e identificar características estruturais subaquáticas através da aquisição de imagens acústicas. No entanto, apesar de ser uma tecnologia robusta, ainda existem vários problemas que afetam o seu desempenho, como a presença de ruído de fundo e de múltiplos ecos nas imagens adquiridas, o que aumenta o potencial de ocorrência de falsas deteções.

Este estudo explora o processamento e a análise de imagens acústicas, por meio dos dados adquiridos por um sonar de varrimento mecânico (MSIS), a fim de extrair características ambientais relevantes que possibilitem a estimação da localização do veículo. Para esse fim, foram avaliados os desempenhos de diferentes algoritmos de extração de pontos de interesse de última geração. Além disso, é proposta uma melhoria na etapa de correspondência de pontos de interesse, a fim de adaptar este procedimento às propriedades das imagens acústicas. Os pontos de interesse extraídos são utilizados para alimentar um estimador de localização composto por um algoritmo de Localização e Mapeamento Simultâneo (SLAM), implementando um Filtro de Kalman Extendido (EKF). Vários testes foram realizados em ambiente estruturado e são apresentados resultados experimentais do processo de extração de pontos de interesse e do processo de localização e mapeamento.

Palavras-Chave: AUV, Correspondência de Pontos de Interesse, EKF, Extração de Pontos de Interesse, Localização, Ponto de Interesse, SLAM, Sonar

Abstract

The ability of an autonomous underwater vehicle (AUV) to locate itself in an environment as well as to detect relevant environmental features is of key importance for navigation success. However, this task is particularly challenging in underwater environments due to the rapid attenuation suffered by global positioning system or other radio-frequency signals.

Sonars are one the most common sensing devices for underwater localization and mapping, being used to detect and identify underwater structural features through the acquisition of acoustic images. Nevertheless, despite being a robust technology, there are still several problems affecting its performance such as the presence of background noise and multiple echoes in the acquired images, which increase the potential for false detections.

This study explores the processing and analysis of acoustic images, through the data acquired by a mechanical scanning imaging sonar (MSIS), in order to extract relevant environmental features that enable location estimation. For this purpose, the performances of different state-of-the art feature extraction algorithms were evaluated. Furthermore, an improvement to the feature matching step is proposed, in order to adapt this procedure to the characteristics of acoustic images. The extracted features are then used to feed a location estimator composed of a Simultaneous Localization and Mapping (SLAM) algorithm implementing an Extended Kalman Filter (EKF). Several tests were performed in a structured environment and experimental results of the feature extraction process and localization and mapping process are presented.

Keywords: AUV, EKF, Feature, Feature Extraction, Feature Matching, Localization, SLAM, Sonar

Acknowledgements

The conclusion of this dissertation marks the end of a significant journey in my life, since it culminates five years of studies and hard work. At the same time, it marks the beginning of a new one. I reckon that all of this would not be possible without the people mentioned hereafter.

First of all, I would like to express my sincere gratitude to both my supervisors, INESC senior researchers Bruno Ferreira and Nuno Cruz for all the mentoring, guidance and learning moments in a journey that started long before the development of this dissertation.

To my parents and family, I can not put into words how grateful I am for all the unconditional support, help and comprehension throughout my entire life.

Lastly, but not least, I would like to thank all my friends for all the unforgettable moments that have marked this journey.

António José Ventura de Oliveira

"Why do we fall?
So that we can learn to pick ourselves back up."

Sir Michael Caine

Contents

1 Introduction **1**
- 1.1 Context . . . 1
  - 1.1.1 Autonomous Underwater Vehicles . . . 1
- 1.2 Motivation . . . 2
- 1.3 Objectives . . . 3
- 1.4 Contributions . . . 3
- 1.5 Document Structure . . . 4

2 State of the Art **5**
- 2.1 Underwater Navigation . . . 5
  - 2.1.1 Methodologies . . . 5
  - 2.1.2 Sensing . . . 8
  - 2.1.3 Environment Representation . . . 11
- 2.2 Feature Extraction Algorithms . . . 15
  - 2.2.1 SURF - Speeded UP Robust Features . . . 16
  - 2.2.2 BRIEF - Binary Robust Independent Elementary Features . . . 16
  - 2.2.3 Harris Corner Detector . . . 17
  - 2.2.4 FAST - Features from Accelerated Segment Test . . . 17
  - 2.2.5 BRISK - Binary Robust Invariant Scalable Keypoints . . . 18
  - 2.2.6 ORB - Oriented FAST and Rotated BRIEF . . . 19
  - 2.2.7 Feature Matching . . . 20
- 2.3 Summary . . . 21

3 System Overview **23**
- 3.1 Tools and Methods . . . 25

4 Keypoint Detection and Description Analysis **27**
- 4.1 Introduction . . . 27
- 4.2 Feature Extractors Performance Analysis . . . 28
  - 4.2.1 Methodology . . . 28
- 4.3 Results and Discussion . . . 33
- 4.4 Summary . . . 39

5 Feature Matching **41**
- 5.1 Introduction . . . 41
- 5.2 SURF's Spectra Descriptor . . . 41
- 5.3 Proposed Solution . . . 42
- 5.4 Results and Discussion . . . 44

5.5 Summary 48

6 Simultaneous Localization and Mapping **49**
6.1 Introduction 49
6.2 System Integration 50
6.3 State Estimator 52
6.3.1 The Prediction Cycle 52
6.3.2 The Update Cycle 53
6.3.3 Feature Point Clustering 54
6.4 Results and Discussion 57
6.5 Summary 60

7 Results and Performance Analysis **63**
7.1 Introduction 63
7.2 Static Scenario 64
7.3 Dynamic Scenario 66
7.4 Summary 68

8 Conclusions and Future Work **69**
8.1 Conclusions 69
8.2 Future Work 70

List of Figures

2.1 AUV localization techniques categorization. Adapted from Paull et al., 2014 . . . 6
2.2 BRISK sampling pattern presented in Leutenegger et al., 2011 18
2.3 The feature matching procedure 20

3.1 Block diagram representation of the developed solution 23
3.2 Testing environment and corresponding acoustic representation 24

4.1 Acoustic image obtained through mechanical scanning imaging sonar 27
4.2 Image processing procedure for feature extraction comparison 29
4.3 The SHAD AUV and the Water tank structured used for testing procedures 29
4.4 Results of different image composition techniques 30
4.5 Intermediate results of image pre-processing routine applied to image 3 of *dataset*2 32
4.6 Interest point detected in image 3 of dataset 2 with the algorithms under analysis 34
4.7 Total number of detected interest points by each algorithm in each image of *dataset*1 35
4.8 Total number of detected interest points by each algorithm in each image of *dataset*2 35
4.9 Number of false-positives detected amongst the 50 best interest points detected in each image of *dataset*1 36
4.10 Number of false-positives detected amongst the 50 best interest points detected in each image of *dataset*2 36
4.11 Feature points matched in image 3 of *dataset*2 with the algorithms under analysis 38
4.12 Number of matches performed for each image pair of *dataset*2 39
4.13 Number of outliers detected amongst the matched features in each image pair of *dataset*2 39

5.1 Example of the correction introduced by a new filtering layer in the matching process 44
5.2 The final proposed solution procedure 45
5.3 Feature point matching results for image 3 of *dataset*2 46
5.4 Number of outliers detected per image pair of *dataset*2 47
5.5 Outlier ratio per image pair of *dataset*2 48

6.1 Developed solution module sequence 50
6.2 Intermediate results of the final image pre-processing routine applied to image 6 of *dataset*2 50
6.3 Image pre-processing procedure impact on feature detection applied to image 6 of *dataset*2 51
6.4 Feature points detected in image 1 of *dataset*1 54
6.5 Results of the clustering procedure employed to image 1 of *dataset*1 55
6.6 Effect of the motion of the AUV on the clustering procedure 56
6.7 AUV pose estimation evolution throughout *dataset*1 57

6.8 Floater pose estimation evolution throughout *dataset*1 58
6.9 Vehicle (filled blue circle) and landmarks estimated positions and associated uncertainty ellipses (*dataset*1) 58
6.10 AUV pose estimation evolution throughout *dataset*2 59
6.11 Floater pose estimation evolution throughout *dataset*2 59
6.12 Vehicle (filled blue circle) and landmarks estimated positions and associated uncertainty ellipses (*dataset*2) 60

7.1 Scheme of the static scenario 64
7.2 AUV estimated positions during the static scenario 65
7.3 Landmark estimated positions drift for the static scenario 65
7.4 Scheme of the dynamic scenario 66
7.5 AUV estimated positions during the dynamic scenario 66
7.6 Landmark estimated positions drift for the dynamic scenario 67

List of Tables

1.1 Applications for AUVs. Adapted from Sahoo et al., 2019 2

2.1 Common onboard AUV sensors used for localization. Adapted from Paull et al., 2014 9
2.2 Common sensor fusion algorithms 9
2.3 Main sonar imaging and ranging devices for underwater applications. Adapted from Paull et al., 2014 10
2.4 Advantages and disadvantages of common SLAM algorithms 12
2.5 Most common techniques for feature detection and description 15

4.1 Number of matches performed and corresponding number of outliers for *dataset*1 36
4.2 Number of matches performed and corresponding number of outliers for *dataset*2 37
4.3 Computation time measurements results performed for each algorithm for *dataset*1 37
4.4 Computation time measurements results performed for each algorithm for *dataset*2 37

5.1 Number of matches performed and corresponding number of outliers and computation time required for *dataset*1 46
5.2 Number of matches performed and corresponding number of outliers and computation time required for *dataset*2 47

7.1 Number of detected landmarks for the static scenario 64
7.2 Ground-truth measurements and estimated poses for the static scenario 65
7.3 Number of detected landmarks for the dynamic scenario 67
7.4 Ground-truth measurements and estimated poses for the dynamic scenario 67

Abbreviations

AI	Artificial Intelligence
AUV	Autonomous Underwater Vehicle
BRIEF	Binary Robust Independent Elementary Features
BRISK	Binary Robust Invariant Scalable Keypoints
DR	Dead Reckoning
DVL	Doppler Velocity Log
EIF	Extended Information Filter
EKF	Extended Kalman Filter
FAST	Features from Accelerated Segment Test
GIBs	GPS Intelligent Buoys
GPS	Global Positioning System
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
MSIS	Mechanical Scanning Imaging Sonar
ORB	Oriented FAST and Rotated BRIEF
PF	Particle Filter
ROV	Remotely Operated Vehicle
SBL	Short Baseline
SIFT	Scale-Invariant Feature Transform
SLAM	Simultaneous Localization and Mapping
SURF	Speeded UP Robust Features
SUSAN	Smallest Univalue Segment Assimilating Nucleus
TOF	Time Of Flight
YKF	Unscented Kalman Filter
USBL	Ultra-Short Baseline
UUV	Unmanned Underwater Vehicle

Chapter 1

Introduction

1.1 Context

The majority of earth's surface is covered by water, whether in the form of oceans, rivers or lakes. These environment have a clear impact on human life, directly or indirectly. Therefore it is of utmost importance to study them, in order to guarantee their safety against environmental pollution and to allow for exploration of the available natural resources. For this purpose, underwater robotic vehicles are, nowadays, a solution to always take into consideration.

The high risks and operational costs associated to the deployment of manned underwater vehicles have led researches to an increasing set of solutions. In this context, unmanned underwater vehicles (UUVs) are becoming increasingly popular, since they allow for navigation at greater depths and extremely harsh conditions. Among the most common solutions, we shall outline the following:

- **Towed Vehicles** - towed behind ships and boats, such that the vehicle itself is controlled through a combination of on-board hydrodynamic actuators and the drag tether. Commonly used in oceanographic data collection and surveying underwater cables or pipelines;

- **Remotely Operated Vehicles (ROVs)** - remotely controlled by a pilot (on a ship or platform), usually via a tether. This tether is also used for supplying power to the vehicle and can sometimes be used to retrieve said vehicle to the surface. Nevertheless, all these requirements make its use complex and expensive, mainly when larger size vehicles are required;

- **Autonomous Underwater Vehicles (AUVs)** - able to perform operations without any user intervention or control. The pretended mission plan is programmed into the AUV's system, so that it is able to carry out the desired operations.

1.1.1 Autonomous Underwater Vehicles

The development of AUVs came as an improvement to the solution offered by ROVs, since these vehicles require no tether or support ship and, therefore, are able to navigate autonomously, relying

solely on their navigation algorithms and data acquired from sensors. Furthermore, as the authors state in Sahoo et al., 2019, AUVs *"have high manoeuvrability, can travel to remote locations, narrow complex pathways, involve no human fatigue and operation cost is very less"*. Table 1.1 presents the most common applications for this type of vehicles.

Nowadays, AUVs can have up to six degrees of freedom and travel at speeds higher than $20m/s$, while detecting obstacles and mapping their surroundings, as stated by Sahoo et al., 2019. While more sophisticated and affordable, there are still some key aspects that need to be tackled when it comes to sensing, navigation, guidance and control, underwater wireless communication, batteries, manufacturing techniques or computational power.

Category	**Description**
Military	Surveillance
	Anti-submarine warfare
	Mine countermeasures
	Inspection of wreckage
	Payload delivery to ocean floor
	Search and rescue
	Air crash investigation
Scientific	Ocean exploration and bathymetric study
	Marine biology studies
	Geological survey
	Archaeological survey
	Environment monitoring
Industry	Repair and maintenance
	Track and repair of underwater cables
	Underwater structure inspection
Other	Underwater video footage collection
	Fishing
	Entertainment and tourism

Table 1.1: Applications for AUVs. Adapted from Sahoo et al., 2019

1.2 Motivation

In order for an AUV to be able to perform properly and safely, a precise navigation is essential. This objective is closely related to the performance of its localization estimation procedure and capability to represent the surrounding environment.

According to the authors in Paull et al., 2014, AUV localization *"is a challenging problem due primarily to the rapid attenuation of higher frequency signals and the unstructured nature of the undersea environment"*. At surface, most systems rely on radio or spread-spectrum communications and global positioning for localization purposes, which is unfeasible for AUVs since, underwater, these signals can only propagate at short distances. Some localization solutions employ external devices such as acoustic beacons or sensors of higher accuracy. However both come at higher costs. Cameras are also amongst the most common solutions to overcome this problem, offering significantly lower costs than other typical solutions. Yet, these present fairly low area coverage and its performance is heavily dependent on water visibility conditions.

Sonars are amongst the most common sensing devices for underwater navigation, making use of sound waves propagation for mapping purposes. Nevertheless, despite being a fairly mature and robust technology, there are still several problems concerning its application, so studies continue to be carried out in order to improve its performance.

The analysis of acoustic images acquired through a sonar allows the extraction of information that is of great value not only to solve the localization problem but also to map the surroundings of the vehicle. However, the presence of background noise or clutter, multiple echoes and multi-path transmission puts severe challenges to these objectives, mainly due to the high potential for false detections. Moreover, the motion of the AUV will result in the deformation of the acquired features, presenting additional challenges to a correct identification and correspondence of these throughout a series of acoustic images.

Image processing techniques, more precisely, feature extraction methodologies, have the potential to form the basis of a solution that will allow the detection of environmental features in the surroundings of the AUV. However, these procedures have been originally developed for optical image analysis and its applicability to acoustic images requires further study and development. Computation time constraints must also be taken into consideration since, when moving, the vehicle will require quick but also reliable information regarding any obstacle in its vicinity.

1.3 Objectives

The purpose of this dissertation is to develop a system that allows the localization of a vehicle and the mapping of its surroundings in underwater environments. This solution is based on a mechanical scanning imaging sonar (MSIS) data acquisition process.

It is expected that the acquired acoustic images can be analysed through image processing techniques in order to extract valuable environmental feature information. Therefore, a study and evaluation of the applicability of feature extraction algorithms for acoustic image analysis is required.

This information can then be fed to a state estimator, that will enable data fusion of the vehicle's position, as well as of the detected landmarks. Landmark association is intended to be performed based on the comparison of feature descriptors.

1.4 Contributions

At the moment, the development of AUVs is one of the most active areas of scientific research. The work carried out throughout this dissertation brings further contributions regarding the application of MSIS for localization and mapping purposes and the challenges faced. Furthermore, the analysis conducted on the application of common feature extraction algorithms to acoustic image analysis presents valuable conclusions regarding the detected problems and possible required adaptations to usual methods implemented for optical image analysis.

It is also important to note that, despite the main focus of this dissertation being related to AUV navigation, the work developed is not only relevant for this type of vehicles, since it can be extended to others.

1.5 Document Structure

This document is composed of six more chapters, apart from the present one, and corresponding sections.

Chapter 2 is focused on a review of the current state of AUV navigation, from common methodologies to sensor technology employed. Furthermore, an overview of common feature extraction algorithms is also introduced.

Chapter 3 introduces the structure of the developed solution in further detail and the logical reasoning behind it. The work carried out for this purpose is more thoroughly detailed throughout chapters 4, 5 and 6.

In chapter 7 a careful analysis of the results obtained for different scenarios is presented and the solutions performance evaluated.

Lastly, chapter 8 highlights the main conclusions and contributions of this work, as well as some considerations regarding future work.

Chapter 2

State of the Art

This section presents a review on underwater navigation. Common principles, sensors and algorithms are presented, with especial focus on localization techniques. Due to the importance of sonar technology for the developed solution, this sensing device in presented in greater detail. Furthermore, an overview on environment representation techniques is also introduced. Then, an introduction to feature detectors and descriptors is presented, as well as further detail on some of the most common feature extraction algorithms.

2.1 Underwater Navigation

The rapid attenuation of higher frequency signals and the nature of underwater environments puts significant challenges to AUVs navigation. Accuracy in position is crucial for an AUV to navigate. Without external references and due to the AUV's motion, this accuracy deteriorates over time. Therefore, a number of methodologies, sensors and algorithms have been proposed in order to overcome so called challenges and are presented hereafter.

2.1.1 Methodologies

Over time, different methods, making use of a number of different sensors and algorithms, have been proposed to tackle AUVs localization problems. According to the authors in Paull et al., 2014, these techniques can be grouped into three main categories:

1. **Inertial/Dead Reckoning** - Inertial navigation makes use of data acquired from inertial sensors, such as accelerometers, gyroscopes or magnetometers, in order to estimate the vehicle's relative position, velocity and attitude. Nevertheless, all methods in this category are subjected to unbounded position error growth.

2. **Acoustic Transponders and Modems** - acoustic navigation methods are based on the measurement of the time of flight (TOF) signals that are sent to and received from external acoustic beacons and modems.

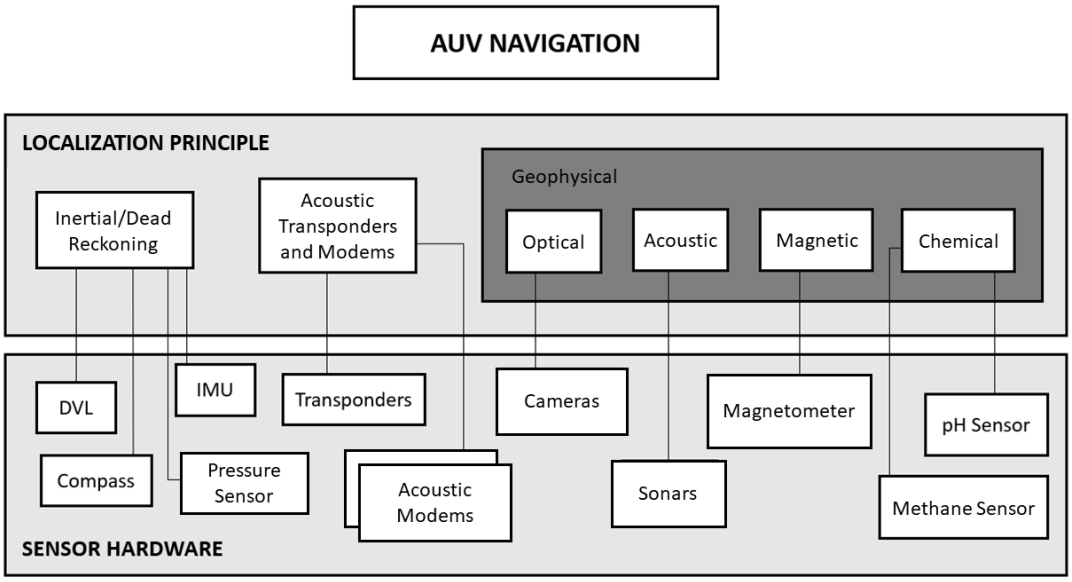


Figure 2.1: AUV localization techniques categorization. Adapted from Paull et al., 2014

3. **Geophysical** - the methods in this category use different sensors and processing techniques in order to detect, identify and classify environmental features. These features are then used as references to estimate the AUV's position.

Figure 2.1 presents a detailed categorization of existing AUV localization techniques.

2.1.1.1 Inertial/Dead Reckoning

Dead reckoning (DR) allows for an AUV to position itself autonomously without acoustic support from a ship or acoustic transponders. Upon knowledge of its orientation and velocity or acceleration vector, the AUV is able to estimate its new position. DR methodology is not a primary means of navigation, due to unbounded position error growth, but it is the basis for many modern navigation systems.

An inertial navigation system (INS) aims at improving the DR pose estimation by integrating data from motion sensors (accelerometers) and rotation sensors (gyroscopes). It is a compact, inexpensive and self-contained solution that does not require any external references. However, an INS can accumulate error over time since its position and velocity estimations are a result of mathematical integration of sensor data (that are subjected to drift over time). Therefore, the data provided by the INS must be combined with other navigation techniques for a more accurate navigation.

2.1.1.2 Acoustic Transponders and Beacons

Acoustic navigation techniques measure ranges through TOF of acoustic signals for posterior position estimation. The most common methods are presented hereafter:

- **Ultra-Short Baseline (USBL)** - the AUV is localized relatively to a surface vehicle that carries an array of acoustic transducers. TOF allows the calculation of the distance to the surface vehicle while orientation is acquired through the phase difference of the acoustic signals received by each transducer;

- **Short Baseline (SBL)** - same working principle as USBL. However, the transducers are positioned in the front and back of the surface vehicle, limiting the baseline to its length and, consequently, the position accuracy of the AUV;

- **Long Baseline (LBL)** - several transducers are placed over the operation area, at seabed level. AUV localization is performed by triangulation of the estimated range for each acoustic transducer;

- **GPS Intelligent Buoys (GIBs)** - utilization of transducers with GPS buoys in order to overcome the limitations of LBL, by assuring a two-way communication between the buoys and the AUV. The two way travel time of the acoustic signal is used to estimate the vehicle's position;

- **Single Transponder** - use of a single transponder to reduce the costs associated with multiple systems. The baseline is simulated by extending the range until the next signal is received;

- **Acoustic Modem** - the employed modems allow the transmission of information along with the signal used for range estimation. Therefore, before the mission, these need to be fixed and located in the global frame.

Trilateration and Triangulation

Trilateration algorithms consist on determining the vehicle's position based on the measurement of several distances to external beacons, whose location is well-known in advance. When, instead of distances, angles between beacons and the vehicle are involved, the approach is called Triangulation. With this information, the AUV's position can be calculated through a non-linear optimization problem, by minimizing the error between the acquired measurements and the expected ones.

