Sonar Attentive Underwater Navigation in Structured Environment

Hashim Kemal Abdella

Thesis submitted for the Degree of
Doctor of Philosophy

Heriot-Watt University

School of Engineering and Physical Sciences

Institute of Sensors, Signals and Systems

July 2018

ABSTRACT

One of the fundamental requirements of a persistently Autonomous Underwater Vehicle (AUV) is a robust navigation system. The success of most complex robotic tasks depends on the accuracy of a vehicle's navigation system. In a basic form, an AUV estimates its position using an on-board navigation sensors through Dead-Reckoning (DR). However DR navigation systems tends to drift in the long run due to accumulated measurement errors. One way of mitigating this problem require the use of Simultaneous Localization and Mapping (SLAM) by concurrently mapping external environment features. The performance of a SLAM navigation system depends on the availability of enough good features in the environment. On the contrary, a typical underwater structured environment (harbour, pier or oil field) has a limited amount of sonar features in a limited locations, hence exploitation of good features is a key for effective underwater SLAM.

This thesis develops a novel *attentive* sonar line feature based SLAM framework that improves the performance of a SLAM navigation by steering a multibeam sonar sensor, which is mounted on a pan and tilt unit, towards feature-rich regions of the environment. A *sonar salience map* is generated at each vehicle pose to identify highly informative and stable regions of the environment. Results from a simulated test and real AUV experiment show an attentive SLAM performs better than a passive counterpart by repeatedly visiting good sonar landmarks.

To my parents

Acknowledgements

First of all I would like to say ‘*Alhamdulillah*’. Everything happens for a reason and a lot of things have happened to get me here.

There are many people who have helped with my research over the last few years, yet it all started when I joined the Ocean Systems Laboratory. Hence, I would like to express my sincere appreciation to my supervisor, prof. David Lane, for giving me the opportunity. A special thank you to Dr. Keith Brown for tirelessly going through my drafts and providing valuable feedbacks. A big thank you to colleagues in the Ocean Systems Laboratory for all the encouragement and moral support.

Endless thanks to my family who have always been a constant source of love, support, encouragement and inspiration. Finally, I am indebted to my wife, Zuleyha, thank you for all your contributions to our family.

Funding

This work has been supported by the FP7-ICT-2011-7 European project PANDORA-Persistent Autonomy through Learning, Adaptation, Observation and Re-planning (Ref. 288273).

ACADEMIC REGISTRY

Research Thesis Submission

Name*:*			
School:			
Version: *(i.e. First, Resubmission, Final)*		Degree Sought:	

Declaration

In accordance with the appropriate regulations I hereby submit my thesis and I declare that:

1) the thesis embodies the results of my own work and has been composed by myself
2) where appropriate, I have made acknowledgement of the work of others and have made reference to work carried out in collaboration with other persons
3) the thesis is the correct version of the thesis for submission and is the same version as any electronic versions submitted***.**
4) my thesis for the award referred to, deposited in the Heriot-Watt University Library, should be made available for loan or photocopying and be available via the Institutional Repository, subject to such conditions as the Librarian may require
5) I understand that as a student of the University I am required to abide by the Regulations of the University and to conform to its discipline.
6) I confirm that the thesis has been verified against plagiarism via an approved plagiarism detection application e.g. Turnitin.

* *Please note that it is the responsibility of the candidate to ensure that the correct version of the thesis is submitted.*

Signature of Candidate*:*		Date:	

Submission

Submitted By *(name in capitals):*	
Signature of Individual Submitting:	
Date Submitted:	

For Completion in the Student Service Centre (SSC)

Received in the SSC by *(name in capitals):*			
Method of Submission *(Handed in to SSC; posted through internal/external mail):*			
*E-thesis Submitted (**mandatory for final theses)***			
Signature:		Date:	

Please note this form should be bound into the submitted thesis.