Trilateration and triangulation algorithms present satisfactory performances if sensors provide data consistently. Nevertheless there are two major restrictions to these. Firstly, if less than three beacons are visible, it is impossible to extract the vehicle's position. Secondly, if vehicle and beacons all lie in the same circumference, position extrapolation is also not possible.

2.1.1.3 Geophysical

Geophysical navigation encompass any methods that make use of external environmental techniques for navigation purposes. The majority of these are able to achieve bounded position error

implement some form of Simultaneous Localization and Mapping (SLAM) technique (Paull et al., 2014). The main categories are:

- **Magnetic** - methods that make use of magnetic field maps for localization. It is the most unexplored strand;
- **Optical** - these methods usually use a monocular or stereo camera to capture images of the seabed. This information is the processed to help navigation;
- **Sonar** - their objective is to acoustically detect, identify and classify environmental features that might be useful for navigation. In the case of bathymetric sonars, features can be extracted almost directly from retrieved data. With sidescan (imaging) sonars, features have to be extracted through image processing techniques;
- **Chemical** - such methods are able to identify specific substances in the water, whose source is used as a reference point for navigation purposes.

2.1.2 Sensing

To deal with dynamical changes in the real world, AUVs make use of various sensors for localization purposes. It is important to know and understand their characteristics in order to integrate them in the desired systems and develop appropriated algorithms. Table 2.1 presents a summary of the most common solutions utilized for underwater localization.

In order to extract relevant information from the sensor data acquired, a number of algorithms have been purposed. Due to the wide range of different sensors used for underwater applications, formulations of their corresponding algorithms vary accordingly.

Sensor Fusion

It is common, when an AUV carries multiple sensors, to fuse their data in order to attain more accurate and robust information for navigation purposes than using solely the data from each individual sensor. In underwater navigation, sensor fusion is typically related to INS.

The most common algorithms for sensor fusion are Kalman Filter and its variants (Chen et al., 2013). Table 2.2 presents a brief review of said algorithms.

Scan Matching

Scan matching techniques aim at estimating the vehicle's translation and rotation movements through the overlapping of a scan contour with either a known map or a previous scan. These techniques can be classified into three different categories: feature-based techniques, compact data methods and point matching techniques. Furthermore, some algorithms have been purposely developed for AUV navigation, as the MSISpIC (Chen et al., 2013).

Sensor	Description	Performance	Cost
Compass	A compass provides an heading reference. Typical magnetic compasses do so by measuring the magnetic field vector, which will subject the sensor to bias in the presence of strong magnetic signature. Thus, in marine applications, is common to resort to gyrocompasses, since they measure heading through a fast spinning disc and the rotation of the earth. Therefore, they are unaffected by metallic objects.	Accuracy between 1^o to 2^o for mid-range price units.	On the order of hundreds of euros.
Pressure Sensor	Pressure sensors and barometers can be used for underwater depth estimation.	Possible accuracy on the order of a few millimeters.	On the order of hundreds of euros.
Doppler Velocity Log	The Doppler Velocity Log (DVL) makes use of acoustic measurements to estimate the AUV's surge, sway and heave velocities. It transmits acoustics pulses and measures the Doppler shifted returns off the seabed.	Standard deviation on the order of 0.3cm/s to 0.8cm/s.	On the order of tens of thousands of euros.
Global Positioning System	Global Positioning Systems (GPS) are common solutions for surface vehicles, since they estimate position using TOF of signals from synchronized satellites.	Typical accuracy between tens of meters and a few centimeter.	From hundreds to thousands of euros.
Sonar	A device for remote detection and location of objects present in the water through sound. There are two main types: passive sonars, listening devices that record sounds emitted by objects in the water, and active sonars, that produces sound waves of specific frequencies and listens for the echoes of these returned from objects in the water.	Resolutions depends on many internal and external factors.	From tens to hundreds of thousands of euros.
Inertial Measurement Unit	Inertial Measurement Units (IMU) use a combination of accelerometers and gyroscopes (sometimes, even magnetometers) to estimate the AUV's heading, velocity and gravitational forces. The gyroscope is responsible for measuring angular rates. The most widely used categories for underwater applications are Ring Laser (Fiber Optic) based and Microelectromechanical Systems based. An accelerometer is able to measure the force necessary to accelerate a given mass. The most common designs include pendulum, microelectromechanical systems and vibrating beam.	Gyroscope - drift variable between 0.0001^o/hr to 60^o/hr. Accelerometer - bias between 0.01mg to 0.001mg.	From hundreds of dollars to hundreds of thousands of euros.

Table 2.1: Common onboard AUV sensors used for localization. Adapted from Paull et al., 2014

Algorithm	Description
Bayes' Filter	Optimal but only computationally feasible for the simplest estimation problems.
Kalman Filter	State distribution assumed to be Gaussian and parameterized by mean and covariance.
Extended Kalman Filter (EKF)	Extension of the Kalman filter that allows dealing with non-linear processes or measurement models. Fast prediction operation but slower measurement update due to matrix inversion.
Unscented Kalman Filter (UKF)	Through higher computation reduces the linearization errors of EKF. Besides mean and covariance, also maps multiple sample points and re-parameterizes the output as Gaussian.
Extended Information Filter (EIF)	State distribution assumed to be Gaussian but parameterized by an information matrix and information vector. It is possible to process multiple measurements at once through addition. Slow prediction step due to matrix inversion, but faster measurement update. Calculating mean and covariance can be time consuming.
Particle Filter (PF)	Non-parametric representation of state distribution. Instead it makes use of discrete particles with associated weights. Can encompass non-Gaussian distributions and non-linear models. Yet, computational requirements scale with the number of particles.
Least Squares Regression	State estimation formulated as a least squares optimization that can be solved analytically. Pasts states are maintained which can be useful for navigation purposes.

Table 2.2: Common sensor fusion algorithms

2.1.2.1 Sonars and Data Interpretation

Sonars have been used for ocean mapping for decades now, even predating AUV technology. It is a fairly mature technology and, therefore, one of the most preferable for underwater applications (Sahoo et al., 2019), being used to detect and identify underwater structural features. A sonar operates at a specific frequency, in function of the required range and resolution.

Sonars can be classified into two main categories, **Imaging Sonars** and **Ranging Sonars**. The most common devices used for underwater navigation are presented in table 2.3.

Sonar	Category	Description	Pros	Trade-Off
Sidescan	Imaging Sonar	Multiple beams are emitted, perpendicularly to travel direction. Intensity of the returns is measured to create a 2D image.	Works at high speeds, providing for an high area coverage.	Resolution is inversely proportional to range.
Forward Looking	Imaging Sonar	In principle is similar to a sidescan sonar, but instead the beams are directed forward.	Fit for obstacle detection or as a nadir gap filter.	Limited distance to depth ratio (maximum 6:1); single angle of view.
Synthetic Aperture	Imaging Sonar	Virtual array synthesized through coherent processing of consecutive displaced returns.	Range independent form resolution.	Optimal at low speeds and deep waters.
Mechanical Scanning Imaging Sonar	Imaging Sonar	Single beam transducer associated to actuator for swath scanning.	Cheaper than multibeam.	Slow. Accuracy depends on AUV attitude.
Echo Sounder	Ranging Sonar	Transducer emits single, narrow beam. Allows for depth measurement.	Yield representation of seabed and possible obstacle underneath the AUV.	Point measurements in one direction.
Profiler	Ranging Sonar	Low-frequency echo sounders that are able to penetrate the seabed.	Data on subsurface features.	Resolution inversely proportional to penetration depth.
Multibeam	Ranging Sonar	Reflection's TOF used to assemble bathymetric maps.	More efficient data gathering than single beam.	Resolution inversely proportional to frequency. First reflection might not be captured.

Table 2.3: Main sonar imaging and ranging devices for underwater applications. Adapted from Paull et al., 2014

The quality and efficiency of the localization algorithm is extremely dependent on the quality of the acquired data. But, as with other acoustic sensors, sonars still suffer from many shortcomings:

- **low data rate**, which constrains the amount of data that can be transmitted;
- **high latency**, due to the low sound speed in the water (only $1500 m/s$);
- **variable sound speed** due to different water temperatures and salinity, which will affect the estimation of detection ranges;

- **multipath transmission** that occur due to the presence of both an upper (water surface) and lower (sea bottom) boundaries and highly variable sound speed, leading to the occurrence of multiple echoes.

Furthermore, the major issue affecting geophysical navigation is related to the shortage of unambiguous reference landmarks, of key importance for the accuracy of the navigation system (Sahoo et al., 2019). Nevertheless, sonar underwater applications for navigation purposes is one of the most active areas of research.

2.1.3 Environment Representation

An AUV's navigation procedure is inherently linked to the robustness and accuracy of its landmark detection and mapping procedures. For that, the vehicle's object detection algorithms are fundamental and must overcome two main problems: being able to distinguish between landmarks and false alarms and to accurately localize these in a frame of reference. It is with this information that the vehicle will be able to make changes to its path plan.

The environment information acquired through the AUV's onboard sensors must be compiled in order to allow for the detection of possible obstacles. Typical approaches rely on mapping techniques. For underwater applications, two solutions stand out: grid-based maps and feature-based maps.

The first approach is based on the discretization of the environment into a grid-based view. In Li et al., 2015 the authors make use of a forward looking sonar and a global occupancy grid (Elfes, 1989) for environment mapping. The grid cells are updated through an implementation of a Bayes' filter. In Gao and Xu, 2013, the same type of sensor and mapping technique are used, but the authors propose a pre-processing of the retrieved acoustic images through K-means clustering segmentation, followed by morphological processing of said image, in order to remove noise. Due to the high computational requirements of such algorithms, Ganesan et al., 2016 presents an adaptation of it into a local occupancy grid, of smaller and fixed size, whose cells are update through a measurement model (based on a Bayes' filter) and the vehicle's motion model.

Feature-based approaches focus on the collection of environmental features. Usually, such techniques are avoided for unstructured environments such as oceans or lakes, due to a lack of unambiguous features. Nevertheless, Dinuka and Bandara, 2017 is dedicated to the study of common feature extractors for surface applications and their performance in underwater environments, with a side-scanning sonar. Through the combination of different feature extractors, very interesting results were obtained. In Li et al., 2017 a simple thresholding technique is applied to the acoustic images and, by making use of prior knowledge of the environment the AUV navigates, features are extracted. In the context of image processing and analysis, Esrafilian and Taghirad, 2017 proposes a feature-based monocular SLAM implementation for an autonomous flight system. Here, FAST corner detector (Dinuka and Bandara, 2017) and ORB extractor (Mur-Artal et al., 2015) are used for feature extraction. Furthermore, in Millar, 2015 the ranges returned by five ranging sonars are,

upon validation, used to generate possible obstacles (each assigned a radius, position and probability factor). Additional returns inside of an object radius increased its probability factor. Only those who exceed a certain probability value are considered.

2.1.3.1 SLAM

SLAM algorithms have been developed as a solution for solving localization and mapping problems all at once, that arises when vehicle does not have previous access to a map of the environment it navigates, having only access to measurements and controls (Thrun, 2002). They allow a vehicle to be placed at an unknown location in an unknown environment and to gradually build a map of said environment, while simultaneously locating itself within this map. The most common SLAM algorithms and their respective advantages and disadvantages are presented on table 2.4.

Nevertheless, there is a set of challenging issues associated to underwater SLAM, related to the unstructured nature of underwater environments and the shortage of reliable features, which are usually scale dependant and sensitive to viewing angle and scale. Furthermore, regarding the map building procedure, problems also arises related to correspondence: when a environmental feature is detected, the algorithm must infer concerning its relation to previously detected features (Thrun, 2002), so as to find out which are the same.

SLAM Algorithm	**Advantages**	**Disadvantages**
EKF SLAM	Good overall performance when features are present and distinct.	Adding new features no state space requires quadratic time.
SEIF SLAM	Updates performed in constant time. Good choice for multiple-robot SLAM.	Information matrix actively sparsified. Matrix inversion required for map recovering.
FastSLAM	No dependence on motion models parametrization.	Particle set defines ability to close loops.
GraphSLAM	Update of previous poses for data post-processing.	Computationally heavier. Hard to recover covariances.
AI SLAM	More efficient.	Trainning or parameter tunning is required.

Table 2.4: Advantages and disadvantages of common SLAM algorithms

EKF SLAM

One of the most influential SLAM algorithms in existence, the EKF SLAM algorithms employes the EKF algorithm in order to solve the SLAM problematic (Alcantarilla et al., 2010). However, as it would be expected, is application is subjected to a number of approximations (Bailey et al., 2006) and assumptions:

- **Feature-based Maps** - the maps built are composed of a number point landmarks, typically less than 1000 (Thrun, 2002). The performance of the map building operation is heavily linked to the employed feature detection methodology.

- **Gaussian Noise** - Gaussian noise assumptions are made regarding robot motion and perception.

- **Positive Measurements** - the EKF SLAM algorithm can only process measurements related to the sighting of landmarks. It cannot process negative measurements that could occur due to the sensor's inability to detect landmarks. This comes as a natural consequence of the Gaussian belief representation.

As any SLAM algorithm, EKF SLAM, besides estimating the vehicle's pose x_t, also estimates the position of every detected environmental feature during an operation. This is achieved by, primarily, extending the state vector y_t to include landmark coordinates, as follows (Thrun, 2002):

$$\mathbf{y}_t = \begin{bmatrix} \mathbf{x}_t \\ \mathbf{m} \end{bmatrix} \tag{2.1}$$

$$= \begin{bmatrix} x & y & \theta & m_{1,x} & m_{1,y} & s_1 & \dots & m_{N,x} & m_{N,y} & s_N \end{bmatrix}^T , \tag{2.2}$$

where N represents the number of observed landmarks, x, y and θ the vehicle's coordinates at time t, $m_{i,x}$ and $m_{i,y}$ the landmark's coordinates, for landmarks $i = 1...N$ and s_i its signature, resulting in a vector of dimension $3+3N$. This will also result in an extended state estimation vecto, μ_t.

As the vehicle moves, the state vector must be updated according to its motion function $g(u_t, y_{t-1})$, which must only affect the vehicle's pose, as all landmarks are expected to remain at stationary positions:

$$y_t = g(u_t,\, y_{t-1}) + \mathcal{N}(0,\, F_x^T\, R_t\, F_x) \,, \tag{2.3}$$

where u_t represent the vehicle's input controls at time t, y_{t-1} the state vector at time $t-1$, R_t the motion noise covariance and F_x is a matrix that transforms a 3-dimensional vector into a vector of dimension $3N+3$, with only the first three columns different than 0, such that

$$F_x = \begin{bmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \end{bmatrix} . \tag{2.4}$$

As characteristic of the EKF algorithm, the motion function g is approximated trough a first order Taylor expansion. So

$$g(u_t,\, y_t) \approx g(u_t,\, \mu_{t-1}) + G_t \cdot (y_t \,-\, \mu_{t-1}) \,, \tag{2.5}$$

where the Jacobian G_t represents the Jacobian of g, with respect to μ_{t-1}, evaluated at μ_{t-1} and u_t, which can be further decomposed into the sum of an identity matrix of dimension $(3n+3)$ x $(3n+3)$ with a low-dimensional Jacobian g_t, that characterizes the vehicle's position change

$$G_t = I + F_x^T\, g_t\, F_x \,. \tag{2.6}$$

The results of the motion update cycle are represented through the mean $\overline{\mu}_t$ and the covariance

$\overline{\Sigma}_t$ of the state estimation at time t and computed as follows:

$$\overline{\mu}_t = g_t(u_t,\ \mu_{t-1}) \tag{2.7}$$

$$\overline{\Sigma}_t = G_t\ \Sigma_{t-1}\ G_t^T\ +\ F_x^T\ R_t\ F_x\ . \tag{2.8}$$

At this time, the integration regarding the acquired measurement z_t is possible, which is performed taking into account its corresponding measurement function $h(y_t,\ j)$:

$$z_t^j = h(y_t,\ j) + \mathcal{N}(0,\ Q_t)\ , \tag{2.9}$$

where j represents the index of the observed landmark at time t and Q_t the measurement noise covariance. Once again, the measurement function is approximated via Taylor expansion

$$h(y_t,\ j) \approx h(\overline{\mu}_t,\ j) + H_t^j(y_t\ -\ \overline{\mu}_t)\ , \tag{2.10}$$

with H_t^j representing the derivative oh h regarding the state vector y_t, which factors into a low-dimensional Jacobian h_t^j and a matrix of dimension $(3N+3)$ x 6, $F_{x,j}$, that extends the dimension of h_t^j to the dimension of the full state vector. So,

$$H_t^i = h_t^i\ F_{x,j} \tag{2.11}$$

and

$$F_{x,j} = \begin{bmatrix} I_{3\times3} & 0_{3\times(3N-3)} & 0_{3\times3} & 0_{3\times(3N-3j)} \\ 0_{3\times3} & 0_{3\times(3N-3)} & I_{3\times3} & 0_{3\times(3N-3j)} \end{bmatrix}. \tag{2.12}$$

It is important to note that, when a new landmark is observed, its initial position estimation is initialized with its expected position, as follows:

$$\begin{bmatrix} \overline{\mu}_{j,x} \\ \overline{\mu}_{j,y} \\ \overline{\mu}_{j,s} \end{bmatrix} = \begin{bmatrix} \overline{\mu}_{t,x} \\ \overline{\mu}_{t,y} \\ s_t^j \end{bmatrix} + r_t^j \begin{bmatrix} \cos(\phi_j^t + \overline{\mu}_{t,\theta}) \\ \sin(\phi_j^t + \overline{\mu}_{t,\theta}) \\ 0 \end{bmatrix}. \tag{2.13}$$

For each acquired measurement, a measurement estimation is calculated and, through that, the corresponding Kalman gain, K_t^j, is computed through:

$$K_t^j = \overline{\Sigma}_t\ H_t^{jT}\ (H_t^j\ \overline{\Sigma}_t\ H_t^{jT} + Q_t)^{-1}\ . \tag{2.14}$$

This information is finally incorporated into the final state and covariance estimation:

$$\mu_t = \overline{\mu}_t\ +\ \Sigma_t^j\ (z_t^j\ -\ \hat{z}_t^j) \tag{2.15}$$

$$\Sigma_t = \overline{\Sigma}_t\ -\ K_t^j\ H_t^j\ \overline{\Sigma}_t\ . \tag{2.16}$$

It is important to note that, due to the Kalman gain being fully populated for every state

variable, the observation of a landmark does not only improve the position estimation of this landmark but also that of the remaining observed landmarks. This is a direct result of improving the position estimation of the vehicle, since it will eliminate some uncertainty regarding preciously detected landmarks.

2.2 Feature Extraction Algorithms

Feature extraction algorithms have been of great importance for many computer vision applications such as visual odometry, pose estimation, object recognition, object tracking, among others (Tareen and Saleem, 2018). Their main goal is to determine key features on provided images which remain locally invariant, so that is possible to detect these even in the presence of rotation or scale changes (Dinuka and Bandara, 2017). These procedures present a robust solution to solve problems in identifying reliable reference landmarks for position estimation, which affect geophysical navigation procedures and, more precisely, sonar acoustic images. They encompass two main sub algorithms: feature detectors and feature descriptors.

Feature detectors enable the detection of feature-points (also called interest points or keypoints) in an image. Typically, the detected features take the form or corners, blobs, edges, junctions or lines. These are subsquentely logically described through the unique patterns associated to the pixels in their neighbourhood (Tareen and Saleem, 2018). This descriptor enables the feature effective recognition, which is of utmost importance for posterior matching. Some feature detectors are made available with their proper feature descriptor, such as SURF, BRISK or ORB, while others are designed individually and can be paired with a range of descriptors. The most common methods are presented in table 2.5. Hereafter is presented a brief explanation of the feature detectors and descriptors used in this study.

Detectors	Laplacian of Gaussian (LoG)
	Harris & Stephens corner detector
	Difference of Gaussian (DoG)
	Determinant of Hessian (DoH)
	Salient Regions
	SUSAN
	Morphological Interest Points
Descriptors	Local Binary descriptors
	Spectra descriptor
	Basic Space descriptor
	Polygon Shape descriptor

Table 2.5: Most common techniques for feature detection and description

2.2.1 SURF - Speeded UP Robust Features

The SURF algorithm (Bay et al., 2008) is composed of a feature detector and a descriptor. It is inspired on the SIFT algorithm (Low, 2004), and takes advantage of integral images (Bay et al., 2008) for better performance.

For better performance in computation time and accuracy, the SURF detector is based on the computation of the Hessian matrix. Furthermore, we rely on the determinant of the Hessian for location and scale selection. Therefore, for a given point x = (x, y), at scale σ, the Hessian matrix $H(\mathrm{x},\sigma)$ is defined as follows

$$H(\mathrm{x},\sigma) = \begin{bmatrix} L_{xx}(\mathrm{x},\sigma) & L_{xy}(\mathrm{x},\sigma) \\ L_{xy}(\mathrm{x},\sigma) & L_{yy}(\mathrm{x},\sigma) \end{bmatrix}, \tag{2.17}$$

where $L_{xx}(\mathrm{x},\sigma)$ represents the convolution of the Gaussian second order derivative, $\frac{\partial^2}{\partial x^2} g(\sigma)$, such as for $L_{xy}(\mathrm{x},\sigma)$ and $L_{yy}(\mathrm{x},\sigma)$. Since, in practice, the application of Gaussian filters does not prevent the occurrence of aliasing, making them non-ideal, the SURF descriptor approximates these with box filters (Bay et al., 2008), which allows for a faster processing of the image. Due to the application of box filters and integral images, scale space is analyzed by simply up-scaling filter size. Subsequently, a non-maximum suppression in a 3 x 3 x 3 neighbourhood is applied, in order to localise interest points in the image and over scales.

The SURF descriptor implies two main operations. Firstly, a reproducible orientation is assigned to each previously detected interest point. Thus, Haar Wavelet (Dengsheng, Zhang, 2019) responses are calculated in the x and y directions within a circular neighbourhood of the interest point. Again integral images are employed for a faster filtering. Then, the descriptor is built by firstly constructing a square region around the keypoint and oriented according to the calculated orientation. Haar Wavelet responses are calculated for several sub regions of our initial square region, which will compose the feature descriptor. Scale invariance is achieved by turning the descriptor into a unit vector.

2.2.2 BRIEF - Binary Robust Independent Elementary Features

The BRIEF descriptor (Calonder et al., 2010) makes use of binary strings, in an effort, according to the authors, to achieve more efficient computation times and memory storage by constructing a shorter and simpler descriptor.

This algorithm was developed over the premise that image patches may be effectively classified through a relatively small number of pair-wise intensity comparisons. Therefore, by defining a series of tests τ on a patch p of size S x S such as

$$\tau(\mathrm{p};\mathrm{x},\mathrm{y}) = \begin{cases} 1 & \text{if } \mathrm{p(x)} < \mathrm{p(y)} \\ 0 & \text{otherwise} \end{cases}, \tag{2.18}$$

where p(x) represents the pixel intensity at x = (u, v^T), on a set of n_d (x,y)-location pairs it is possible to create a descriptor, which is presented as the following n_d-dimensional bitstring

$$f_{n_d} = \sum_{1 \leq i \leq n_d} 2^{i-1} \, \tau(\mathrm{p}; \mathrm{x}, \mathrm{y}). \tag{2.19}$$

The tests locations are selected through a sampling geometry based on a Gaussian distribution.

However, since the tests only take into account the information of single pixels, these are very noise-sensitive. Therefore, a smoothing of the patch is performed before building the descriptor, in order to reduce sensitivity and increasing stability and repeatability of these.

2.2.3 Harris Corner Detector

One of the most popular interest point detectors, the Harris Corner Detector (Derpanis, 2004) is based on the local auto-correlation function of a signal. When applied to an image, it allows to measure local changes with patches shifted by a small amount in different directions. So, for a given shift $(\Delta x, \Delta y)$ and a image point (x, y), the auto-correlation function is defined as follows:

$$c(x,y) = \sum_{W} [I(x_i, y_i) - I(x_i + \Delta x, y_i + \Delta y)]^2, \tag{2.20}$$

where $I(\cdot, \cdot)$ represents the pixel intensity and (x_i, y_i) are points in the Gaussian window W centered on (x, y). By approximating the shifted image by a Taylor expansion, the auto-correlation function can be expressed as:

$$c(x,y) = \begin{bmatrix} \Delta x & \Delta y \end{bmatrix} \begin{bmatrix} \sum_W (I_x(x_i, y_i))^2 & \sum_W I_x(x_i, y_i) I_y(x_i, y_i) \\ \sum_W I_x(x_i, y_i) I_y(x_i, y_i) & \sum_W (I_y(x_i, y_i))^2 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \tag{2.21}$$

$$= \begin{bmatrix} \Delta x & \Delta y \end{bmatrix} C(x,y) \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}, \tag{2.22}$$

where the matrix $C(x, y)$ captures the intensity structure of the local neighbourhood. By analysing its eigenvalues, it is possible to extract some conclusions regarding the analysed points. If both eigenvalues are small, that indicates that the windowed region is of approximately constant intensity. If one eigenvalue is high and the other low, that is indicative of an edge while a corner results in simultaneous high eigenvalues.

2.2.4 FAST - Features from Accelerated Segment Test

The FAST detector (Rosten et al., 2010) is essentially a corner detector. Despite not being much robust to noise, this algorithm is, in general, much faster than the other existing feature detectors (Dinuka and Bandara, 2017).

For a corner candidate p, the detection procedure, by analysing the circular neighbourhood of 16 pixels around that point, classifies it as a corner if a set of n contiguous pixels, with all intensities higher than the one of the candidate pixel, I_p, plus a threshold t, or lower than I_p minus the same

threshold, is detected. This analysis of the corner candidate's neighbourhood is performed by testing the intensity in each position $x \in \{1...16\}$, relative to p, denoted by $p \rightarrow x$, and classifying it as follows

$$S_{p\rightarrow x} = \begin{cases} d, & I_{p\rightarrow x} \leq I_p - t \quad (darker) \\ s, & I_p - t < I_{p\rightarrow x} < I_p + t \quad (similar) \\ b, & I_p + t \leq I_{p\rightarrow x} \quad (brighter) \end{cases}. \tag{2.23}$$

After this initial assessment, the points are divided into three groups, according to the previous classification results, and analysed through a decision tree, which validates the candidate as a corner or not. This results are lastly subjected to a non-maximum suppression performed in a 3 x 3 mask.

2.2.5 BRISK - Binary Robust Invariant Scalable Keypoints

The BRISK algorithm (Leutenegger et al., 2011) was developed with the objective of achieving high quality performance at lower computational costs than reference algorithms such as SURF or SIFT (Leutenegger et al., 2011). For that purpose, it makes use of the FAST detector and bitstring-based descriptor.

The BRISK detector presents an extension of the FAST detector. With the aim of achieving invariance to scale, FAST is employed for scale-space analysis. Therefore, a scale-space pyramid is constructed, whose layers consist of n octaves c_i and n intra-octaves d_i, formed by progressively half-sampling the original image. Afterwards, the FAST detector is applied to each octave and intra-octave, using the same threshold, in order to identify possible interest points. Interpolation is applied at the boundaries of the patch. Finally, the detected points are subjected to non-maximum suppression in scale-space.

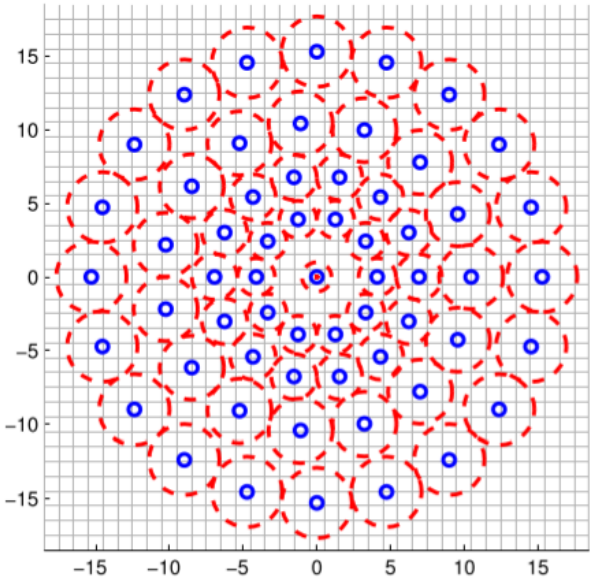


Figure 2.2: BRISK sampling pattern presented in Leutenegger et al., 2011

For the descriptor building, the algorithm starts off by sampling the neighbourhood of each interest point based on an unique pattern, shown in figure 2.2 (N location defined equally spaced on circles concentric with the interest point). To avoid aliasing effects, a Gaussian smoothing is applied to the points in the pattern. For each sampled point pair (p_i, p_j), the local gradient $g(p-i, p-j)$ is computed by

$$g(p_i, p_j) = (p_j - p_i) \cdot \frac{I(p_j, \sigma_j) - I(p_i, \sigma_i)}{||p_j - p_i||^2}. \tag{2.24}$$

Then, the overall characteristic pattern direction of the interest point is estimated through

$$g = \begin{pmatrix} g_x \\ g_y \end{pmatrix} = \frac{1}{L} \cdot \sum_{(p_i, p_j) \in \mathscr{L}} g(p_i, p_j), \tag{2.25}$$

where $\mathscr{L}$ represents the subset of long-distance pairs.

The binary descriptor is afterwards built by re-sampling the interest point neighbourhood with the sampling pattern oriented accordingly to the previously calculated direction. Each of the b bits of the descriptor are determined by performing a series of intensities comparisons between all point pairs.

2.2.6 ORB - Oriented FAST and Rotated BRIEF

As the name implies, the ORB algorithm (Rublee et al., 2011) makes use of a combination of the FAST detector and the BRIEF descriptor, whilst proposing solutions to deal with these methods limitations, such as the lack of an orientation component in FAST or the impossibility of computing oriented BRIEF features (Rublee et al., 2011).

The ORB interest point detection routine is focused on corner-like features. Therefore, firstly, FAST interest points are extracted from each level of the scale pyramid of the analysed image and, afterwards, a Harris corner measure is used to remove responses along edges. The orientation assignment is then performed based on the calculation of the intensity centroid (Rublee et al., 2011). For each interest point and associated neighbourhood, the corresponding moments are given as:

$$m_{pq} = \sum_{x,y} x^p y^q I(x, y), \tag{2.26}$$

which allows to calculate its centroid through

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right). \tag{2.27}$$

Therefore, the interest point orientation is given by

$$\theta = \text{atan2}(m_{01}, m_{10}), \tag{2.28}$$

where atan2 represents the quadrant aware version of the arctangent.

Regarding the descriptor construction, the BRIEF descriptor is based on the performance of a series of binary intensity tests on a smoothed version of the original image. In order to achieve rotation invariance, a steered version of the BRIEF algorithm is employed. Here, for any set of n binary intensity tests at location (x_i, y_i), S is defined as

$$S = \begin{pmatrix} x_1, ..., x_n \\ y_1, ..., y_n \end{pmatrix}. \tag{2.29}$$

Making use of the interest point orientation θ and the corresponding rotation matrix R_θ, a steered version of S, S_θ is defined as follows:

$$S_\theta = R_\theta S, \tag{2.30}$$

which defines the new rotation invariant test locations.

2.2.7 Feature Matching

Feature extraction algorithms are a valuable tool for applications related to image analysis. However, in a number of them, such as pose estimation or object tracking, the detection of key feature points, by itself, does not provide sufficient information for accomplishing the intended goal. Therefore, this procedure must be complemented with a mechanism that enables the association of the the same feature point in different images, taking advantaged of the corresponding feature descriptor. Thus, usually, a feature matching procedure is associated to the application of feature extraction techniques. The general feature matching procedure is further detailed on figure 2.3.

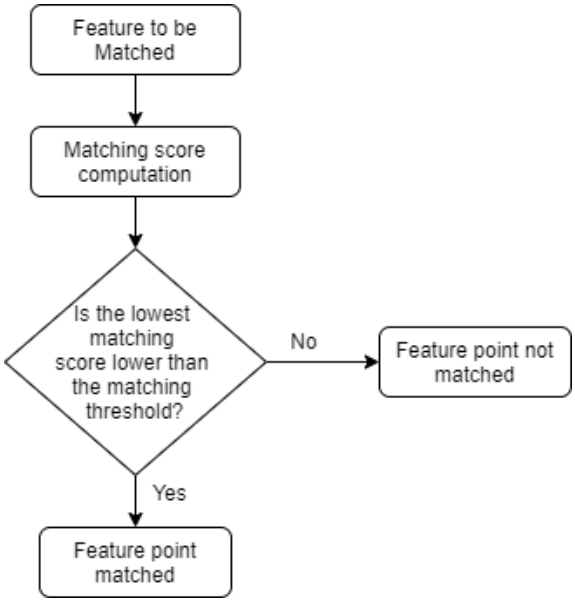


Figure 2.3: The feature matching procedure

Here, feature descriptors are compared and the points that present minimum deviations between their descriptors are taken as matches (Dinuka and Bandara, 2017). The matching procedure can be carried out according to different strategies, such as nearest neighbour, threshold based matching or nearest neighbour distance ratio (Tareen and Saleem, 2018). The comparison between a given feature point and match candidates feature points is performed based on the computation of L1 or L2 (Euclidean) norms, in the case of string based descriptors, or the Hamming distance, in the case of binary descriptors, between their corresponding descriptors.

2.3 Summary

In this chapter, a review of the most common algorithms and sensors used for underwater navigation has been presented. Common localization principles have been introduced in greater detail, making it possible to argue that one of greatest issues concerning this topic is the growing drift in position estimation. A number of solutions employing different sensor technologies have been purposed in order to solve this problem.

Sonars represent one of the most mature technologies for underwater localization and thus have been more carefully analysed in the present chapter. However, these are affected by a number of problems which pose significant difficulties to the identification of reliable environmental features that would allow the estimation of the vehicle's position. Therefore, a detailed overview of feature extraction algorithms has also been presented, since these techniques are intended to address such problems. Despite extensive performance comparisons for optical imaging being presented in literature, the same does not stand for acoustic imaging and, therefore, further information regarding this topic is required.

Chapter 3

System Overview

In this chapter, a more comprehensive overview of the developed solution is presented. Further detail is given regarding the logical construction of the overall proposed solution and of each specific block. Figure 3.1 presents the structure of this solution in the form of a block diagram.

Firstly, a more comprehensive analysis of the problems portrayed by acoustic images is in order. Imaging sonars are able to emit acoustic beams and measure the intensity values of con-

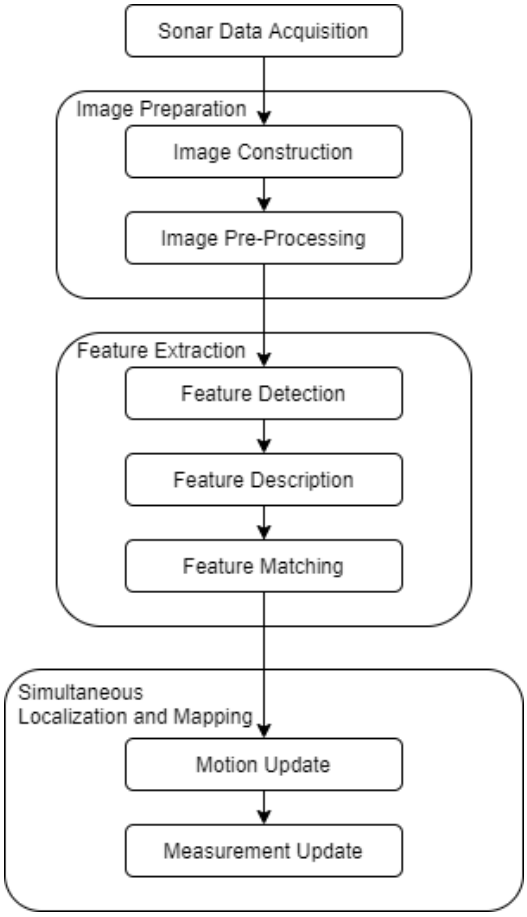
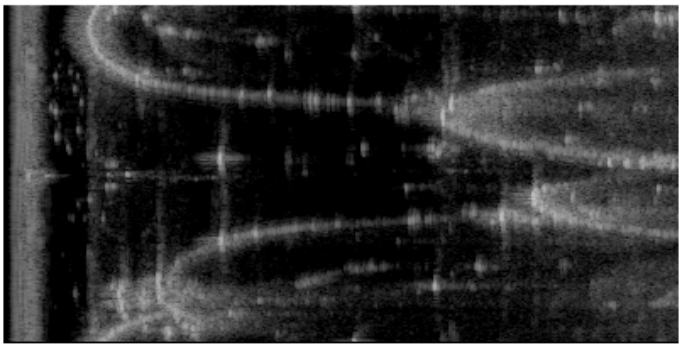


Figure 3.1: Block diagram representation of the developed solution

(a) Water tank structure used for testing



(b) An acoustic image from the water tank

Figure 3.2: Testing environment and corresponding acoustic representation

sequent echoes from particular places. This information can be subsequently used to generate acoustic images of the surrounding environment. Figure 3.2 presents the testing environment for the work carried out, an almost square water tank, and a acoustic image acquired from inside this same structure. The arches observable in the image represent the water tank walls and the intersections of these, its corners. These contrasting shapes are due to the nature of the sonar data, since every detected reflection is represented by a detection angle and a detection range, thus, in polar coordinates. Everything represented after these arches is the result of acoustic echoes and multipath transmission. Also, as it can be observable, there is quite a deal of background noise affecting these images. Further complications arise as the AUV starts it motion, since, again due to the nature of the sonar's data acquisition process, the structures visible in the images are affected by geometric deformation.

The literature review carried out for the construction of chapter 2 allowed a deeper study of feature extraction algorithms. The robustness to noise and geometric deformation presented by each detection and description procedure studied enable a satisfactory behaviour under the challenges presented by optical imaging, at a significant reduced computation time cost. Furthermore, the association performed between features in consecutive images by the matching procedure presents a direct and simple solution for the correspondence problem in state estimation procedures (Thrun,

2002).

As already pointed out in chapter 1, the developed solution is intended to be based on the application of a MSIS for the collection of environmental data. This data consists of a series of intensity measurements, which are organized in the form of intensity arrays of the emitted and subsequently reflected acoustic signals. Therefore, for each scanning angle α_i, an ordered set of m intensities values, represented by 8-bit integers, is generated (Fula et al., 2018). The total number of measurements bins, m, returned by the sonar, for each scanning angle, is a function of the maximum range and the bin length for which the sonar is configured. The entirety of the acoustic scan is composed through the rotation of the sonar's head. Therefore, the first goal of the **Image Preparation** module is the construction of the acoustic images for posterior analysis. This procedure is further detailed in chapter 4, since the coordinate system considered in this procedure requires a careful analysis. The second goal of this module is to perform a pre-processing of the acoustic image generated, so as to remove some of the noise present and improve some of its characteristics. So, the final goal of the **Image Preparation** module is to obtain a filtered acoustic image, in order to enhance the performance of the **Feature Extraction** module and, consequently, of the overall solution.

The **Feature Extraction** module development is based on the application of a feature extraction algorithm, making use of the characteristics of feature detectors and feature descriptors. The selection of the methodology to employ requires further analysis of some of the most common solutions for feature extraction, which is presented in chapter 4. The need for this comparison arises from the diversity of solutions presented in literature and a lack of more comprehensive studies regarding their applicability for the case of underwater acoustic imaging. This module is also responsible for performing the association of previous and current detected features.

The set of matched features acquired through the **Feature Extraction** module requires further processing so that a trustworthy AUV pose estimation may be retrieved, as well as a representation of the environment the vehicle navigates. The objective of the **Simultaneous Localization and Mapping** module is to perform this pose estimation, while also creating and constantly updating a map of the vehicle's surroundings. The development of this module is carefully detailed in chapter 6.

3.1 Tools and Methods

For the purpose of this dissertation, the SHAD AUV (Small Hovering AUV with Differential actuation) (Goncalves et al., 2016) was the robotic platform utilized. SHAD is a micro-sized AUV, modeled on a torpedo-like geometry, designed for operations in indoor and outdoor environments. Its dimensions ($0.15m$ hull diameter and $1.2m$ length) and weight ($16kg$) make it a vehicle easy to deploy and operate. Furthermore, it presents a modular design that allows expanding its structure making use of modules that are previously developed and tested in isolation. The thrusters configuration grant SHAD four degrees of motion freedom (surge, heave, pitch and yaw).

MSISs are of particular interest for AUV navigation due to the trade-off offered between cost and performance. Moreover, they enable covering quite extensive areas, over a few dozen of meters, at configurable sweeping angles. Therefore, in the context of this study, a Tritech Micron sonar (International, n.d.) has been equipped on the vehicle, allowing the application of echo-localization techniques. This sonar presents an operating frequency of $700kHz$, associated to a cone shaped beam of 35^{o} vertical width and 3^{o} horizontal width. Furthermore, it presents a configurable scanning range, from a minimum of $0.3m$ up to a maximum of $75m$, and a configurable scanning resolution, of 0.45^{o}, 0.9^{o} or 1.8^{o}. Lastly, the scanned sector may also be configured and is variable up to 360^{o}.

Regarding the data acquisition procedures and further tests performed in the context of this dissertation, these were undertaken in the water tank presented in figure 3.2a, an almost square shaped tank of $4.4m$ length, $4.6m$ width and $1.8m$ deep. Moreover, the acquisition of ground-truth information in the several testing scenarios considered was performed through physical measurement.

Chapter 4

Keypoint Detection and Description Analysis

4.1 Introduction

Accurate navigation is fundamental for an AUV to be able to safely explore unknown environments. Therefore, sensors with high resolution and accuracy are desirable but the rapid attenuation of higher frequency signals and the unstructured nature of the underwater environment pose great restrictions to the range of solutions available. For their overall robustness, sonars emerge as one of the most preferable technologies for this purpose. When deployed on an AUV, sonars are able to generate acoustic images of the AUV's surroundings through natural features present in this environment, as shown in figure 4.1.

In image based navigation, natural features, depicted on the acquired images, are used for localization and obstacle detection purposes. However, such constitutes no ordinary task, since acoustic images are affected by a number of problems, such as the presence of acoustic shadow, low resolution, distortion and range-varying attenuation (Tueller et al., 2019). Therefore, in recent years, there has been increased research interest on this thematic, with especial focus on feature extraction algorithms.

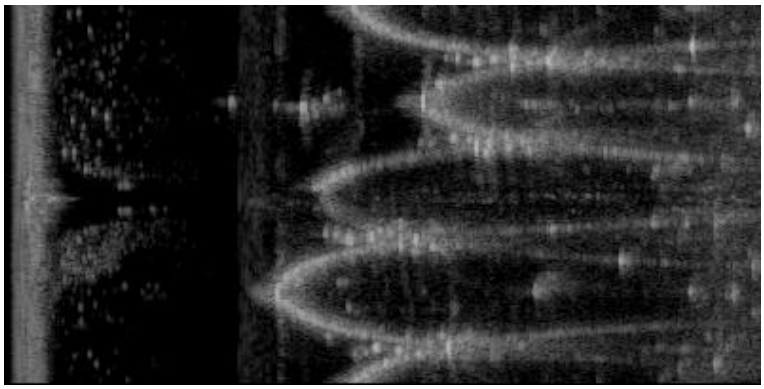


Figure 4.1: Acoustic image obtained through mechanical scanning imaging sonar

However, most of the feature extraction algorithms available have been originally developed for optical images (Dinuka and Bandara, 2017). Therefore, it is important to identify the most suitable algorithms to face the challenges presented by acoustic images and the environment in question. Some work has already been developed on this subject but the conclusions diverge. As reported in the literature, for traditional optical images, ORB and BRISK are better in repeatability than SURF or SIFT and, in addition, the most efficient ones (Tueller et al., 2019). Furthermore, SURF and SIFT are found to be the ones with higher computational costs (Tareen and Saleem, 2018). FAST, Harris and Shi-Thomas are very susceptible to scale variations (Tareen and Saleem, 2018). SIFT is the most accurate algorithm (Tueller et al., 2019) while SURF presents itself as the most robust one (Dinuka and Bandara, 2017). However, when it comes to acoustic imaging, these premises do not hold true. In this scenario, Harris and FAST are able to extract the most number of keypoints (Tueller et al., 2019) but present low matching ratings. SURF presents the highest matching ratio and an overall robustness to changes in rotation, scale and brightness (Dinuka and Bandara, 2017). BRISK returns the lesser number of keypoints (Dinuka and Bandara, 2017). Furthermore, it has been noticed that, for acoustic images, the association between the same feature in different images is troublesome, due to their properties.

Therefore, this chapter presents a performance analysis and comparison of the SURF, ORB, BRISK and SURF-Harris algorithms applied to acoustic images, generated through a MSIS. This analysis is based in the collection of typical statistics in image processing evaluation but also in a more qualitative evaluation, since some aspects of each algorithm's performance are quite to laborious quantify.

4.2 Feature Extractors Performance Analysis

Feature extraction algorithms have proven to be an useful tool for visual odometry purposes. Moreover, due to the characteristics of acoustic images and feature detectors capabilities, initial studies point to the possibility of making use of these techniques for object detection purposes, at low computational costs.

Therefore, since the success of sonar image processing based navigation heavily depends on the performance of the employed feature extraction algorithm, a comparison of the SURF, BRISK, ORB and SURF-Harris (proposed in Dinuka and Bandara, 2017, it combines the Harris detector and the SURF descriptor) algorithms was executed and is presented hereafter, in order to assess their adequacy to the proposed task.

4.2.1 Methodology

For the purpose of this study, MATLAB R2019a software was used, along with functions and tools available in the Image Processing Toolbox, for image preparation and feature extraction and matching. Furthermore, in order to gather the necessary data for the performance evaluation, the SHAD AUV, equipped with a Tritech Micron Sonar was deployed. The sonar is mounted in the

forward hull of the vehicle, orientated accordingly to its longitudinal axis. The comparison was performed taking into account the image processing methodology presented in figure 4.2.

Figure 4.2: Image processing procedure for feature extraction comparison

4.2.1.1 Sonar Data Acquisition

Data surveys were carried out in an almost square water tank, using the SHAD AUV, both depicted in figure 4.3. Furthermore, a floater was also deployed and moored close to the left bottom corner of the tank, so as to provide for a more distinguishable obstacle for later studies.