Academic Registry/Version (1) August 2016

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

AUV	Autonomous Underwater Vehicle
DoF	Degrees-of-Freedom
DR	dead reckoning
EKF	Extended Kalman Filter
FISST	Finite Set Statistics
FoA	Focus of Attention
FOG	Fiber-optic gyroscope
FoV	Field of View
GM	Gaussian Mixture
IMU	Inertial Measurement Unit
IoLR	Inhibition of Large Rotation
IoR	Inhibition of Return
KF	Kalman Filter
LBL	Long Baseline
MSIS	Mechanically Scanned Imaging Sonar
PHD	Probability Hypothesis Density
RFS	Random Finite Set
ROV	Remotely Operated Vehicle
SBL	Short Baseline
SDK	Software Development Kit
SFoA	Sonar Focus of Attention
SLAM	Simultaneous Localization and Mapping

STD	Standard Divatation
UKF	Unscented Kalman Filter
USBL	Ultra-Short Baseline
UWSim	UnderWater Simulator
WTA	winner-take-all

Symbols

G_d	Distance weight
β	Sonar beam angle of incidence
$\mathbf{A}^{-1}$	the inverse of matrix $\mathbf{A}$
$\{A_k\}$	AUV local coordinate frame at time step k
$\{W\}$	map coordinate frame
$\mathbf{A}^\intercal$	the transpose of matrix $\mathbf{A}$
$\mathcal{N}\left(;\hat{X},P\right)$	normal distribution with mean $\hat{X}$ and covariance P
ω	rate of rotation
$f(\cdot)$	state transition model
F_w	the lineraization of $f\left(\cdot\right)$ with respect to the velocity noise
F_x	the lineraization of $f\left(\cdot\right)$ with respect to the state
G_a	Angle of incidence weight
G_o	Overlap weight
$h(\cdot)$	observation model
H_k	the lineraization of $h\left(\cdot\right)$
k	time step
P_k	the state covariance matrix
Q_k	velocity noise covariance matrix
R_k	measurement noise covariance matrix
T	sampling period
v_x	forward velocity
v_y	lateral velocity
$[\cdot]^{\{W\}}$	sonar measurement with respect to $\{W\}$ coordinate frame
X_k	AUV state at time k

Chapter 1

Introduction

Today the demand for sea exploration is increasing. Commercial companies are making substantial investments for surveying, running and maintain offshore structures. Governments spend a significant portion of their budget to safeguard their sea territory or protect harbours. Scientists are putting a huge effort in researching the open water which covers three quarter of the earth's surface, shelters enormous biodiversity, and affects the whole life on earth directly or indirectly. Some of these tasks require reaching inaccessible locations or dangerous areas for human divers. Additionally, some tasks require frequent attendance which make them cumbersome. Remotely Operated Vehicles (ROVs) can be used to alleviate this problem and limit human involvement in order to avoid substantial risk. However, ROVs lack autonomy, requiring human pilots at all times and they are physically constrained within the proximity of a support vessel which provides power and control through a cable. Typically the length of the linking cable dictates the operation area within a close range of the connected vessel. This lack of flexibility and autonomy in ROVs together with the recent progress in the technology is making Autonomous Underwater Vehicles (AUVs) a more favourable alternative.

Lately, AUVs are being used for various underwater applications at different scales. There have been reports of AUVs being used for seabed mapping, environmental monitoring and assessment, mine counter measure, underwater search and rescue, and many other applications. AUVs have taken charge of most tasks that used to be executed by human operators and ROVs. In addition to reducing human involvement, AUVs are more efficient when surveying or periodically inspecting underwater structures like harbours, dams and offshore structures. However achieving a true persistent autonomy is still a challenge that requires further research, hence it became the main agenda of an EU FP7 PANDORA project [1]. The project aimed to achieve persistent autonomy of AUVs for tasks such as: inspection, cleaning and intervention. The success of these tasks is highly dependent on the accuracy and robustness of the core vehicle navigation system.

In comparison to land robot navigation, underwater navigation is far more challenging due to the limitation that arises on certain types of sensors due to the environment. For instance, the visibility for optical cameras is very poor in underwater as a result they can be used only for close range applications. Similarly receiving a GPS signal is only possible when a vehicle is close to the surface which, for most applications, is out of the vehicle's workspace. As a result most traditional AUVs get their position estimation from an on-board sensors. Although such systems are capable of executing short missions their long term autonomy is significantly affected by a measurement drift arising from associated sensors. This led to the growing interest in underwater Simultaneous Localization and Mapping (SLAM) using acoustic sensors. There has been various implementations of underwater SLAM algorithms to assist underwater navigation (more discussion on this will be presented in Chapter 2). A SLAM based navigation system has to address three main challenges for underwater applications: *sensor restriction*, *scarcity of landmark* and *limited resource*.