During the deployment of SHAD, the installed sonar was able to collect acoustic data through a series of 360^{o} scans of the AUV's surroundings. A total of 19 scans were performed, divided into 2 sets, one with 8 scans and other with the remaining 11. For the first group, $dataset1$, the AUV was kept immobile at surface, at the right top corner of the tank. The second scan, $dataset2$, was collected with the vehicle still at surface but, in this case, moving, at almost constant velocity, from the top right corner of the tank to the left one. This scenarios were designed in order to better portray missions situations and preform an analysis in the light of the challenges presented by these. Ground-truth measurements were performed regarding the AUV's initial and final positions and the floater's position, for both datasets. The tank's dimensions were also measured.

4.2.1.2 Acoustic Image Composition

For the purpose of this study, the sonar was calibrated for a maximum range of 5.0 meter, with a bit length of $1.25x10^{-2}$ meters and an angle step of 1.8^{o}. Therefore, each collected intensity array is composed of 399 bins and a single scan comprises 200 arrays. The acquisition of a complete scan

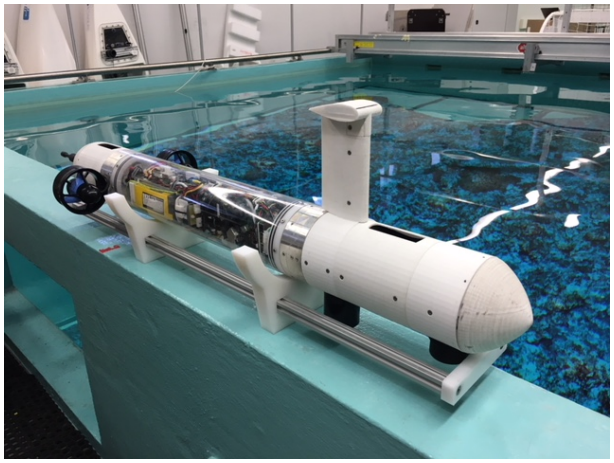
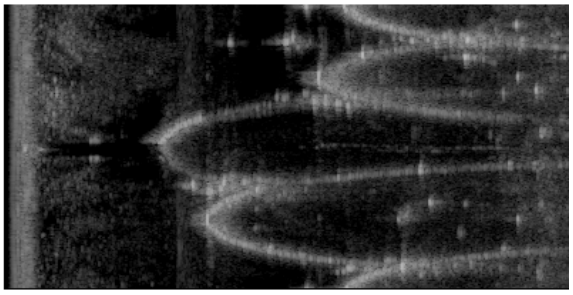
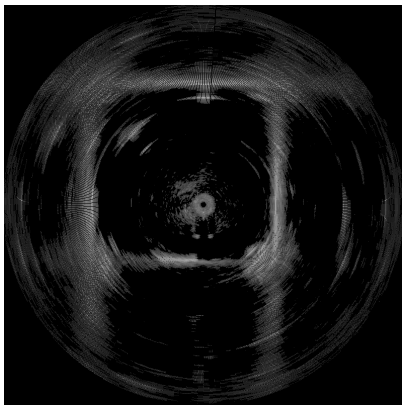


Figure 4.3: The SHAD AUV and the Water tank structured used for testing procedures

requires a total time of 8.0 seconds. These parameters are key for the construction of an acoustic image. For this purpose, it is also important to take into consideration that acoustic images can be presented according to a cartesian coordinate system or a polar coordinate system.



(a) Image composed based on polar coordinate system



(b) Image composed based on cartesian coordinate system

Figure 4.4: Results of different image composition techniques

As detailed above, each intensity value is associated with a detection position, represented, by default, in polar coordinates, through the scanning angle α_i and the bin position ρ_j. Therefore, an acoustic image I_{m_p}, as presented in figure 4.4a, can be constructed by concatenating the 200 intensity arrays of a scan as follows:

$$I_{m_p} = \begin{bmatrix} i(\alpha_1,\rho_1) & i(\alpha_1,\rho_2) & \cdots & i(\alpha_1,\rho_{399}) \\ i(\alpha_2,\rho_1) & i(\alpha_2,\rho_2) & \cdots & i(\alpha_2,\rho_{399}) \\ \vdots & \vdots & \cdots & \vdots \\ i(\alpha_{200},\rho_1) & i(\alpha_{200},\rho_2) & \cdots & i(\alpha_{200},\rho_{399}) \end{bmatrix}, \tag{4.1}$$

where $i(\alpha_i,\rho_j)$ represent the intensity value at position j of the intensity array for the α_i scanning angle. Therefore, the obtained image presents a size of 399 x 200 pixels. Because of the sonar's configurations, the initial scanning angle is set to its 180º reference, aligned with the AUV's longitudinal axis. Note that only the bin position is used and not the corresponding detection range since this information is sufficient for image composition.

The composition of an acoustic image considering a cartesian coordinate system presents a more complex problem. Considering the sonar's frame of reference, each intensity measurement bin can be associated with an image pixel $\mathrm{p_{xy}}$, whose position (x,y) may be extrapolated by converting the bin's relative position information into cartesian coordinates through:

$$(\mathrm{x},\mathrm{y}) = \begin{cases} x = x_o + \rho_j \cdot \cos(\alpha_i) \\ y = y_o + \rho_j \cdot \sin(\alpha_i) \end{cases}, \tag{4.2}$$

where (x_o, y_o) represent the central image pixel and referential origin. However, since the data acquisition process is based on environment discretization, point density becomes smaller as the

distance from (x_0, y_0) increases, which pose significant problems to a correct analysis of the image. Therefore, instead of associating each intensity measurement with a single pixel, each bin is used to update several pixels, by considering the sonar beam model presented in International, n.d. Thus, each pixel intensity $I(p_{xy})$ is calculated as

$$I(p_{xy}) = \frac{\sum_n i(\alpha_i, \rho_j)}{n} , \tag{4.3}$$

where n is corresponds to the total number of intensity measurements associated with a given image pixel. The final image has a size of 799 x 799, as the one presented in figure 4.4b.

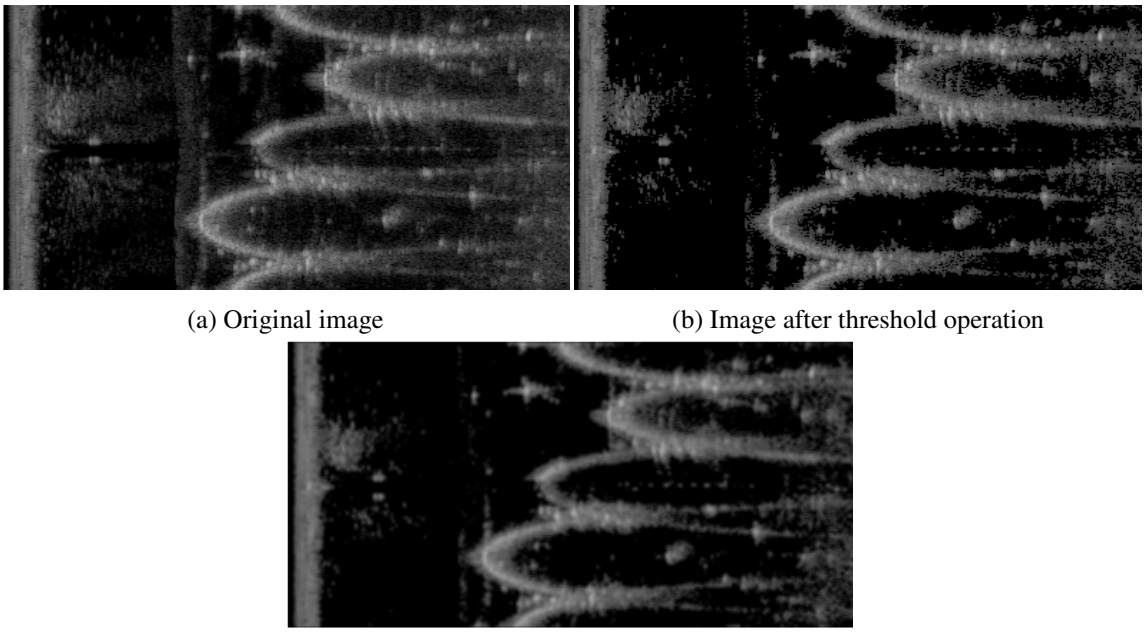
A brief analysis of both composition processes was conducted. As it can be perceived through the previous descriptions, the cartesian image composition algorithm is more computationally demanding and has increased memory storage requirements, mainly due to the resulting image size. Furthermore, as it is shown in figure 4.4, the position from which data is acquired has significant impact on the portrayed features in the resulting image, posing additional challenge to the efficient performance of feature extraction algorithms. This problem is much less noticeable on polar based images. Therefore, for the purpose of this study, the polar coordinate based image composition methodology was used.

4.2.1.3 Image Pre-processing

As it has been mentioned and is perceivable in figure 4.4, the obtained acoustic images present a variety of environmental features, as well as several problems, such as background noise, multi-path interference and multiple echoes. Therefore, an image preparation step was included, in order to decrease the amount of clutter present in each image, which will help preventing the occurrence of false-positives. Note that, in the context of this study, a false-positive refers to any point which is classified as a feature-point but is in fact the result of noise or echoes. The proposed methodology is divided in two sub-stages, the threshold stage and the filtering stage. The intermediate results of each sub-stage are presented on figure 4.5, applied to image 6 of $dataset2$, as an example.

The presence of background noise of low intensity poses as one of the main reasons for the occurrence of false-positives. To minimize its effects, a simple threshold operation is performed, targeting lower intensity points. The results of this operation are exemplified on figure 4.5b. Since, typically, the intensity difference between a point and its neighbourhood points is a key factor for classifying it as an interest point, the effect of contrast enhancement techniques was also studied. However, due to the overall feature ambiguity present in acoustic images, the achieved results were not satisfying. By altering the brightness level of the image, many present features were compromised and the performance of the feature extraction algorithms severely affected.

In the second sub-stage, the image is passed trough an average filter to remove the "salt and pepper" noise generated in the previous stage, as shown in figure 4.5c. The most common solutions in image processing rely on median filters, average filters or Gaussian filters. Median filters are usually the best option for removing this type of noise. However, the amount of blurriness to which the acoustic images are subjected compromises feature integrity. A similar problem is noticeable



(a) Original image (b) Image after threshold operation

(c) Image after filtering operation

Figure 4.5: Intermediate results of image pre-processing routine applied to image 3 of *dataset*2

in the case of Gaussian filters. Average filters provide the most satisfying balance between noise removal capabilities and image blurring. Furthermore, the insertion of some degree of smoothing in the image contributes to an increasing stability and repeatability of feature descriptors (Calonder et al., 2010). Also, image blurring allows feature extractors to focus on larger keypoints (Dinuka and Bandara, 2017).

4.2.1.4 Feature Extraction

After being pre-processed, in this stage, the image is passed through the feature extraction algorithms selected, in order to extract interest points and their respective descriptors. Depending on the algorithm used, the type and overall shape of the feature varies. The main objective lies on finding locally invariant points, so that, even in the presence of strong image transformations, environmental features may be extracted and processed for navigation purposes.

As it has been presented in section 2.2, each of the studied feature detectors makes use of its own metric, that henceforward will be referred to as detection score, for classifying a point as an interest point or not.

However, contrary to what would be expected, there is not a clear relation between the detection threshold used and the number of false-positives that occur. This is due to the characteristics of acoustic images, since, despite being generated by acoustic echoes or noise, false-positive points are very similar to interest points generated by environmental features, fulfilling the detection criteria. For this reason, the selected feature extractors' accuracy is crucial.

Therefore, in order to achieve uniformity across all the methods compared, the detection threshold of each one was set as to allow the detection of at least 50 interest points in each image

of both datasets. Having collected the desired interest points, the corresponding descriptors are calculated, stored and passed to the next stage.

4.2.1.5 Feature Matching

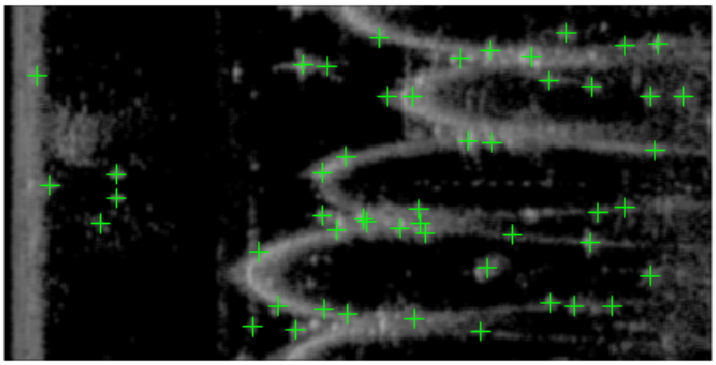
In this last step of the image processing procedure, feature matching is performed. For the purpose of image based navigation, this step presents itself as of key importance for the overall success of the employed methodology. Firstly, it is through this step that data which enables the application of visual odometry techniques is extrapolated. Secondly, since false-positive points are extremely volatile, it is expected that a vast majority of these are not possible to be matched throughout a collection of scans. Therefore, the matching step acts as a filtering step in the current procedure, improving the confidence on the used data.

In this step, only the interest points detected in each pair of consecutive images are taken into account. For SURF and SURF-Harris, the sum of squared differences is used to perform the required comparison, since their respective descriptors are string based. In the case of BRISK and ORB, the feature comparison is performed through Hamming distance computation. These metrics compose our matching score and allow the application of the nearest neighbour search algorithm as a matching strategy. Furthermore, a matching threshold is employed in order to remove any possible outliers, this is, any incorrect match. A feature is only matched if its closest neighbour matching score is lower than this threshold. Otherwise, the feature is not matched. For string based descriptors, the applied matching threshold was set to 5.0, whereas in the case of binary descriptors, the corresponding threshold was set to 20.0. All in all, the robustness of the feature descriptor used is of utmost importance for the successful navigation of the AUV.

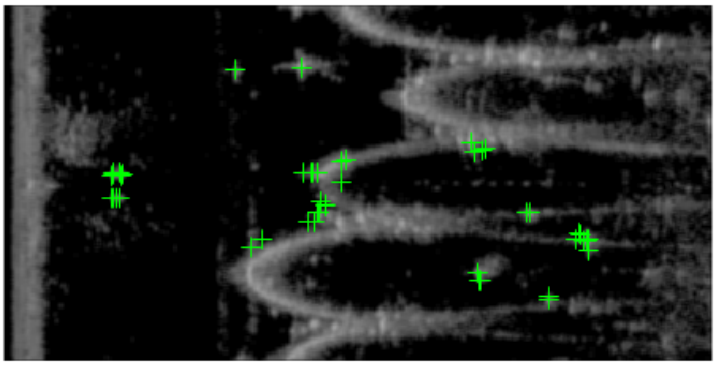
4.3 Results and Discussion

As discussed in section 4.2.1, the pre-processed acoustic images are passed through the SURF, ORB, BRISK and SURF-Harris feature extractors. Figure 4.6 presents an example of the obtained results for each of these algorithms.

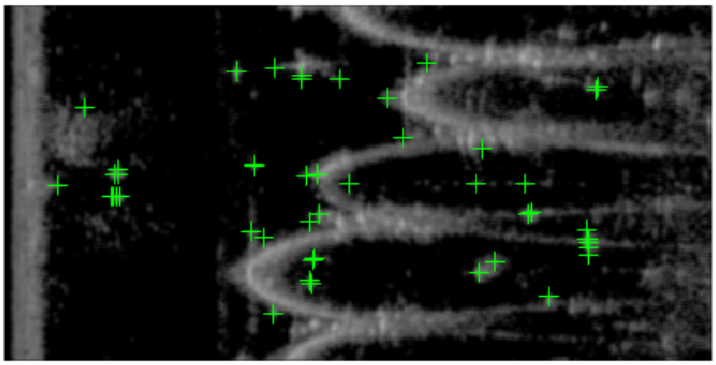
An initial qualitative assessment is ought to be made regarding these results. As it can be perceived by an analysis of figure 4.6, it is clear that the distribution of the detected interest points varies according to the employed algorithm. SURF points are typically more scattered throughout the image. This is mainly due to the greater variety of feature shapes (blobs, edges, corners, ...) that SURF is intended to be able to detect, along with its scale-space analysis methodology. SURF-Harris presents a similar behaviour but it is notorious a tendency for this algorithm to generate some point clusters in the presence of strong isolated features. In the case of ORB and BRISK, the propensity to generating point clusters is even more notorious. Feature distribution takes key importance especially during the position estimation procedure. The data that can be extracted from points located very close to each other ends up being quite similar and not adding any real value to our estimation.



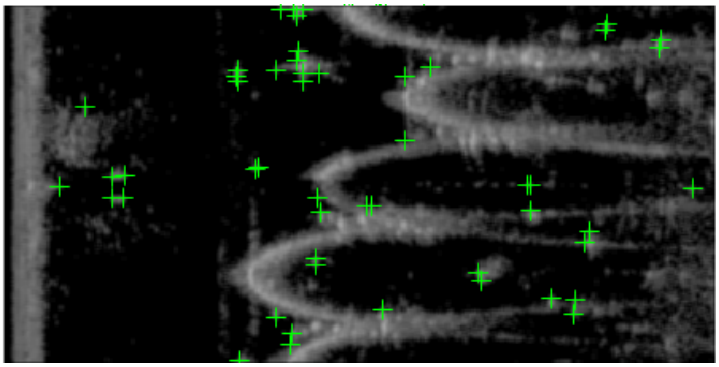
(a) Interest point detection with SURF algorithm



(b) Interest point detection with ORB algorithm



(c) Interest point detection with BRISK algorithm



(d) Interest point detection with SURF-Harris algorithm

Figure 4.6: Interest point detected in image 3 of dataset 2 with the algorithms under analysis

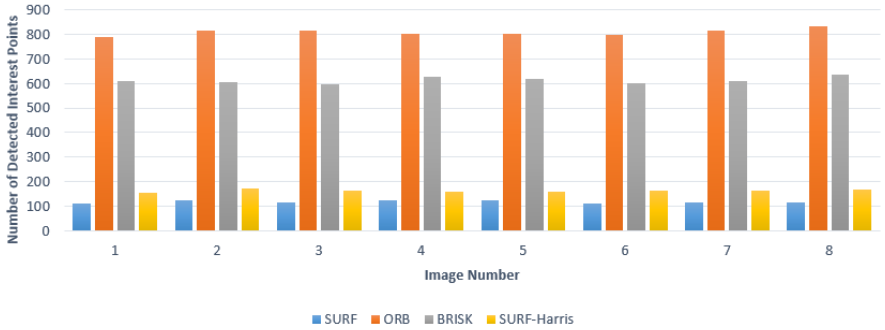


Figure 4.7: Total number of detected interest points by each algorithm in each image of *dataset*1

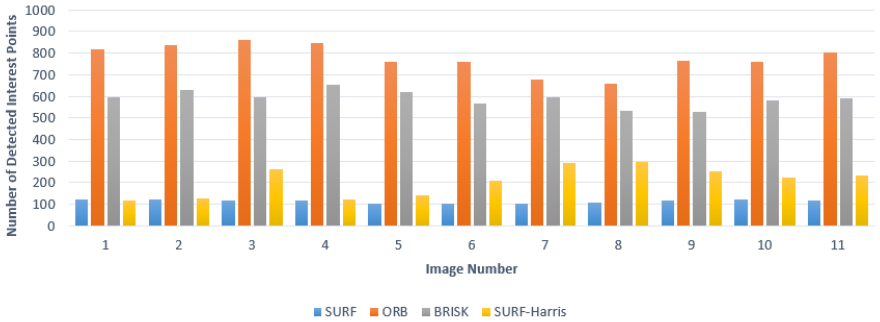


Figure 4.8: Total number of detected interest points by each algorithm in each image of *dataset*2

For an initial feature detector analysis, the total number of features detected through each method in each image was recorded. The results are presented in figures 4.7 and 4.8, where it is possible to observe that ORB and Brisk are able to extract the most features in every single image, while SURF and SURF-Harris are on the lower side of feature detection. It is also important to note that, as it is shown in the presented results, the movement of the AUV has a small impact on the algorithms' performance.

In order to better assess each detector's accuracy and corresponding descriptor robustness, only the 50 best interest points detected, according to the associated detection score, in each image were taken into account. This measure was taken in order to guarantee uniformity across the performed evaluation. The chosen value was determined experimentally and found to be sufficient for a sound representation of the algorithms' performance.

Regarding detection accuracy, the best 50 interest points in each image were carefully analysed and the number of false-positives among these was collected. As it can be observed in figures 4.9 and 4.10, ORB and SURF perform very similarly, producing less false-positives than BRISK or SURF-Harris. Furthermore, it is possible to conclude that the motion of the AUV has no significant effect on the algorithms' accuracy. Although, as shown by the results for images 9, 10 and 11 of *dataset*2, presented in figure 4.10, some viewpoints are less susceptible to the occurrence of false-positives, due to the characteristics of the testing environment.

For the assessment of descriptor robustness, once again, only the 50 best interest points detected in each image where taken into account. As detailed in section 4.2.1, during the matching procedure interest points' descriptors collected from consecutive images are compared. The pro-

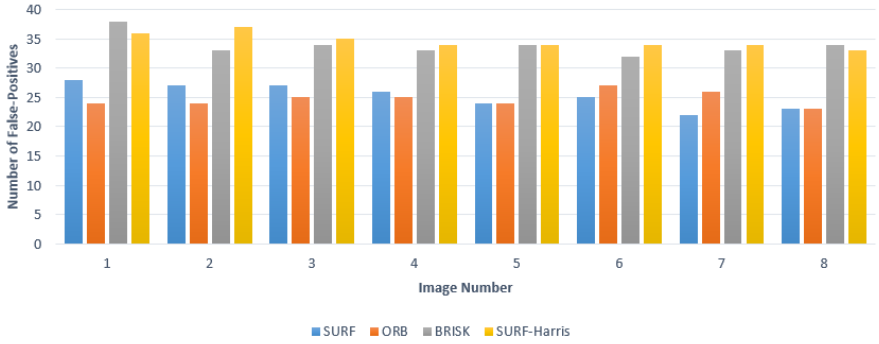


Figure 4.9: Number of false-positives detected amongst the 50 best interest points detected in each image of *dataset*1

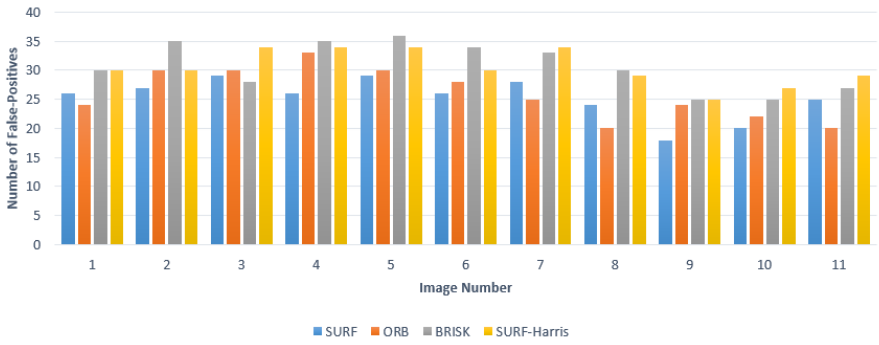


Figure 4.10: Number of false-positives detected amongst the 50 best interest points detected in each image of *dataset*2

	SURF		ORB		BRISK		Harris-SURF	
Image Pair	Matches	Outliers	Matches	Outliers	Matches	Outliers	Matches	Outliers
1-2	41	1	38	1	32	0	29	0
2-3	43	0	45	0	40	0	35	0
3-4	43	0	42	0	44	0	34	1
4-5	41	0	43	0	43	0	38	0
5-6	47	0	43	0	43	0	37	1
6-7	43	0	41	0	44	1	33	0
7-8	44	0	42	0	38	0	38	0

Table 4.1: Number of matches performed and corresponding number of outliers for *dataset*1

cedure was employed to both datasets and the number of matches per image pair, as well as the number of outliers generated was tabulated and are presented in tables 4.1 and 4.2. Sample results are presented in figure 4.11 An important conclusion can be made from these results. The movement of the AUV has a great impact on the number of matches performed. It is possible to observe significant decreases in the number of matched features as well as a rise in the number of outliers from *dataset*1 results to *dataset*2 results.

Further observations can be made by an analysis of figures 4.12 and 4.13. Despite presenting similar performances for *dataset*1, the algorithms behave differently in the case of *dataset*2. It is

	SURF		ORB		BRISK		Harris-SURF	
Image Pair	Matches	Outliers	Matches	Outliers	Matches	Outliers	Matches	Outliers
1-2	29	3	23	1	13	1	20	3
2-3	24	2	20	1	16	1	21	3
3-4	18	8	9	4	2	1	21	11
4-5	16	9	4	2	2	0	23	13
5-6	14	5	7	2	2	1	15	8
6-7	15	6	2	1	2	2	11	8
7-8	11	4	4	0	4	1	15	9
8-9	17	7	2	1	4	1	14	13
9-10	17	5	5	1	3	2	14	9
10-11	16	5	8	2	7	1	15	14

Table 4.2: Number of matches performed and corresponding number of outliers for *dataset*2

clear that both SURF and SURF-Harris are able to extract far more features in every image than both BRISK and ORB. Furthermore, a strong decrease in matched features is observable from image pair 2-3 to image pair 3-4. The reason for this is associated with the fact that image 2 is acquired while the AUV is just starting its movement action, so feature deformation affecting this image is not so intense. This decrease is far more notorious in the case of BRISK and ORB, whose descriptors are binary, which indicates a greater inadequacy of such descriptors to the scenario in study. However, it is also possible to observe that both SURF and SURF-Harris produce higher numbers of outliers than BRISK and ORB.

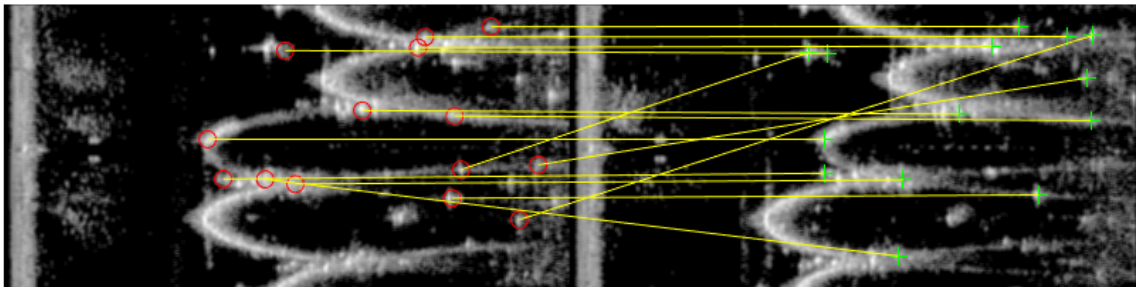
	SURF			ORB			BRISK			Harris-SURF		
Image Pair	Detection Time (s)	Matching Time (s)	Total Computation Time (s)	Detection Time (s)	Matching Time (s)	Total Computation Time (s)	Detection Time (s)	Matching Time (s)	Total Computation Time (s)	Detection Time (s)	Matching Time (s)	Total Computation Time (s)
1-2	0,1406	0,0049	0,1455	0,0203	0,0036	0,0239	0,5707	0,0035	0,5742	0,0329	0,0038	0,0367
2-3	0,1427	0,0034	0,1461	0,0185	0,0036	0,0220	0,5811	0,0032	0,5843	0,0453	0,0543	0,0996
3-4	0,1561	0,0036	0,1597	0,0315	0,0055	0,0371	0,5899	0,0023	0,5922	0,0823	0,0073	0,0895
4-5	0,1665	0,0032	0,1697	0,0339	0,0030	0,0369	0,6866	0,0024	0,6890	0,1032	0,0064	0,1096
5-6	0,1720	0,0036	0,1756	0,0198	0,0056	0,0254	0,8010	0,0023	0,8033	0,1319	0,0441	0,1760
6-7	0,1629	0,0029	0,1657	0,0173	0,0033	0,0206	0,8465	0,0026	0,8492	0,1132	0,0567	0,1699
7-8	0,1549	0,0035	0,1583	0,0213	0,0029	0,0242	0,8124	0,0023	0,8147	0,0582	0,0558	0,1141

Table 4.3: Computation time measurements results performed for each algorithm for *dataset*1

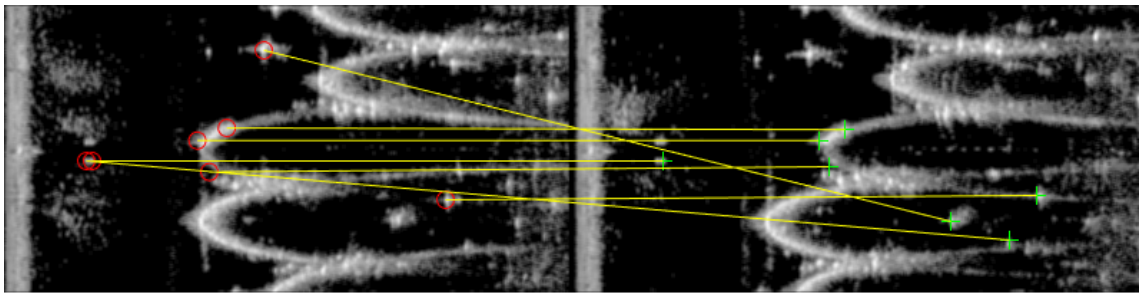
	SURF			ORB			BRISK			Harris-SURF		
Image Pair	Detection Time (s)	Matching Time (s)	Total Computation Time (s)	Detection Time (s)	Matching Time (s)	Total Computation Time (s)	Detection Time (s)	Matching Time (s)	Total Computation Time (s)	Detection Time (s)	Matching Time (s)	Total Computation Time (s)
1-2	0,1409	0,0029	0,1438	0,0154	0,0032	0,0186	0,5433	0,0186	0,5619	0,0938	0,0538	0,1476
2-3	0,1427	0,0029	0,1456	0,0122	0,0042	0,0164	0,5548	0,0025	0,5572	0,1204	0,0051	0,1255
3-4	0,1486	0,0031	0,1517	0,0121	0,0027	0,0149	0,5756	0,0031	0,5786	0,0734	0,0519	0,1252
4-5	0,1555	0,0026	0,1581	0,0169	0,0035	0,0204	0,5838	0,0294	0,6132	0,0778	0,0516	0,1295
5-6	0,1596	0,0052	0,1648	0,0196	0,0025	0,0221	0,5822	0,0114	0,5935	0,0791	0,0060	0,0851
6-7	0,1466	0,0029	0,1495	0,0200	0,0024	0,0224	0,5740	0,0023	0,5762	0,0676	0,0051	0,0727
7-8	0,1533	0,0032	0,1564	0,0176	0,0022	0,0198	0,5613	0,0026	0,5638	0,0680	0,0044	0,0724
8-9	0,1665	0,0036	0,1701	0,0153	0,0026	0,0180	0,5659	0,0028	0,5687	0,0505	0,0035	0,0540
9-10	0,1969	0,0039	0,2008	0,0179	0,0025	0,0205	0,5760	0,0026	0,5786	0,0596	0,0037	0,0634
10-11	0,2087	0,0032	0,2120	0,0191	0,0043	0,0234	0,6390	0,0023	0,6413	0,0728	0,0076	0,0803

Table 4.4: Computation time measurements results performed for each algorithm for *dataset*2

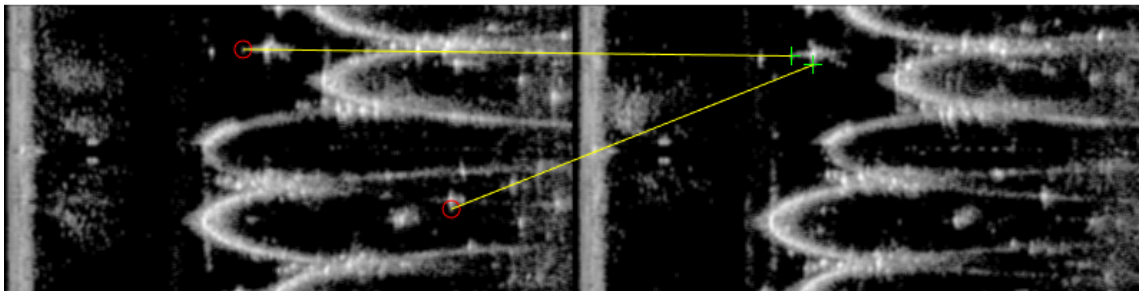
Due to the real time requirements of navigation procedures, computation time measurements regarding feature extraction and feature matching routines were performed. These are presented



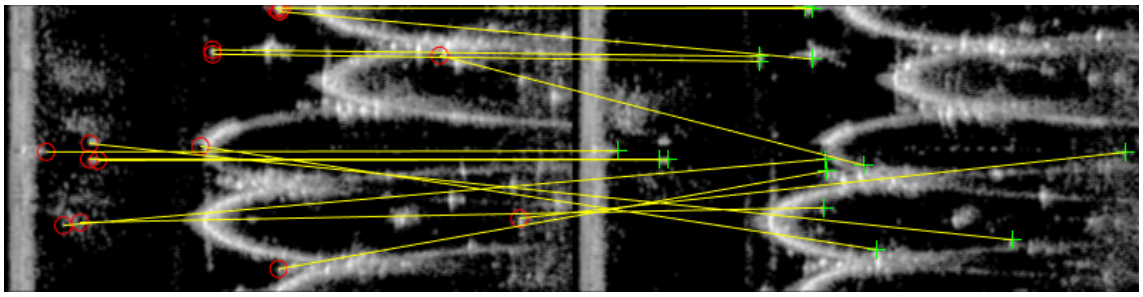
(a) Feature matching with SURF algorithm



(b) Feature matching with ORB algorithm



(c) Feature matching with BRISK algorithm



(d) Feature matching with SURF-Harris algorithm

Figure 4.11: Feature points matched in image 3 of *dataset*2 with the algorithms under analysis

in tables 4.3 and 4.4. Note that, since these time measurements are presented for each consecutive image pair, the tabulated feature extraction computation time is the result of the sum of the feature extraction computation time of each image of the pair. These results reveal that ORB is the fastest algorithm, whereas BRISK is the slowest. SURF-Harris presents shorter computation times than SURF. These differences are mainly due to each algorithm's feature extraction computation speed. It is important to note that the time measurements obtained are heavily dependent on the tools and computational capacity available.

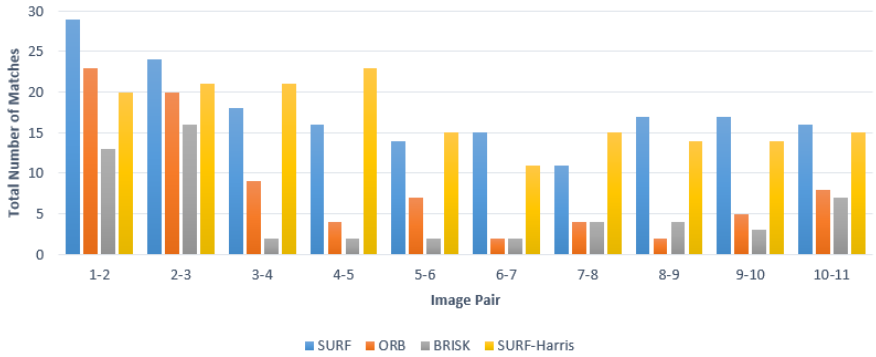


Figure 4.12: Number of matches performed for each image pair of *dataset*2

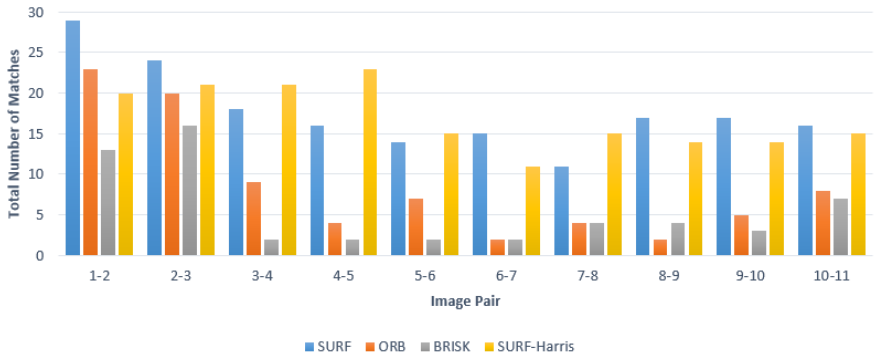


Figure 4.13: Number of outliers detected amongst the matched features in each image pair of *dataset*2

Therefore, taking into account all the different aspects analysed, it is possible to conclude that SURF is the algorithm best suited for the task at hands. Its required computation time, despite not being the shortest, does not compromise real time applicability. It has proven to be one of the most accurate algorithms and the one that is able to return the most matches. Furthermore, the presented interest point distribution is ideal for localization purposes. Moreover, due to the nature of its descriptors, the incorporation of additional information into it, that will allow for a decrease in the number of registered outliers, can be performed more readily.

4.4 Summary

In this chapter, a comprehensive performance evaluation of the SURF, ORB, SURF-Harris and BRISK feature extraction algorithms has been presented. The results obtained indicate that the characteristics of acoustic images put significant challenges to all these algorithms. Nevertheless, SURF and ORB have shown the most satisfactory behaviours under these constraints. However, SURF presents valuable characteristics regarding the distribution of detected features and type of feature it is able to detect. Therefore, it has been considered the most promising methodology for the main objective of this work.

Furthermore, it has been shown that the matching procedure, especially when acoustic images are acquired with the AUV in movement, presents significant problems, due to the reduced number of matches performed and high outlier ratios shown. It is then possible to conclude that the current information portrayed by the studied feature descriptors is insufficient for acoustic images matching. Therefore, further pertinent data must be incorporated into SURF's descriptor. This necessity was also pertinent in the choice of SURF as the solution to use throughout the remainder of this work, since, due to its non-binary nature, it will allow a simpler integration of new information.

Chapter 5

Feature Matching

5.1 Introduction

The results presented in the previous chapter prove that the feature matching procedure is a challenging operation under the several constraints presented by acoustic images. In truth, the performance of the feature descriptor and, therefore, of the matching procedure is highly related to the combination detector-descriptor used (Dinuka and Bandara, 2017). Due to the nature of the acquisition process, the acoustic images are highly susceptible to viewpoint variations or environment changes such that, as portrayed by the results obtained for $dataset1$, even features extracted from successive images of the same scenario, taken from the same viewpoint, may not be matched with any other. Furthermore, the lack of unambiguous features in the underwater environment lead to a number of incorrect matches between different interest points.

Due to the importance of the matching step on the navigation operation, it matters to improve the robustness of SURF's spectra descriptor, as well as the effectiveness of the matching procedure. In this chapter, a solution is proposed, based on extending the existing descriptor, in order to incorporate further information regarding the interest point. The modification of the feature descriptor implied a redesigning of the matching procedure, so as to accommodate for this changes. However, this initial solution has proved to be flawed and insufficient to overcome the problems detected during the analysis presented in the last chapter. Therefore, further development of this new procedure was carried out and is presented hereafter, along with representative results of its performance.

5.2 SURF's Spectra Descriptor

Although a brief summary of the SURF algorithm has already been presented, a more detailed overview of its descriptor is in order, so as to better understand its building process. As it has been stated, SURF's descriptor building process is divided into two steps, the orientation assignment step and the descriptor extraction step.

In the first step, a reproducible orientation is identified for each interest point. For that purpose, for each interest point, the Haar-wavelet responses in the x and y directions are calculated taking into account a circular neighbourhood of radius $6s$, where s represents the detection scale, around the point. Due to the dimensions wavelets may reach, this operation is performed making use of integral images. The resulting wavelet responses are then weighted with a Gaussian, of $\sigma = 2.5s$, centered at the interest point. Finally, the assigned orientation is estimated by the sum of the weighted responses within a sliding sector window of size $\frac{\pi}{3}$.

Having calculated an orientation for each interest point, the description extraction procedure starts by constructing a square region centered around the point and aligned with its orientation. This region is then split into smaller 4 x 4 sub-regions. For each sub-region, its Haar-wavelet responses in the horizontal direction, d_x, and vertical direction, d_y, are calculated and weighted with a Gaussian of $\sigma = 3.3s$. These responses d_x and d_y are then summed up over each sub-region, along with its absolute values, $|d_x|$ and $|d_y|$. Hence, each sub-region is associated with a descriptor vector v_i, such that

$$v_i = \begin{bmatrix} \sum d_x & \sum d_y & \sum |d_x| & \sum |d_y| \end{bmatrix}, \tag{5.1}$$

resulting in a final descriptor vector D, for each of the K detected interest points, of length 64, that encompasses all the singular sub-region descriptors, as follows:

$$D_k = \begin{bmatrix} v_1 & v_2 & \cdots & v_{64} \end{bmatrix}. \tag{5.2}$$

Invariance to scale is achieved by turning the final descriptor into an unit vector.

5.3 Proposed Solution

SURF's Spectra Descriptor aims at describing an interest point by capturing and portraying its intensity structure. It makes use of relative strengths and orientations of gradients in order to reduce the effect of photometric changes. But, as it has been shown by the analysis presented in chapter 4, this methodology is not sufficiently robust when applied to acoustic images. Furthermore, image distortion resulting from viewpoint changes presents additional challenges to the selected methodology. This statement relies on the fact that, in several cases, the same interest points are indeed detected in consecutive images but not every point is correctly matched.

Taking this difficulties into consideration, it is reasonable to assume that, during the matching procedure, restricting the number of candidate interest points that will be compared with a given initial interest point can result in an improvement to the method's performance. Since, between consecutive images, it is not expected a feature point's image localization to suffer great variations, the intended goal can be achieved by incorporating such information into the existing descriptor, which will take the following form:

$$D_k = \begin{bmatrix} v_1 & v_2 & \cdots & v_{64} & x_k & y_k \end{bmatrix}, \tag{5.3}$$

where x_k and y_k represent the feature point's image position coordinates. Thus, the resulting matching score will reflect the distance between the candidate point and the initial feature point, therefore allowing the rejection of candidates far form an expected detection region, depending on the employed matching threshold. Note that, with the AUV in movement, since the analysed acoustic images are generated based on its frame of reference, variations on features' relative localization are to be expected and accounted for when defining this threshold.

However, while introducing a dependency from feature localization into the matching procedure, it is important to stress that the movement of the AUV will impact features' image localization differently, according to their respective detection range. This is mainly due to a loss of detection resolution with the distance to the AUV. Therefore, features detected closer to the vehicle will suffer greater variations in their image position than the ones detected further away. So, it is possible to conclude that descriptor comparison must no longer be performed making use of a fixed matching threshold, due to the feature descriptor being extended. Therefore, an adaptive threshold methodology was developed.

The loss of detection resolution with increasing detection range implies that, for feature points at lower detection ranges, the matching threshold should be higher than the one employed to feature points at higher detection ranges, since higher relative positional drifts are expected for the firsts, with the AUV in movement. With these constraints taken into account, a threshold function is proposed, which allows determining a matching threshold according to the feature point detection range, presented below:

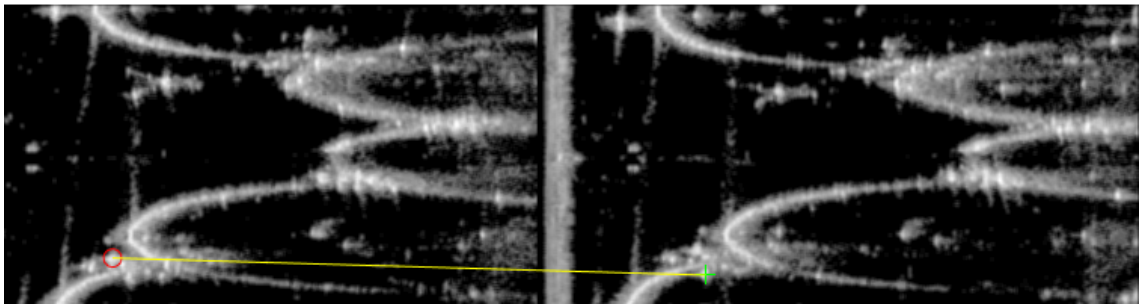
$$f_{threshold}(r) = b + d \cdot e^{\frac{-r}{N_b}} \,, \tag{5.4}$$

where r the feature detection range, b represents an offset value that will act as the matching threshold when the vehicle is stopped, N_b the total number of intensity bins, d an AUV's velocity related gain intended to adjust the matching threshold according to it and calculated as follows:

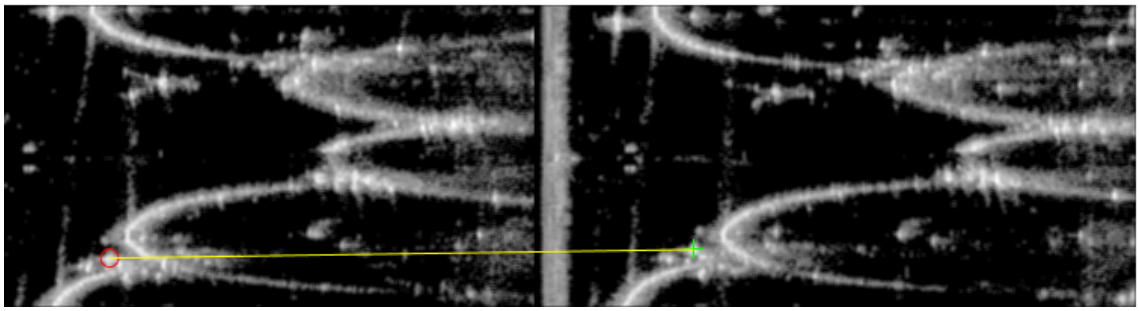
$$d = k \cdot \sqrt{v_t^2 + w^2} \,, \tag{5.5}$$

where v_t represents the AUV's translation velocity, w the AUV's rotation velocity and k a simple correction gain defined experimentally. Thus, the matching procedure still relies on a nearest neighbour search algorithm but the matching score is now obtained through the computation of the Euclidean Distance for feature descriptor comparison.

An initial evaluation of this first solution has shown a significant improvement in the number of performed matches. However, despite restricting the number of feasible candidate points for a correct match, through the inclusion of feature points' image position information into the feature descriptor, it is still notorious a number of incorrect associations. As exemplified in figure 5.1a, extending the feature descriptor allows a correct differentiation between features significantly distant, but does not hold sufficient strength to enable the distinction between close similar features. Therefore, a new filtering layer was inserted into the proposed solution, in order to make up for the shortcomings of the initial solution.