The limitation on the choice of sensor makes it impossible to use those sensors that can provide high quality measurement or good amount of features. As a result an underwater SLAM system has to relay on relatively noisy measurements coming out of acoustic sensors. To make things worse, in a typical underwater environment the number of acoustic landmarks available is limited which in turn restricts the amount of SLAM feature that can be gathered. Hence a good underwater navigation systems need to exploit these limited landmarks. Lastly, for a typical AUV, there is a hard-line restriction on the amount of available resource. Everything including batteries need fit into a small watertight capsule which typically lack a dedicated cooling mechanism or a powerful computer. Therefore, in order to achieve persistent autonomy navigation systems need to use light weighted algorithms and make the best out of the limited landmark features available in underwater environment using noisy measurement coming from acoustic sensors.

This thesis is intended to explore the possibility to turn available light weighed SLAM algorithms into a robust active navigation systems. To achieve this goal we will introduce a focus of attention mechanism to the underwater domain by devising a sonar based salience map that allows for efficient utilization of limited landmark features.

1.1 Problem Statements and Thesis Aim

1.1.1 Problem Statements

The work in this thesis explores methodologies that address the problem in improving robustness of underwater SLAM navigation systems in the absence of abundance sonar landmarks. We aim to address the most sensitive parts of a feature based underwater SLAM algorithm that need to be handled carefully in order to improve the localization performance. In this work three stages of SLAM implementations are identified as the main source of failure in underwater SLAM implementations: *poor sensor set-up*, *inaccurate feature extraction* and *unreliable measurement model*.

A typical feature acquisition require the use of mechanically fixed sonar sensor to observe underwater landmarks that fall into its field of view. However landmarks do not always lie in front of such sensors hence an effective utilization of available environmental features requires an active sensor set-up. In addition to proposing a focus of attention based active sensor set-up this work presents an efficient feature extraction technique that uses the full sensor information for accurate identification of landmarks. The third stage of failure is the measurement model used for a line feature based SLAM. Currently, there is a discrepancy among the literature in how the measurement model for 2D line segment should be represented. For instance, Ribas *et al.* [2] propose a generic single model while in case of Garulli *et al.* [3] two distinct motion models are suggested where the choice is made based on the line segment position relative to the current vehicle pose and the map origin. This discrepancy is addressed aiming to avoid the resulting localization error from a wrong measurement model.

1.1.2 Thesis Aims and Objectives

As it is going to be discussed in the next chapter, most attentive systems or active vision schemes are developed for non acoustic applications; depriving underwater researchers the possible benefit of such systems. The aim of this thesis is to adapt the principle of active vision and localization for underwater environment by developing a persistent active localization and mapping technique for structured underwater environment. The proposed system can extract useful landmark information from the surroundings efficiently using actively steerable sonar sensor which facilitates the convergence of sonar based SLAM navigation system. It also aims to explore the benefits of the Probability Hypothesis Density (PHD) filter for underwater sonar based SLAM. In this work we intend to achieve the following objectives:

- asses techniques for representing sonar salient features and review systems for underwater navigation.
- devise a mechanism for generating sonar based salience map to dictate the collection of sonar data from areas rich with informative features.
- propose an active Simultaneous localization and mapping (SLAM) technique that could re-detect available landmarks and track them for an improved navigation.

1.1.3 Evaluation Metrics

The overall performance of the proposed system is measured in terms of the reduction on the localization error relative to conventional SLAM algorithms. These errors are computed as X-Y position RMSE (Root-Mean-Square Error) values between estimated robot poses and ground truth. Initially, simulated tests are used to asses the performance boost that each individual changes can introduce to a feature based underwater SLAM algorithm. Later tests in an indoor tank are executed to confirm these results.

Euclidean distance measure is adopted to estimate the distance between two robot positions. However, other metrics are used to compare distance between line segments and uncertainty in Chapter 3.

1.2 Platform and Test Environments

Before going any further, lets introduce some of the test facilities that are used to test and validate the proposed techniques.