(a) Matching result example for initial proposed solution



(b) Matching result example for final proposed solution

Figure 5.1: Example of the correction introduced by a new filtering layer in the matching process

This new filtering layer is essentially based on the analysis of the original feature descriptor, since it only focus on the feature point's intensity structure. Thus, taking into account the already developed methodology, after a possible match is performed, a new feature descriptor comparison is performed between the matched points but, this time, making use of the only the original SURF's Spectra Descriptor, corresponding to the first 64 bytes of the extended descriptor. Therefore, this additional feature comparison, again based on the computation of the Euclidean Distance between the two descriptors, whose result constitutes the deformation score, will focus solely on the feature point's intensity structure. If this deformation score is below a predefined threshold, the deformation threshold, the performed matched is kept. Otherwise, the second best possible match, according to the matching score, is analysed, until a candidate point that fulfills both criteria is found. If no candidate point is found, the initial point is not matched. This procedure is described in figure 5.2.

So, by introducing this new sub-step in the matching procedure, the ability to better differentiate close interest points is achieved, as exemplified in figure 5.1b. Further comparison results between the initial and final proposed solutions are presented in section 5.4.

5.4 Results and Discussion

For a more comprehensive analysis of the proposed solution performance, both the initial version and the final version were employed to $dataset1$ and $dataset2$ and compared with the results obtained for the original matching procedure. To guarantee uniformity in the comparison, the

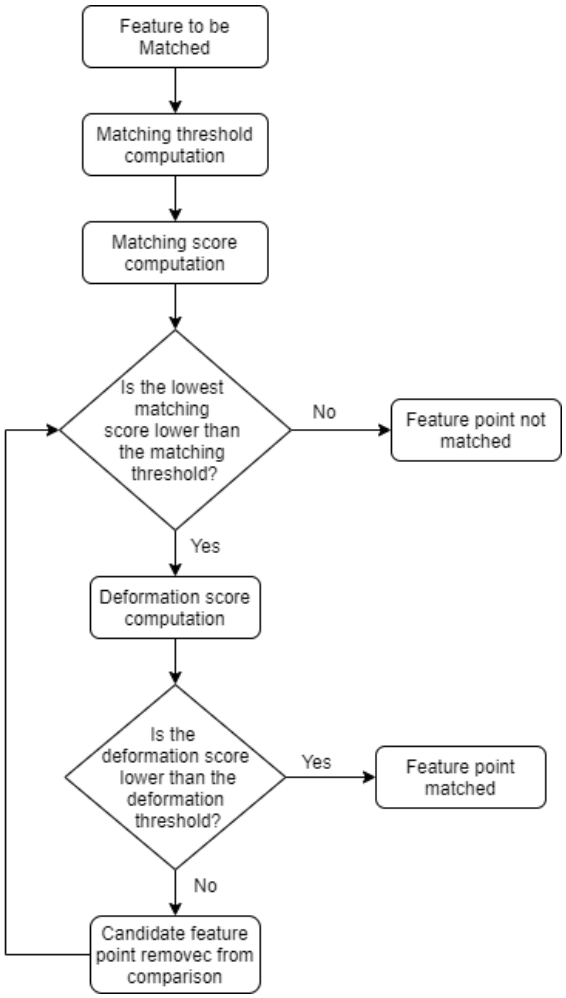
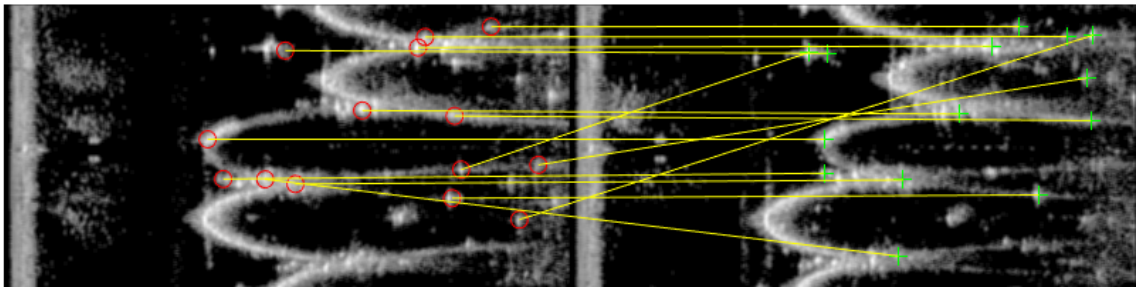


Figure 5.2: The final proposed solution procedure

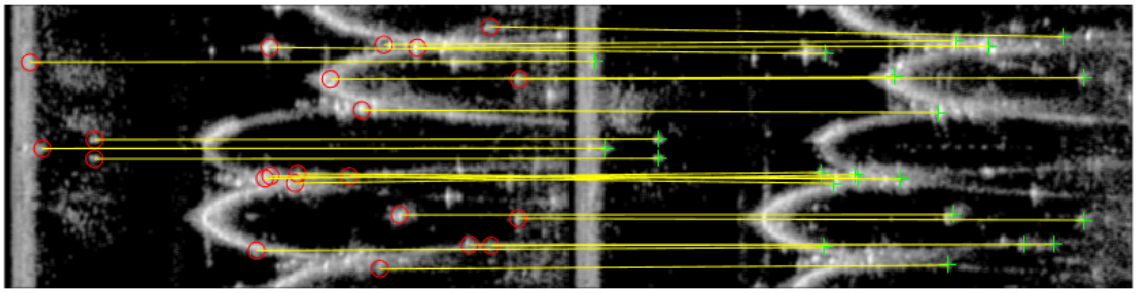
same methodology presented in figure 4.2, back in chapter 4, was employed. Once again, only the 50 best interest points detected, according to the associated detection score, were taken into account for this analysis. The number of performed matches, the number of detected outliers and the required computation time for the matching procedure were collected and are tabulated in table 5.1 and table 5.2.

Due to the extension of the feature descriptor, the employed matching threshold is required to account for the corresponding increase in matching scores. Consequently, and due to the implementation of a dynamic matching threshold methodology, the number of matched features is higher for both the initial and final proposed solutions. This effect is even more notorious in image regions associated with lower detection ranges, as depicted in figure 5.3, since these regions are associated to higher matching thresholds. Note that, for *dataset*1, due to the absence of motion, the matching threshold assumes a fixed value and, therefore, the difference between the obtained results is not so accentuated.

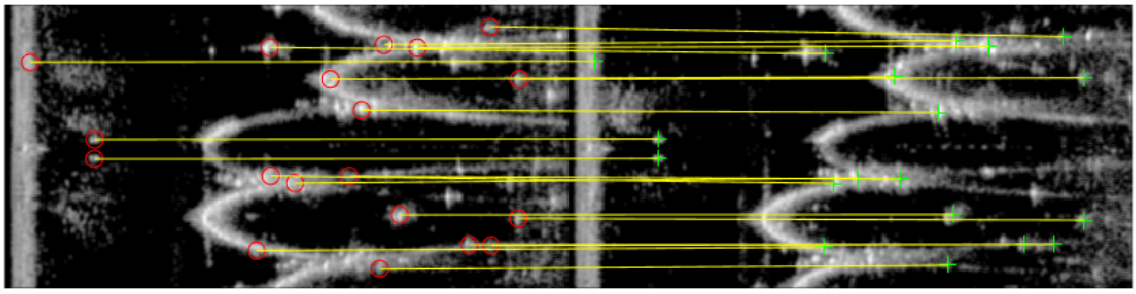
However, as it has been stated in the previous section, the initial proposed solution is insufficient to distinguish feature points close to each other and thus, a significant high number of



(a) Feature matching results with original procedure



(b) Feature matching results with initial solution



(c) Feature matching results with final solution

Figure 5.3: Feature point matching results for image 3 of *dataset*2

	Original Matching Procedure			Initial Proposed Solution			Final Proposed Solution		
Image Pair	Matches	Outliers	Matching Time (s)	Matches	Outliers	Matching Time (s)	Matches	Outliers	Matching Time (s)
1-2	41	1	0,0049	45	0	0,0030	42	0	0,0044
2-3	43	0	0,0034	46	0	0,0055	45	0	0,0055
3-4	43	0	0,0036	46	2	0,0026	44	0	0,0026
4-5	41	0	0,0032	46	0	0,0026	45	0	0,0028
5-6	47	0	0,0036	47	0	0,0026	47	0	0,0024
6-7	43	0	0,0029	48	0	0,0032	47	0	0,0035
7-8	44	0	0,0035	48	0	0,0033	47	0	0,0025

Table 5.1: Number of matches performed and corresponding number of outliers and computation time required for *dataset*1

outliers is registered with this procedure. This shortcoming is especially worrying when the AUV is in motion, since, as depicted in figure 5.4, this initial solution produces the highest number of outliers for the majority of the testing image pairs. Furthermore, for the case of image pair 9-10,

	Original Matching Procedure			Initial Proposed Solution			Final Proposed Solution		
Image Pair	Matches	Outliers	Matching Time (s)	Matches	Outliers	Matching Time (s)	Matches	Outliers	Matching Time (s)
1-2	29	3	0,0029	36	1	0,0024	30	0	0,0049
2-3	24	2	0,0029	39	2	0,0043	35	1	0,0051
3-4	18	8	0,0031	33	5	0,0028	29	3	0,0030
4-5	16	9	0,0026	21	9	0,0030	16	3	0,0029
5-6	14	5	0,0052	26	9	0,0026	17	4	0,0031
6-7	15	6	0,0029	23	7	0,0023	14	4	0,0035
7-8	11	4	0,0032	28	9	0,0025	22	4	0,0026
8-9	17	7	0,0036	20	8	0,0026	15	4	0,0024
9-10	17	5	0,0039	27	9	0,0026	19	4	0,0021
10-11	16	5	0,0032	29	8	0,0026	23	1	0,0034

Table 5.2: Number of matches performed and corresponding number of outliers and computation time required for *dataset*2

this procedure results in a higher ratio of detected outliers than the one registered for the original image, which accentuates the flaws already outlined.

Therefore, as also described in section 5.3, an additional filtering layer has been introduced into the proposed solution. So, as it can be observed on both tables, despite the number of matched features suffers a reduction, the same happens for the case of the number of detected outliers. Further observations can be made by analysing figure 5.5, where it is clear that the final solution achieves the lowest outlier ratios for every single image pair of *dataset*2, the most challenging scenario depicted.

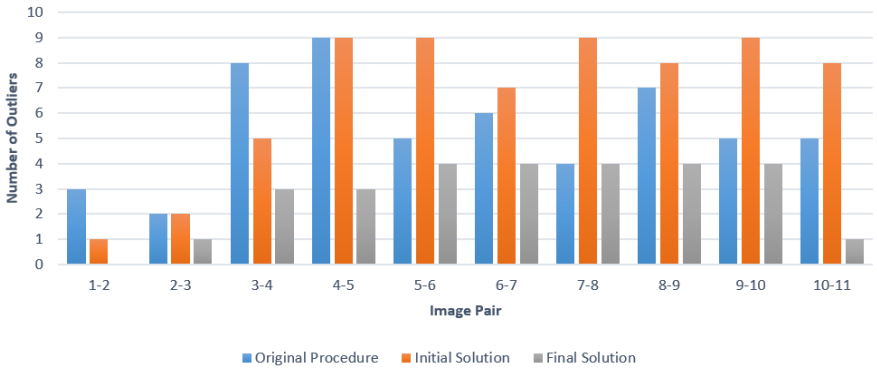


Figure 5.4: Number of outliers detected per image pair of *dataset*2

It is also important to exalt that, despite both the initial and final proposed solutions being fairly more complex than the original matching procedure, the results acquired for the required computation time are very similar. If the original solution fulfils the requirements for real time applications, depending on the tools and computation capacity available, so should the proposed solution.

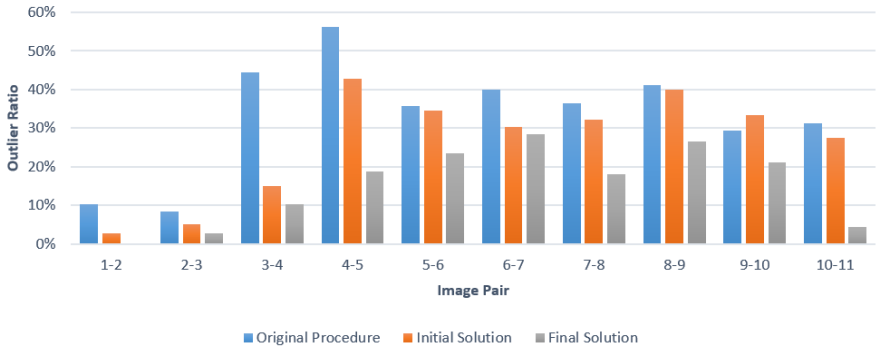


Figure 5.5: Outlier ratio per image pair of *dataset*2

5.5 Summary

This chapter presented the development of a solution for the matching procedure, based on extending the original SURF's Spectra Descriptor and redesigning the actual process in order to account for this changes. The new descriptor includes the feature location information, in an attempt to further restrict the number of possible matches. However, as it has been shown, performing the match association solely based on the computation of the matching score through this extended descriptor proved to be inefficient to surpass the problems presented by the original procedure. Therefore, a new filtering layer has been introduced, which allows a better trade-off between the number of matched feature points and the number of consequent outliers and thus, achieving the intended goal for this part of the work.

It matters to point out that high matching ratios are never to be expected with the AUV in movement. This is due to possibility of occurrence of false-positives, caused by noise or acoustic echoes, which are sensitive to viewpoint variations. This aspect further accentuates the importance of the matching procedure, as these false detections are expected to disappear from one scan to another and, so, being disregarded during this procedure.

Chapter 6

Simultaneous Localization and Mapping

6.1 Introduction

The work developed and presented throughout chapters 4 and 5 had the purpose of studying and providing a viable solution for landmark detection. This landmarks, that through the implementation of SURF's feature extractor are represent by feature points, provide essential information for the localization of the AUV and, additionally, to construct a map of the environment the vehicle navigates. This objective is achieved through the implementation of an Extended Kalman Filter Simultaneous Localization and Mapping algorithm.

Therefore, in this chapter, the development of the localization and mapping module is detailed, taking into account the general algorithm for EKF SLAM presented in chapter 2. Here, further details regarding the prediction cycle and the update cycle are introduced, with especial focus on the implemented motion model and measurement model.

The study of acoustic images made it possible to observe that, due to the conditions of the testing environment (a closed and structured environment), a number of acoustic echoes and reflections on the AUV's own body were consistently portrayed. Therefore, in order to minimise the effect of these problems on the final pose estimate and map built, the image pre-processing routine has suffered further development. Furthermore, it was also noticed that the structure of the testing environment has an impact on the distribution of the detected feature points. The analysis of previous results has shown that a great number of features points are typically detected in image regions corresponding to the water tank's corners. Thus, the study of the application of clustering techniques, that will enable reducing the number of features fed to the state estimator, is presented in this chapter.

6.2 System Integration

As mentioned throughout this document, the objective of the analysis of the acquired acoustic images is to detect possible landmarks in the surroundings of the AUV, feeding this information to the developed localization and mapping module, as represented by the block diagram in figure 6.1.

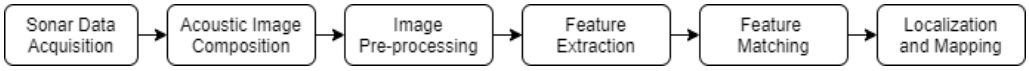
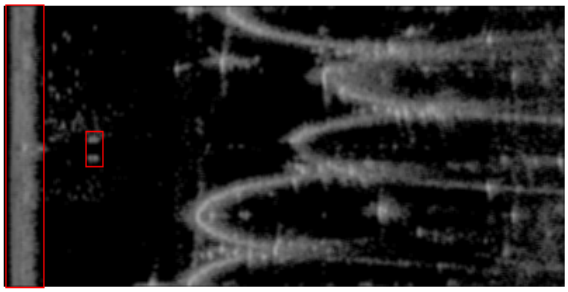
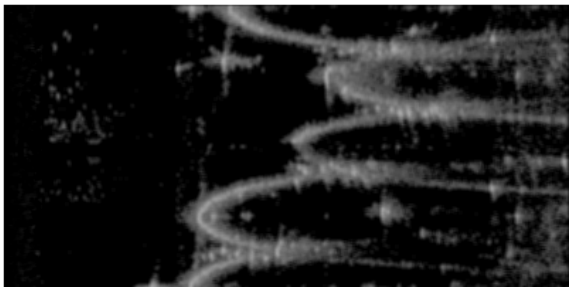


Figure 6.1: Developed solution module sequence

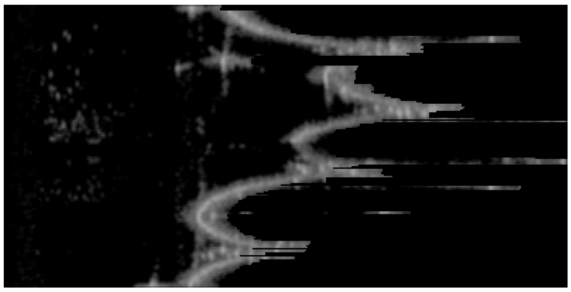
It comes without saying that the performance of the overall procedure is conditioned by the output of each individual module, especially concerning the matched feature points that are fed to the localization and mapping module. Taking this premise into consideration, it is possible to conclude that the image pre-processing procedure presented in chapter 4 proves to be insufficient for the intended objective. This is due to the characteristics of the testing environment, a closed structure, which promotes the occurrence of more acoustic echoes than it would be expected in an open area, resulting in an increased number of false detections. Furthermore, due to the shape of the emitted acoustic beam and the sonar's mounting position on the AUV, the image also portrays acoustic reflections on the vehicle's own body. Therefore, additional features were added to this module, in order to tackle the enumerated problems.



(a) Original image with AUV body reflections marked with red rectangles

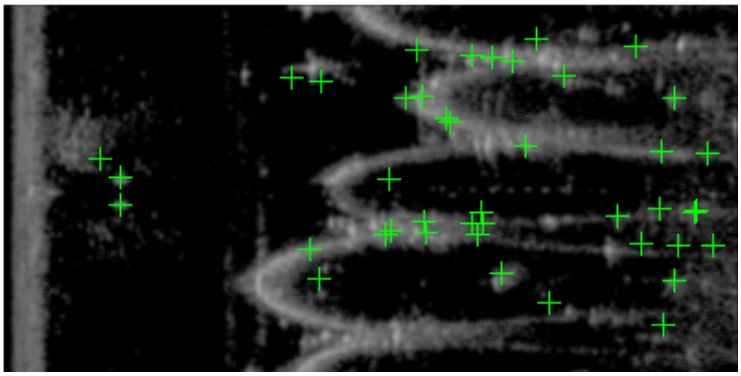


(b) Image after background subtraction

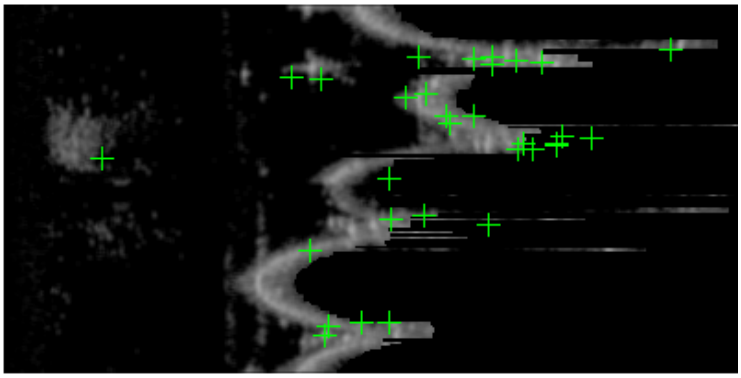


(c) Image after acoustic echo removal

Figure 6.2: Intermediate results of the final image pre-processing routine applied to image 6 of *dataset*2



(a) Initial pre-processing procedure



(b) Final pre-processing procedure (background subtraction plus 1st and 2nd echoes)

Figure 6.3: Image pre-processing procedure impact on feature detection applied to image 6 of *dataset*2

Regarding the detected reflections on the AUV's body, marked in figure 6.2a with red rectangles, since these are stationary throughout both testing datasets, a solution based on background subtraction (Grant, 1984) was implemented. Initially, through the acquired images, a representation of these segments of the image was created and stored. Thus, during the image pre-processing routine, this representation is subtracted to each analysed acoustic image, which allows the removal of the aforementioned segments. The results of such operation are represented in figure 6.2b.

With respect to acoustic echoes portrayed in the acquired images, it is important to take into consideration that, due to the nature of these echoes, lower intensity values are usually associated with these. Therefore, higher intensity values are typically related to reflections on actual landmarks. So, if the maximum intensity value in each intensity beam is found, intensity values found in subsequent bins may be disregarded, since they will most probably be referent to acoustic echoes. A security margin must be taken into consideration, as detected landmarks present thickness and, therefore, will generate reflections of different intensities.

However, for the case presented by the testing datasets, both the first and second maximum intensity values of each retrieved sonar intensity beam are taken into consideration. This mea-

surement is intended to account for the existence of an additional obstacle inside the water tank and avoid the removal of part of the water tank's wall located behind it. More precisely, if the difference between the two maximum values is lower than a given threshold value and the second maximum is detected at a higher detection range than the first one, then, echo removal is performed from this point on, as it can be observed in figure 6.2c.

As it can be observed in figure 6.3, the improvements made to the image pre-processing routine had a significant impact on the number of false-positives detected. Nevertheless, methodology presents its shortcomings, since some acoustic echoes reflections are not removed, due to their high intensity values. Furthermore, it is notorious that a few segments of the water tank have been erased from the final image, due to the presence of prior higher intensity points.

6.3 State Estimator

The implemented state estimator follows the EKF SLAM algorithm presented initially in chapter 2, which makes use of new measurements z_t^j, provided by the acquired matched features, and the previous state vector y_{t-1} to estimate the current state vector y_t. Since this state vector, besides the vehicle's pose, includes each matched feature estimated position, it will also allow the creation and update of an environment map. Note that, to locate the vehicle in the environment, it is only taken into account a two dimensional horizontal plane, assuming constant depth and that environmental features are not affected by depth.

Common EKF SLAM algorithms encompass two major procedures during state estimation, the prediction cycle and the update cycle. This section is focused in presenting further detail regarding the EKF SLAM implementation and state estimation computation.

6.3.1 The Prediction Cycle

The prediction cycle takes place between the acquisition of two consecutive acoustic images. During it, the motion model presented hereafter is used to predict the AUV pose in the next image:

$$\begin{bmatrix} x(t) \\ y(t) \\ \theta(t) \end{bmatrix} = \begin{bmatrix} x_0 \ + \ \int_0^t v(t) \cdot \cos\left(\theta_0 + w(t)\right) \, dt \\ y_0 \ + \ \int_0^t v(t) \cdot \sin\left(\theta_0 + w(t)\right) \, dt \\ \theta_0 \ + \ \int_0^t w(t) \, dt \end{bmatrix}, \tag{6.1}$$

where x_0, y_0 and θ_0 represent the vehicle's initial x and y position and heading, respectively, v its translation velocity, w its rotation velocity and t the time reference.

In order to perform the intended state prediction, the model introduced in equation 6.1 is further discretized as follows:

$$\begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} = \begin{bmatrix} x_{t-1} \\ y_{t-1} \\ \theta_{t-1} \end{bmatrix} + \begin{bmatrix} v_{t-1} \cdot \cos\left(\theta_{t-1} + w_{t-1} \cdot \Delta t\right) \cdot \Delta t \\ v_{t-1} \cdot \sin\left(\theta_{t-1} + w_{t-1} \cdot \Delta t\right) \cdot \Delta t \\ w_{t-1} \cdot \Delta t \end{bmatrix}, \tag{6.2}$$

where x_t, y_t and θ_t represent the vehicle's predicted x and y position and heading at the current iteration, x_{t-1}, y_{t-1} and θ_{t-1} represent the vehicle's x and y position and heading at the previous iteration, v_{t-1} and w_{t-1} the translation and rotation velocities, respectively, at the previous iteration and Δt represents the time interval between each iteration. Note that, between each prediction iteration, the vehicle's motion is considered to be performed at constant velocity. Taking this discretized model into consideration, the state prediction is performed accordingly, following equation 2.7:

$$\overline{\mu}_t = \mu_{t-1} + F_x^T \begin{bmatrix} v_{t-1} \cdot \cos(\mu_{t-1,\theta} + w_{t-1} \cdot \Delta t) \cdot \Delta t \\ v_{t-1} \cdot \sin(\mu_{t-1,\theta} + w_{t-1} \cdot \Delta t) \cdot \Delta t \\ w_{t-1} \cdot \Delta t \end{bmatrix}, \tag{6.3}$$

where $\mu_{t-1,\theta}$ represents the vehicle's heading estimate at $t-1$. Moreover, the Jacobian G_t takes the following form, due to the motion model employed:

$$G_t = I + F_x^T \begin{bmatrix} 1 & 0 & -v_{t-1} \cdot \sin(\mu_{t-1,\theta} + w_{t-1} \cdot \Delta t) \cdot \Delta t \\ 0 & 1 & v_{t-1} \cdot \cos(\mu_{t-1,\theta} + w_{t-1} \cdot \Delta t) \cdot \Delta t \\ 0 & 0 & 1 \end{bmatrix} F_x \,, \tag{6.4}$$

which enables the computation of the covariance estimate $\overline{\Sigma}_t$ through equation 2.8.

6.3.2 The Update Cycle

The purpose of the update cycle is to fuse the measurements $z_t^j = (r_t^j \ \phi_t^j \ s_t^j)$ information, where r_t^j represents the feature's detection range, ϕ_t^j its detection angle and s_t^j its signature, associated to each matched feature j, into the previous state estimation, reaching a final estimation. Note that the feature's signature, s_t^j, is a term used to associate each input measurement to the corresponding feature position estimation.

Therefore, the measurement update is performed based on the following measurement model:

$$z_t^j = \begin{bmatrix} \sqrt{(\overline{\mu}_{j,x} - \overline{\mu}_{t,x})^2 + (\overline{\mu}_{j,y} - \overline{\mu}_{t,y})^2} \\ \text{atan2}(\overline{\mu}_{j,y} - \overline{\mu}_{t,y}, \overline{\mu}_{j,x} - \overline{\mu}_{t,x}) - \overline{\mu}_{t,\theta} \\ \overline{\mu}_{j,s} \end{bmatrix}, \tag{6.5}$$

where $\overline{\mu}_{t,x}$, $\overline{\mu}_{t,y}$ and $\overline{\mu}_{t,\theta}$ represent the vehicle's x and y position and heading estimates, respectively, $\overline{\mu}_{j,x}$ and $\overline{\mu}_{j,y}$ the feature's x and y position estimate and $\overline{\mu}_{j,s}$ the feature's estimate signature.

The integration of the measurement information into the estimation is performed through the computation of the Kalman gain K_t^j, for each matched feature. As presented in equation 2.14, this is performed through the Jacobian of the measurement model H_t^j, which takes the following form, due to the model presented in equation 6.5:

$$H_t^j = \frac{1}{q} \begin{bmatrix} \sqrt{q}\delta_x & -\sqrt{q}\delta_y & 0 & -\sqrt{q}\delta_x & \sqrt{q}\delta_y & 0 \\ \delta_y & \delta_x & -1 & -\delta_y & -\delta_x & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{6.6}$$

Finally, the update to the state vector and state covariance is performed for each acquired measurement, through the expressions presented in equations 2.15 and 2.16.

During the AUV's motion it is possible that some feature points are detected in one image and not on the following one, due to the sonar's limited range, for example. If this feature point is detected again, its match will no longer be possible, which results in a loss of information. Therefore, it matters to emphasize that, for the purposes of position estimation, not only features matched in consecutive acoustic images are taken into account. Instead, a feature register, used to store every detected feature point, corresponding descriptor and signature throughout a mission was created. The matching procedure is then performed taking into account the feature points detected in the most recent acoustic image and the points stored in this register. Once a feature is matched, their corresponding feature descriptor is updated, to account for any geometric deformation it might had suffered and to allow its correct association in future images. If a detected feature point is not matched, it is simply added to the register.

6.3.3 Feature Point Clustering

Throughout the analyses performed during this work, regarding the application of feature extraction algorithms, it was possible to observe that the characteristics of the testing environment had an impact on the distribution of the detected features. As it is represented in figure 6.4, the regions portraying the water tank's corners show a higher concentration of detected features.

Having such concentrations of feature points will result in similar information contributions to the location estimation procedure, which leads to a reduced correction of the belief on this estimation. Furthermore, as it has been related on chapter 5, despite the new matching procedure significantly reducing the number of occurred outliers, these are not completely avoided and typically occur in regions of higher concentration of feature points, where the distinction between these points is more difficult.

Therefore, in order to reduce these concentrations of feature points and so, attenuating the effects of possible outliers, a density-based clustering methodology (Kriegel et al., 2011) has been employed. Clustering algorithms aim at finding groups of similar objects in a given data collection

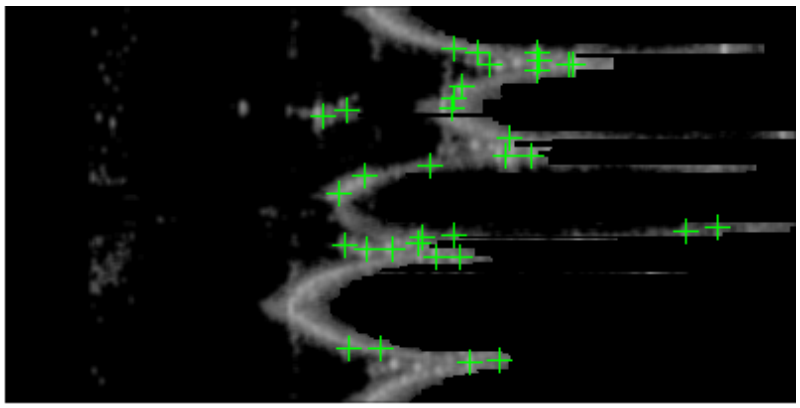
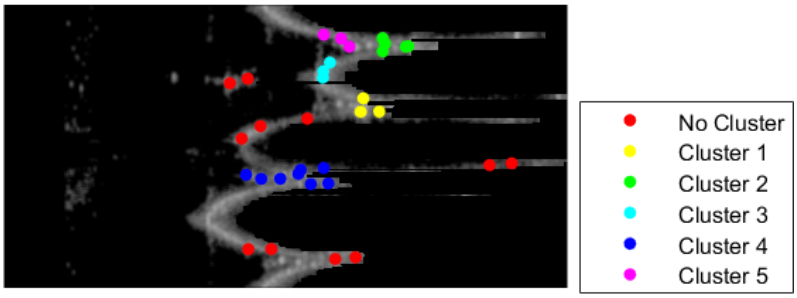
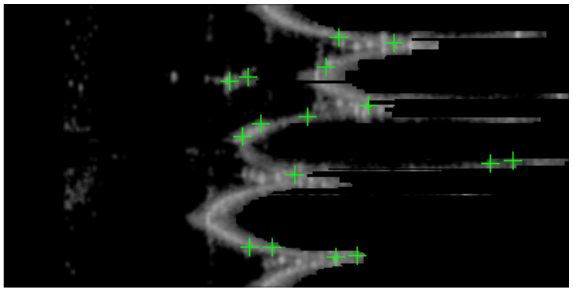


Figure 6.4: Feature points detected in image 1 of *dataset*1



(a) Feature point association during the clustering procedure



(b) Cluster centers and the remaining matched features

Figure 6.5: Results of the clustering procedure employed to image 1 of *dataset*1

(Kriegel et al., 2011). The density-based approached makes use of the data localization information to form this groups. A maximum distance between group points is defined, along with a minimum number of points to assemble a cluster. So, for each given point, the euclidean distance from it to the remaining points of the collection is calculated. The points located at distances lower than the defined maximum are associated to the initial point and the process is performed for each of these, until no new point can be added to the group. A cluster is created if the final number of associated points is higher than the minimum required.

This methodology is directly employed to each set of matched features, making use of each feature localization information, as portrayed in figure 6.5a. The created clusters are represented through their clusters centers, which are computed by an average of the positions of the associated feature points, as presented in image 6.5b. The resulting clusters centers are transformed into a set of measurements $z_t^c = (r_t^c \ \phi_t^c \ S_t^c)$, where c represents the number of created clusters, and fed to the state estimator. Therefore, the state vector has been redesigned to account for this changes, as follows:

$$\mathbf{y}_t = \begin{bmatrix} \mathbf{x}_t \\ \mathbf{m} \\ \mathbf{M} \end{bmatrix} \tag{6.7}$$

$$= \begin{bmatrix} x & y & \theta & m_{1,x} & m_{1,y} & s_1 & \dots & m_{n,x} & m_{n,y} & s_n & M_{1,x} & M_{1,y} & S_1 & \dots & M_{c,x} & M_{c,y} & S_c \end{bmatrix}^T, \tag{6.8}$$

where $M_{c,x}$, $M_{c,y}$ and S_c represent the clusters' x and y positions and signature, respectively, and

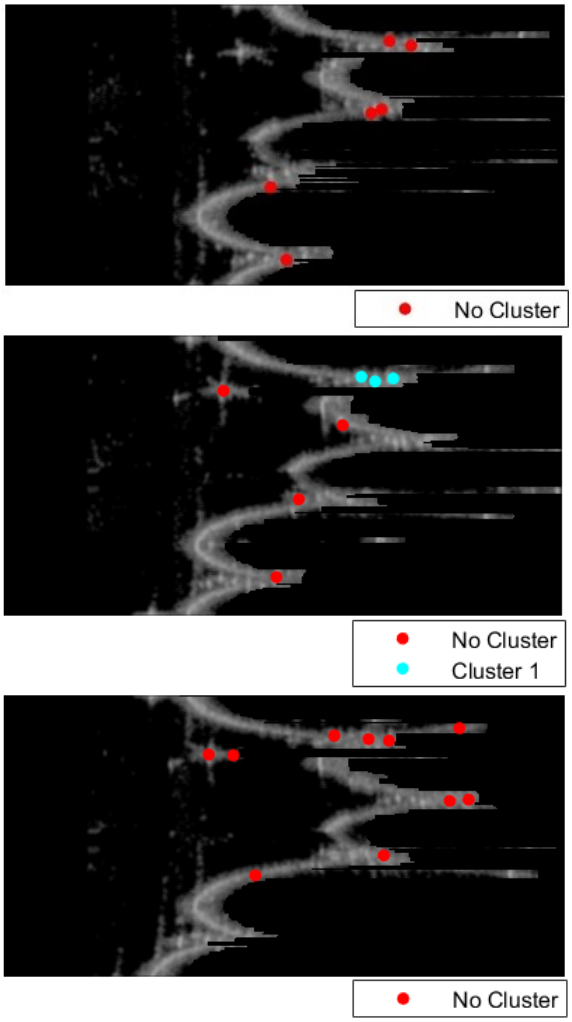


Figure 6.6: Effect of the motion of the AUV on the clustering procedure

$m_{n,x}$, $m_{n,y}$ and s_n the x and y positions and signature of the detected features which have not been associated with a cluster. The incorporation of the cluster measurements is performed as detailed in section 6.3.

It is important to emphasize that the correspondence between detected clusters is performed through the implementation of a cluster register. When a new cluster is created, its localization information is stored in the register, along with the signatures of the feature points that compose that cluster and its own signature. During the analysis of a new acoustic image, after the matching and clustering procedures are performed, the features' signatures associated with each detected cluster are compared to the ones stored in the register, so that the cluster can be identified. This association allows to determine the corresponding cluster signature S_c, key for the update cycle of the state estimator. During this procedure, the feature points associated with each cluster are also updated.

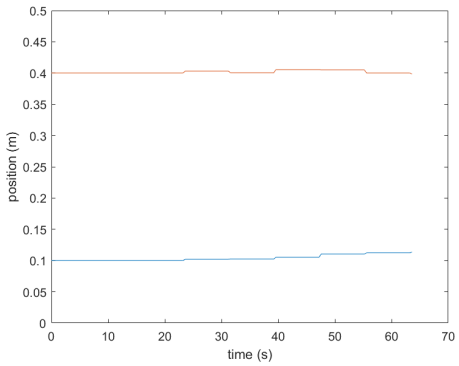
However, when this methodology was employed to the set of images acquired with the AUV in motion it was possible to note the insufficiencies of this procedure. The motion of the AUV has

a significant impact on the matching procedure, leading to a reduction of the matches performed. As it can be observed in figure 6.6, this aspect leads to an impossibility to maintain the structure of the created clusters throughout a series of acoustic images. Therefore, if the cluster information was fed to the state estimator, it would be extremely difficult to update it, adding no significant value to the state estimation. Furthermore, since the feature points that compose a cluster are not fed to the state estimator individually, this would result in a loss of important information. This problem is even more severe taking into consideration the reduction in the number of matches performed with the AUV in motion. Thus, the developed clustering solution was not included in the developed system.

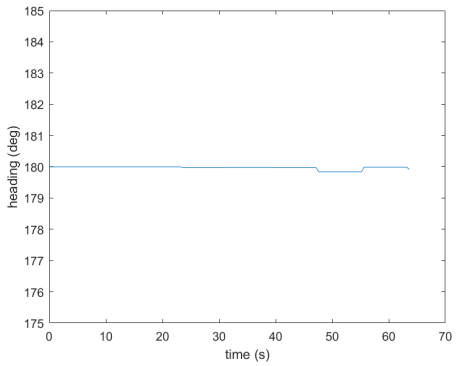
6.4 Results and Discussion

In this section, a comprehensive analysis of the developed localization and mapping module is presented, taking into account the solution structure presented in figure 6.1. Therefore, the feature information extracted from both *dataset*1 and *dataset*2 is fed into the developed state estimator, making use of the image pre-processing procedure introduced in section 6.2. Thus, to assess the performance of this module, the vehicle's pose and landmark's localization estimations throughout time have been collected. The prediction cycle is performed at time intervals of 0.4 seconds while the update cycle at time intervals of 8 seconds. Further information regarding the evolution of the covariance matrices associated to the vehicle's pose estimation and landmark localization estimation has also been extracted. Note that, for the purpose of localization, a cartesian coordinate system was considered and the center of the water tank taken as its origin.

Since it was not possible to collect any information regrading the AUV's velocities throughout both datasets, a simple estimate of these is performed based on the difference between the initial and final vehicle's pose estimations during the update cycle, which are then used during the prediction cycle.

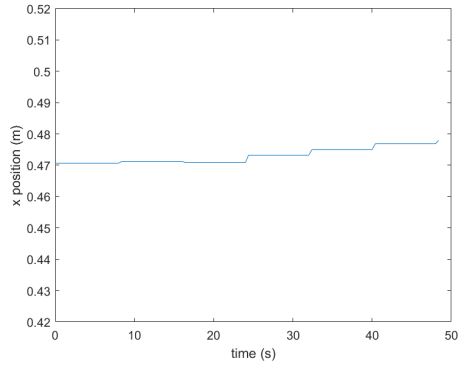


(a) Estimated AUV x (in blue) and y (in red) position over time (in seconds)

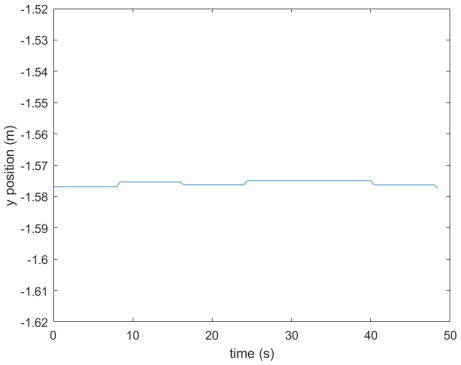


(b) Estimated AUV heading over time (in seconds)

Figure 6.7: AUV pose estimation evolution throughout *dataset*1

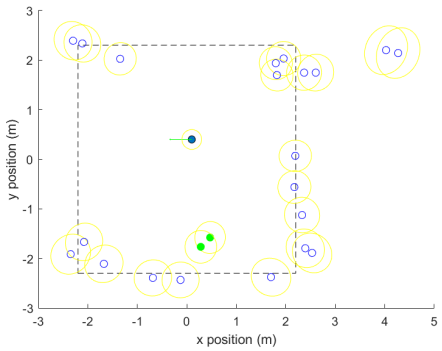


(a) Estimated floater x position over time (in seconds)

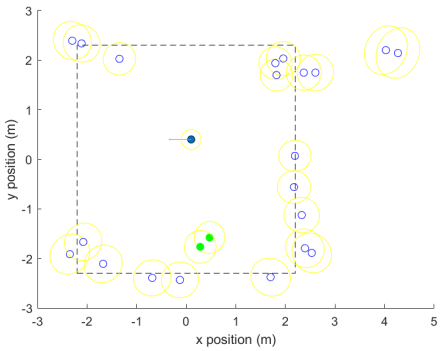


(b) Estimated floater y position over time (in seconds)

Figure 6.8: Floater pose estimation evolution throughout *dataset*1



(a) Results after first update cycle



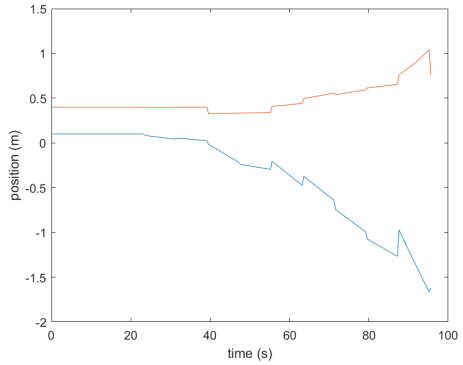
(b) Results after last update cycle

Figure 6.9: Vehicle (filled blue circle) and landmarks estimated positions and associated uncertainty ellipses (*dataset*1)

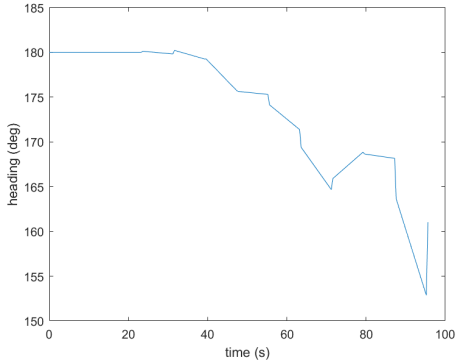
Regarding the application of the localization and mapping module to *dataset*1, it matters to emphasize that this dataset portrays a scenario where the AUV is kept still. Therefore, the vehicle's initial velocities were set to 0 and the initial estimated pose was $(0.1m, 0.4m, 180^{o})$. As it can be observed in figure 6.7, the vehicle's pose estimation does not diverge from the initial state, despite some fluctuation around this value being observable, potentially due to the considered sound speed not being equal to the real one. This effect is more notorious on the x and y positions but still inferior to 2 centimeters. So, it is possible to conclude that the state estimator converges to a final estimation.

This conclusion is furthermore supported by the example presented in figure 6.8. As it is shown, the floater's position estimation suffers slight oscillations, once again, due to variable sound speed, but the corrections performed during the update cycle do not cause this estimation to diverge from its initial value.

As time passes more feature information is incorporated into the state estimation, resulting in changes to the associated covariance matrices, which reflect the uncertainty regarding this estima-

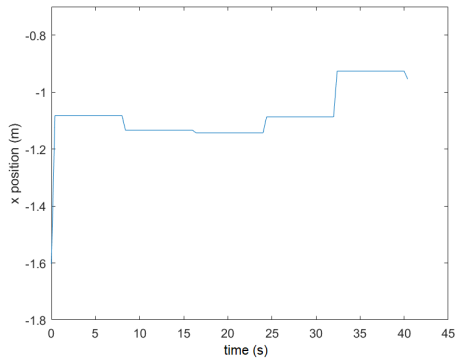


(a) Estimated AUV x (in blue) and y (in red) position over time (in seconds)

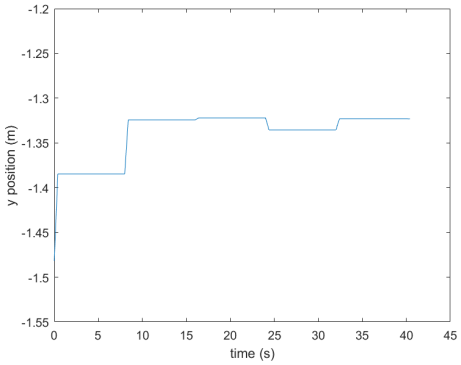


(b) Estimated AUV heading over time (in seconds)

Figure 6.10: AUV pose estimation evolution throughout *dataset*2



(a) Estimated floater x position over time (in seconds)



(b) Estimated floater y position over time (in seconds)

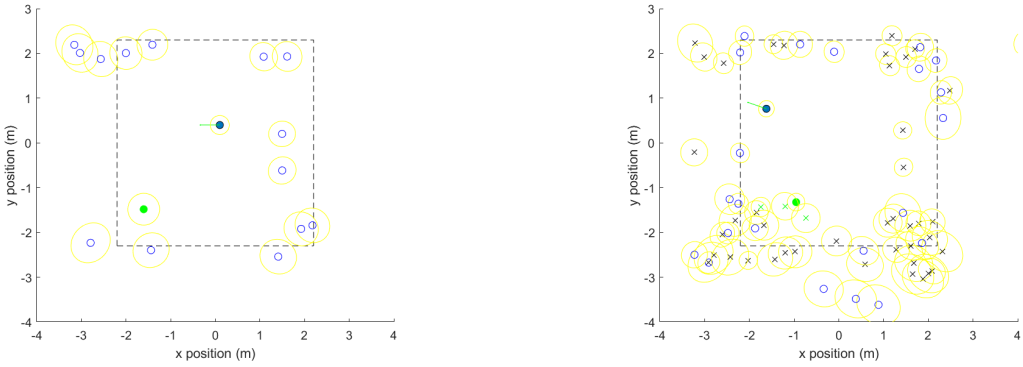
Figure 6.11: Floater pose estimation evolution throughout *dataset*2

tion. These are represented as uncertainty ellipses, centered at the vehicle and landmarks estimated positions, as presented in figure 6.9. At the initial estimation, the ellipses are large, which reflect a higher degree of uncertainty in the estimation, as shown in figure 6.9a. The incorporation of further information results in an increased belief on the state estimation, which is reflected onto the smaller uncertainty ellipses presented in figure 6.9b, asserting the convergence of the filter.