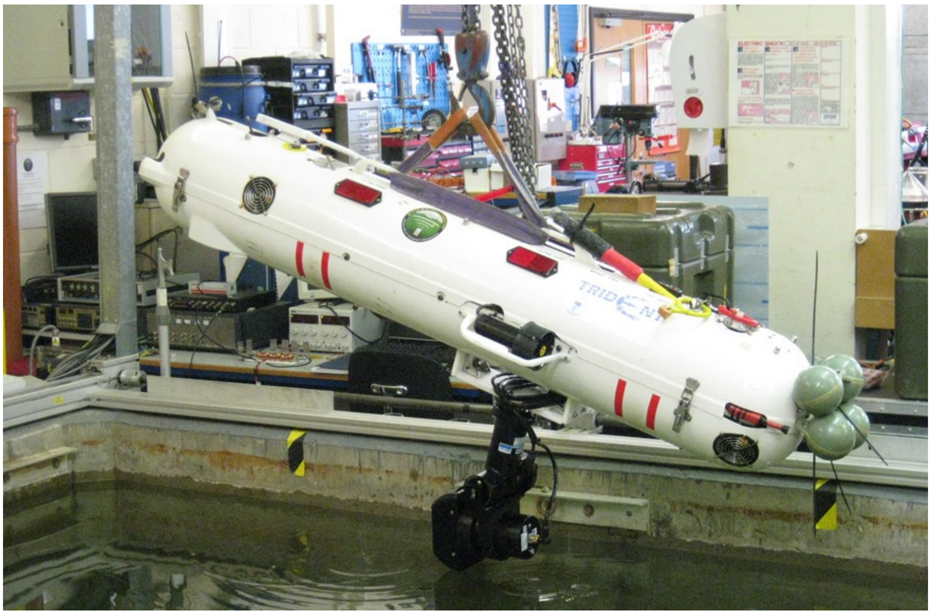


Figure 1.1: *Nessie VII* Ocean Systems Laboratory (OSL) main research AUV with pan and tilt sonar underneath for steering multibeam sonar.

1.2.1 UWSim Simulator

UWSim is an underwater simulator developed for assisting marine robotics research and development [4]. It allows creation of virtual 3D underwater scenarios using standard modelling software to test underwater vehicles. Vehicles can be equipped with simulated sensors including a simulated multi beam sonar profiler. This will allow multiple controlled tests to be done before actual runs and it will also avoid unnecessary costs arising from expensive sea trails.

1.2.2 *Nessie VII* AUV

Simulated tests as well as tank experiments are devised based one of Ocean System Lab's (OSL) AUV called *Nessie VII*. *Nessie VII* is equipped with on board navigation sensors including: Teledyne Explorer PA DVL and A TCM 6 compass. Additionally, two type of sonars can be attached to a pan and tilt unit to provide sonar observations as shown in Figure 1.1. Further details on general navigation sensors and those on *Nessie VII* is presented in Chapter 2.

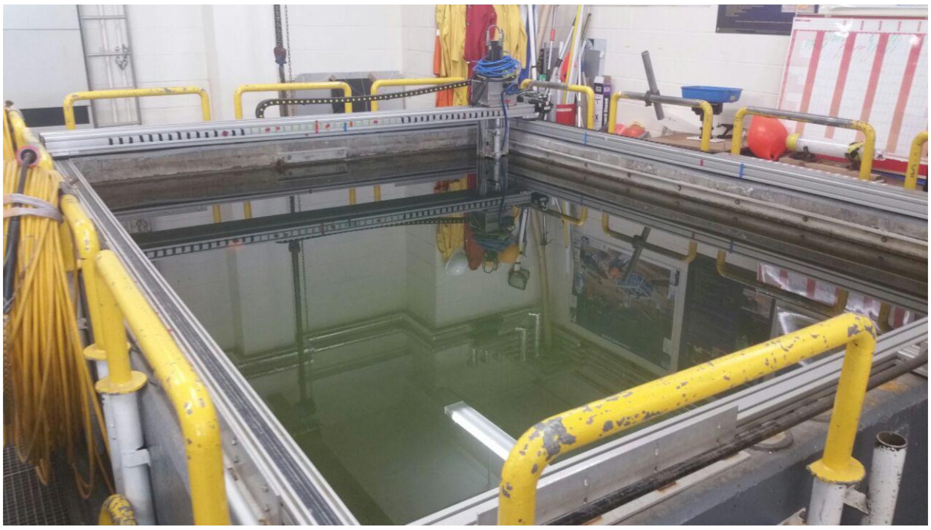


Figure 1.2: Ocean system lab indoor test tank and Cartesian robot.