With regard to the application of the localization and mapping module to *dataset*2, which was acquired with the AUV moving on an horizontal trajectory, the initial vehicle velocities where kept null and, again, the initial estimated pose was $(0.1m, 0.4m, 180^{o})$.

Figure 6.10 portrays the vehicle's pose estimation evolution through time. As it would be expected, the vehicle's x position estimate varies, resulting of the vehicle horizontal trajectory. However, a drift in both its y position and heading estimations is notorious. This is a result of the data acquisition process, since it was not always possible to maintain a perfectly horizontal trajectory during the AUV's motion.

However, a careful analysis of figure 6.11 makes it possible to observe that a clear deterioration



(a) Results after first update cycle

(b) Results after last update cycle

Figure 6.12: Vehicle (filled blue circle) and landmarks estimated positions and associated uncertainty ellipses (*dataset*2)

of landmark position estimate occurs. This is further verified by the variation on landmark position estimate that is observable between figures 6.12a and 6.12a. Such problem is most probably the result of incorrectly performed matches and due to the fact that translation and rotation velocities estimation is only performed during the update cycle, and, therefore, during the prediction cycle these are considered constant.

Nevertheless, 6.12 clearly ascertains the convergence of the filter. While for the initial estimation the uncertainty ellipses are larger, throughout time these have come to decrease in size. However, after the last update cycle is performed, some landmarks still present large uncertainty ellipses, which may result from incorrect matches or the occurrence of false detections.

6.5 Summary

In this chapter, the development of the localization and mapping module of the proposed system, based on the implementation of the EKF SLAM algorithm, was presented. Further detail was given regarding the employed motion model and measurement model and its incorporation into the state estimator. Furthermore, the previous performance analysis of feature extraction algorithms have shown that the testing environment, due to its complexity, presents additional challenges to the performance of the AUV. Therefore, the initial image pre-processing procedure has been further developed and a final solution based on background subtraction and the analysis of the first and second higher intensity reflections is proposed to tackle these problems.

The incorporation of a clustering procedure, in order to condense the information portrayed by features closely located and attenuate the effects of possible incorrect matches, was also presented. However, despite interesting results have been obtained, this procedure was not included in the developed system since it has shown several shortcomings when the vehicle is in motion, due to the reduction of matched features.

A brief analysis of the developed state estimator performance was conducted and it was possible to observe that the evolution of the AUV's pose matches the scenarios described for both

$dataset1$ and $dataset2$, while the filter is able to converge to a final estimate. However, for the case of $dataset2$, some deterioration the landmarks' position estimation is notorious, mainly due to effect of possible incorrect matches.

Chapter 7

Results and Performance Analysis

7.1 Introduction

The present chapter is intended to introduce a thorough analysis of the developed system and of the final results achieved. Therefore, further tests have been performed, in two different scenarios. The sonar data relative to the first scenario was collected with the AUV kept steady and immobile while the readings regarding the second scenario were collected with the AUV in motion. These were performed in an square shaped water tank, with a length of $4.4m$ and width of $4.6m$. Ground-truth measurements were performed for comparison performances.

For data acquisition, the sonar was calibrated for a maximum range of 5.0 meters. The bit length was set to $1.25x10^{-2}$ meters and the angle step to 1.8^{o}.

Regarding system parameters, more precisely, the feature extraction and matching procedures, the employed detection threshold was set to 2500 while the deform threshold was set to 0.35. Regarding the matching threshold function, the b value was set to 8 while the k value was set to 500. For the state estimator, the initial covariance estimate, motion noise covariance and measurement noise covariance were initialized as presented hereafter:

$$\Sigma_0 = \begin{bmatrix} 0.2 & 0 & 0 \\ 0 & 0.2 & 0 \\ 0 & 0 & 0.09 \end{bmatrix}, \tag{7.1}$$

$$R = \begin{bmatrix} \frac{(0.2)^2}{35} & 0 & 0 \\ 0 & \frac{(0.2)^2}{35} & 0 \\ 0 & 0 & \frac{(0.09)^2}{35} \end{bmatrix}, \tag{7.2}$$

$$Q = \begin{bmatrix} r_t^j \cdot e^{-2} & 0 & 0 \\ 0 & 0.05 & 0 \\ 0 & 0 & 10x10^6 \end{bmatrix}, \tag{7.3}$$

where the first term of matrix Q is computed according to each measurement range r_t^j. For

both scenarios, since the AUV's initial pose was the same, the initial pose estimation used was $(0.1m,\ 0.4m,\ 180^{o})$. The initial translation and rotation velocities estimations were kept null.

7.2 Static Scenario

For the first test scenario, the AUV was kept immobile, at surface, almost at the center of the water tank. Furthermore, a cross shaped floater was also deployed and moored close to the lower tank wall. Further detail is depicted in figure 7.1, where the AUV's position and the floater's position are depicted. A total of 8 acoustic images were acquired portraying this scenario, over 64 seconds of operation time. Ground-truth measurements of the AUV's position and heading and the floater's position were collected through physical measurement.

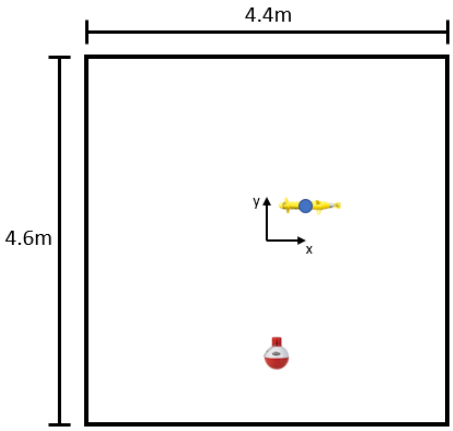


Figure 7.1: Scheme of the static scenario

The data collected was then fed to the developed system. Further information regarding the total number of detected landmarks throughout time and the number of landmarks considered in each iteration of the update cycle is presented in table 7.1. Here it is possible to observe that, for the considered scenario, landmark detection is stable, since the majority of the total detected features are taken into account in each update cycle.

Image Pair	Total Detected Landmarks	Current Detected Landmarks
1-2	23	23
2-3	32	28
3-4	37	34
4-5	39	32
5-6	40	30
6-7	41	33
7-8	42	32

Table 7.1: Number of detected landmarks for the static scenario

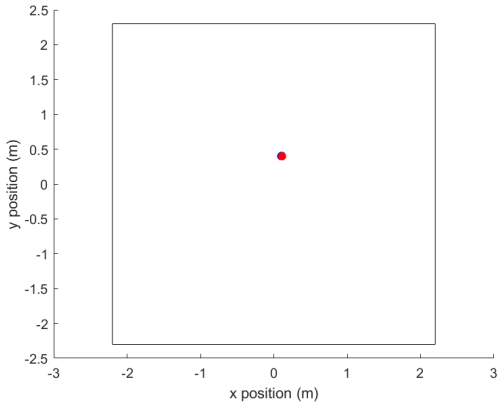


Figure 7.2: AUV estimated positions during the static scenario

	Ground-truth (m, m, deg)	Estimation Results (m, m, deg)	Error (m, m, deg)
AUV	$[0.10, 0.40, 180] \pm [0.10, 0.10, 10]$	$[0.11, 0.39, 180]$	$[0.01, 0.01, 0]$
Floater	$[0.48, -1.57, \sim] \pm [0.10, 0.10, \sim]$	$[0.47, -1.58, \sim]$	$[0.01, 0.01, \sim]$

Table 7.2: Ground-truth measurements and estimated poses for the static scenario

Since no motion of the AUV is portrayed in this scenario, the vehicle's position estimate does not diverge as time passes, as presented in figure 7.2, where the previous position estimates are marked in blue and the final position estimate is marked in red. Some fluctuation of the estimated pose of the vehicle is visible, probably due to the considered sound speed being different from the real one.

The final vehicle pose estimation is presented in table 7.2. By comparing the estimation results for the vehicle's position with the collected ground-truth, it is possible to observe that the associated error is minimum. This is due to initializing the vehicle pose estimation with the actual ground-truth information.

Similarly, the error associated to the floater's final estimated position is also minimum. A

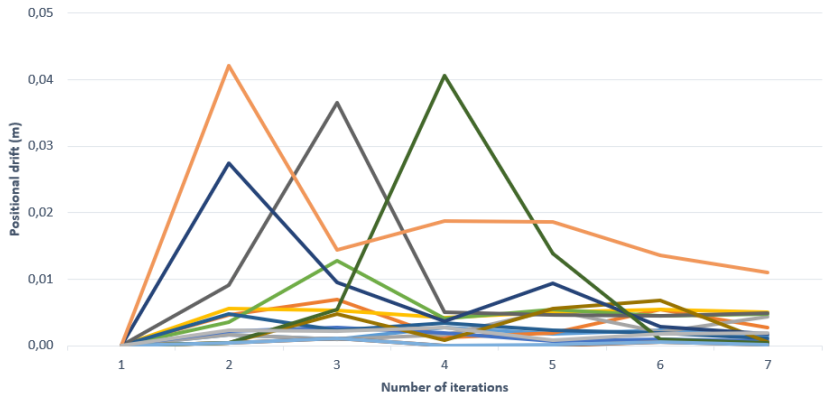


Figure 7.3: Landmark estimated positions drift for the static scenario

small variation of the landmarks' estimated positions throughout time is noticeable in figure 7.3, again, probably due to an error in the considered sound speed, which poses no obstacle to the overall system performance.

7.3 Dynamic Scenario

For the second test scenario, the AUV was initially deployed in the same position as for the previous scenario, again at surface. A cross shaped floater was once more deployed and moored close to the lower tank wall, but in this case, closer to the left tank wall. The data acquisition process was initialized and, after a few seconds, the AUV was dragged as depicted in figure 7.4, where the initial AUV position is represented by a blue circle and the final position by a red circle. A total of 11 acoustic images were acquired throughout 88 seconds of operation time. Ground-truth measurements regarding the AUV's initial and final poses and the floater's position were collected.

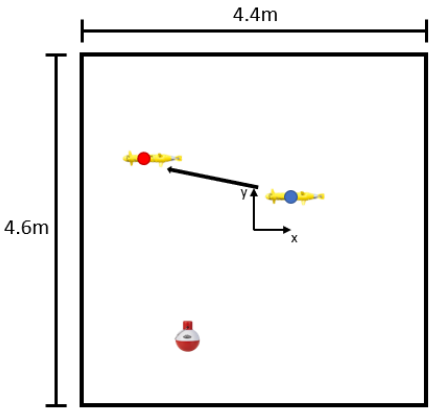


Figure 7.4: Scheme of the dynamic scenario

Again, the collected data is fed to the developed system. The estimated trajectory of the vehicle is represented in figure 7.5, where the past AUV position estimates are represented in

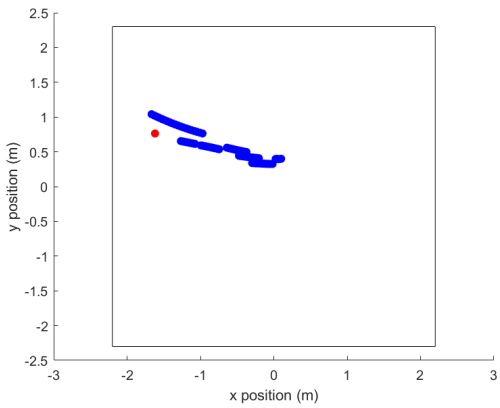


Figure 7.5: AUV estimated positions during the dynamic scenario

Image Pair	Total Detected Landmarks	Current Detected Landmarks
1-2	15	15
2-3	21	18
3-4	25	15
4-5	26	8
5-6	29	5
6-7	41	12
7-8	45	8
8-9	52	12
9-10	60	15
10-11	68	24

Table 7.3: Number of detected landmarks for the dynamic scenario

blue and the final estimate in red. It is possible to observe that the estimated trajectory does not completely match the one that was initially described, because of the occurrence of incorrect matches that affect the state estimator and since the vehicle's velocity is considered constant during the prediction cycle.

The effects of the motion of the AUV are furthermore noticeable in the results presented in table 7.3, where a great variation of the number of detected landmarks considered in each iteration of the update cycle is observable. These results come as a consequence of the decrease in the number of performed matches.

	Ground-truth (m, m, deg)	Estimation Results (m, m, deg)	Error (m, m, deg)
AUV	$[-1.73, 0.69, 170] \pm [0.10, 0.10, 10]$	$[-1.62, 0.76, 161]$	$[0.11, 0.07, 9]$
Floater	$[-1.05, -1.32, \sim] \pm [0.10, 0.10, \sim]$	$[-0.95, -1.33, \sim]$	$[0.10, 0.01, \sim]$

Table 7.4: Ground-truth measurements and estimated poses for the dynamic scenario

It is possible to observe, through a brief analysis of table 7.4, that the resulting AUV and

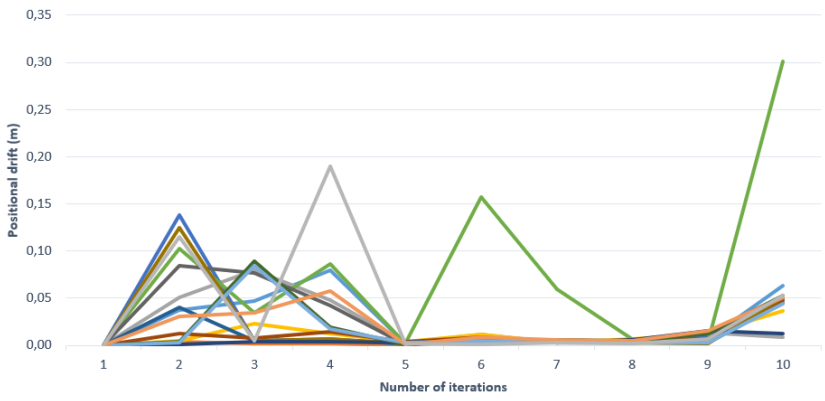


Figure 7.6: Landmark estimated positions drift for the dynamic scenario

floater position error is higher than the one registered for the static scenario. Still, the results are very satisfying, despite the higher landmark estimated position drift that is presented in figure 7.6. This results not only from the difference in sound speed and multiple echoes but, most of all, is due to the occurrence of incorrect matches and to the vehicle's velocity estimation performed.

7.4 Summary

The chapter presented an overview of the system performance under two different scenarios. For the case of the static scenario, the achieved results are extremely satisfying, despite the positional drift that is observable for the detected landmarks. However, some insufficiencies of the proposed solution arise when the AUV is in motion, since the incorrect association of feature points during the matching procedure presents a significant difficulty for state estimation. Furthermore, the vehicle's velocity estimation performed and the assumption of constant velocity during the prediction cycle contribute to the drift in landmark position estimation.

Chapter 8

Conclusions and Future Work

8.1 Conclusions

In retrospective, the main proposed objectives were accomplished as a working solution for underwater vehicles localization and mapping has been developed, based on the operating principles of a MSIS. Its performance was further evaluated with experimental data and analysed in a series of different scenarios.

The analysis and interpretation of sonar data was achieved through the application of image processing techniques, with especial focus on feature extraction algorithms. A comprehensive analysis of some of the most common algorithms presented in literature has been conducted, in order to better understand the most suitable solution for the proposed objectives. Furthermore, this comparison provides valuable information regarding the application of feature extraction algorithms to acoustic images and its challenges.

Due to the insufficiencies of the common feature matching procedure, a solution adapted to the characteristics of acoustic images has been developed, which allowed improving the number of matches performed at lower outlier ratios. Therefore, it was proven that feature extraction algorithms enable a consistent detection and association of environmental features of interest.

Finally, a state estimator based on the common EKF SLAM algorithm has been developed and implemented in order to enable the estimation of the vehicle's pose and mapping of the detected environmental landmarks. Possible associations of close landmarks through clustering techniques has been explored but disregarded due to the effects of the vehicle's motion on the matching procedure.

Despite the fulfilled objectives, the prevailing occurrence of incorrect matches have a negative impact on both vehicle pose and landmark position estimation, which implies that further study and adaptation of feature description methodologies to acoustic imaging is required. Furthermore, it would be important to reduce the number of false detections that still occur and implement the developed system in an actual vehicle, such as the SHAD AUV, to test its performance during online missions.

8.2 Future Work

In the short-term, it would be interesting to carry out further studies regarding the implementation of clustering techniques for feature point analysis. Despite the solution elaborated during this study was not viable, clustering methodologies may allow the extraction of relevant information regarding image regions cluttered with feature points, in order to determine which of the set are more stable and portray more relevant information for position estimation. Furthermore, by reducing the number of feature points fed to the state estimator, it would be possible to attenuate the effects of incorrect matches.

Also, since noise suppression and image smoothing are vital for the performance of feature extraction algorithms, further development of such techniques, focused on acoustic imaging, would bring significant improvements to feature-based localization procedures.

The next step, after the implementation and test of this solutions in an actual vehicle, would be the integration of path planning methodologies that would rely on the built feature map and pose estimation to navigate. The integration of other sensor inputs into localization and mapping procedure would also be of value.

Bibliography

Alcantarilla, P. F., Bergasa, L. M., & Dellaert, F. (2010). Visual odometry priors for robust EKF-SLAM. *Proceedings - IEEE International Conference on Robotics and Automation*, (June), 3501–3506. https://doi.org/10.1109/ROBOT.2010.5509272

Bailey, T., Nieto, J., Guivant, J., Stevens, M., & Nebot, E. (2006). Consistency of the EKF-SLAM algorithm. *IEEE International Conference on Intelligent Robots and Systems*, (October), 3562–3568. https://doi.org/10.1109/IROS.2006.281644

Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, *110*, 346–359. https://doi.org/10.1016/j.cviu.2007.09.014

Calonder, M., Lepetit, V., Strecha, C., & Fua, P. (2010). BRIEF: Binary robust independent elementary features, *6314 LNCS*, 778–792. https://doi.org/10.1007/978-3-642-15561-1_56

Chen, L. W., Sen, M.-M., & Klaus, H. (2013). Towards autonomous localization and mapping of AUVs: a survey. *International Journal of Intelligent Unmanned Systems*, *1*(1), 97–120.

Dengsheng, Zhang. (2019). *Fundamentals Of Image Data Mining - Analysis, Features, Classification And Retrieval* (S. Nature, Ed.; 5th ed.).

Derpanis, K. G. (2004). The harris corner detector. *York University*, (March), 2–3.

Dinuka, H. M., & Bandara, D. (2017). *Sonar Image Based Advanced Feature Extraction and Feature Mapping Algorithm for Under-Ice AUV Navigation* (Master's thesis July).

Elfes, A. (1989). Using occupancy grids for mobile robot perception and navigation. *Computer*, *22*(6), 46–57.

Esrafilian, O., & Taghirad, H. D. (2017). Autonomous flight and obstacle avoidance of a quadrotor by monocular SLAM. *4th RSI International Conference on Robotics and Mechatronics, ICRoM 2016*, 240–245. https://doi.org/10.1109/ICRoM.2016.7886853

Fula, J. P., Ferreira, B. M., & Oliveira, A. J. (2018). AUV Self-localization in Structured Environments Using a Scanning Sonar and an Extended Kalman Filter. *OCEANS 2018 MTS/IEEE Charleston, OCEAN 2018*. https://doi.org/10.1109/OCEANS.2018.8604798

Ganesan, V., Chitre, M., & Brekke, E. (2016). Robust underwater obstacle detection and collision avoidance. *Autonomous Robots*, *40*(7), 1165–1185. https://doi.org/10.1007/s10514-015-9532-2

Gao, L., & Xu, H. (2013). Real-time image processing and mapping algorithm for forward-looking sonar of AUV. *OCEANS 2013 MTS/IEEE - San Diego: An Ocean in Common*, 1–4. https://doi.org/10.23919/OCEANS.2013.6740974

Goncalves, C. S., Ferreira, B. M., & Matos, A. C. (2016). Design and development of SHAD - A Small Hovering AUV with Differential actuation. *OCEANS 2016 MTS/IEEE Monterey, OCE 2016*, 1–4. https://doi.org/10.1109/OCEANS.2016.7761457

Grant, J. T. (1984). Background subtraction techniques in surface analysis. *Journal of Vacuum Science & Technology A: Vacuum, Surfaces, and Films*, *2*(2), 1135–1140. https://doi.org/10.1116/1.572689

International, T. (n.d.). *Micron Sonar Product Hardware Manual* (T. I. Ltd, Ed.; 2nd ed.).

Kriegel, H. P., Kröger, P., Sander, J., & Zimek, A. (2011). Density-based clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *1*(3), 231–240. https://doi.org/10.1002/widm.30

Leutenegger, S., Chli, M., & Siegwart, R. Y. (2011). BRISK: Binary Robust invariant scalable keypoints. *Proceedings of the IEEE International Conference on Computer Vision*, (November), 2548–2555. https://doi.org/10.1109/ICCV.2011.6126542

Li, J. H., Kang, H. J., Park, G. H., Ki, H. S., & Suh, J. H. (2017). Sonar image processing based underwater localization method and its experimental studies. *OCEANS 2017 - Anchorage*, *2017-January*, 1–5.

Li, J. H., Lee, M. J., Lee, W. S., Kim, J. T., Kang, H. J., & Suh, J. H. (2015). Real time obstacle detection in a water tank environment and its experimental study. *2014 IEEE/OES Autonomous Underwater Vehicles, AUV 2014*, 1–5. https://doi.org/10.1109/AUV.2014.7054407

Low, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 91–110. https://www.cs.ubc.ca/%7B~%7Dlowe/papers/ijcv04.pdf

Millar, G. (2015). An obstacle avoidance system for autonomous underwater vehicles: A reflexive vector field approach utilizing obstacle localization. *2014 IEEE/OES Autonomous Underwater Vehicles, AUV 2014*, 1–4. https://doi.org/10.1109/AUV.2014.7054405

Mur-Artal, R., Montiel, J. M. M., & Tardós, J. D. (2015). Orb-slam: A versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, *31*(5), 1147–1163.

Paull, L., Saeedi, S., Seto, M., & Li, H. (2014). AUV navigation and localization: A review. *IEEE Journal of Oceanic Engineering*, *39*(1), 131–149. https://doi.org/10.1109/JOE.2013.2278891

Rosten, E., Porter, R., & Drummond, T. (2010). Faster and better: A machine learning approach to corner detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(1), 105–119. https://doi.org/10.1109/TPAMI.2008.275

Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. *Proceedings of the IEEE International Conference on Computer Vision*, (November 2011), 2564–2571. https://doi.org/10.1109/ICCV.2011.6126544

Sahoo, A., Dwivedy, S. K., & Robi, P. S. (2019). Advancements in the field of autonomous underwater vehicle. *Ocean Engineering*, *Vol. 181*, 145–160.

Tareen, S. A. K., & Saleem, Z. (2018). A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK. *2018 International Conference on Computing, Mathematics and Engineering Technologies: Invent, Innovate and Integrate for Socioeconomic Development, iCoMET 2018 - Proceedings*, 1–10. https://doi.org/10.1109/ICOMET.2018.8346440

Thrun, S. (2002). Probabilistic robotics. *Communications of the ACM*, *45*(3), 52–57. https://doi.org/10.1145/504729.504754

Tueller, P., Kastner, R., & Diamant, R. (2019). A Comparison of Feature Detectors for Underwater Sonar Imagery. *OCEANS 2018 MTS/IEEE Charleston, OCEAN 2018*, 1–6. https://doi.org/10.1109/OCEANS.2018.8604786