1.2.3 OSL Cartesian Robot and Indoor Tank

The ocean Systems Laboratory (OSL) has an indoor tank facility which has a 3-axis (X-Y-Z) Cartesian robot built to it, see Figure 1.2. The Cartesian robot provides an alternative test platform with three degrees of freedom. Test can be run by attaching a pan and tilt unit to the base of the unit. The ROS (Robot Operating System) driver for the Cartesian robot can return the status of the robot in the three major axes. This can be used as a ground truth while generating noisy trajectories out of it. This tank is 4 m long, 3 m wide and 2 m deep. The test tank is used for validating feature extraction techniques as well as the proposed navigation system while the robot emulates the characteristics of *Nessie VII*.

1.2.4 HWU Wave-tank

Some of the tests require a bigger test arena and for that the HWU wave-tank provides a good option. It is a testing facility with in the School of Built Environment, Heriot-Watt university. It covers an area of 12 m by 10 m while the depth varies from 2 m around the edges to 4 m in the centre of the pool. In addition to having artificial beach in one side, this facility provides a means to simulate a rough sea condition by generating various types of waves as show in Figure 1.3.

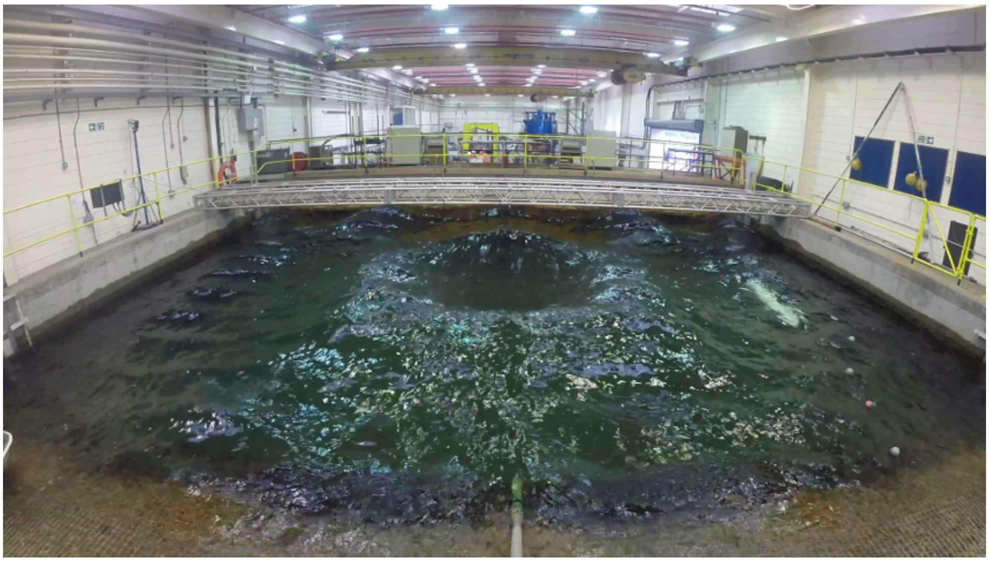


Figure 1.3: HWU wave-tank used for testing attentive SLAM on *Nessie VII*.

1.3 Thesis Layout

The thesis structure is shown in Figure 1.4 over the proposed system architecture. Hardware requirement of the proposed system include a vehicle with basic navigation sensors (*i.e.DVL*, *Depth sensor...* more on this in Chapter 2), a multibeam sonar sensor and a pan & tilt unit to steer the sonar around. On top of this lie a feature extraction module that feeds a feature based underwater SLAM algorithm with landmark features which are picked based on the decision made by a focus of attention mechanism. The remainder of this thesis is organised as follows: In Chapter 2 we provide a brief background on various components of a feature based underwater navigation system in relation to the three problem mentioned in previous section. The chapter begins by reviewing state-of-the-art in underwater sensing and navigation then we discuses the advantages of using a focus of attention based active vision for mobile robotics. The review in this chapter provides the foundation for the subsequent three chapter.

In Chapter 3 the characteristics of multibeam sonar images are studied. The outcome of this study is used to build a realistic multibeam sonar simulator as well as assisting the understanding of a feature extraction process. The sonar simulator becomes an important tool in the evaluation of both the feature extraction and localization process.

Chapter 4 discuses two of the most widely used light weighted SLAM filters: EKF (Extended Kalman Filter) and UKF (Unscented Kalman Filter). This filters are compared using sonar line features extracted using technique proposed in Chapter 3. The main focus

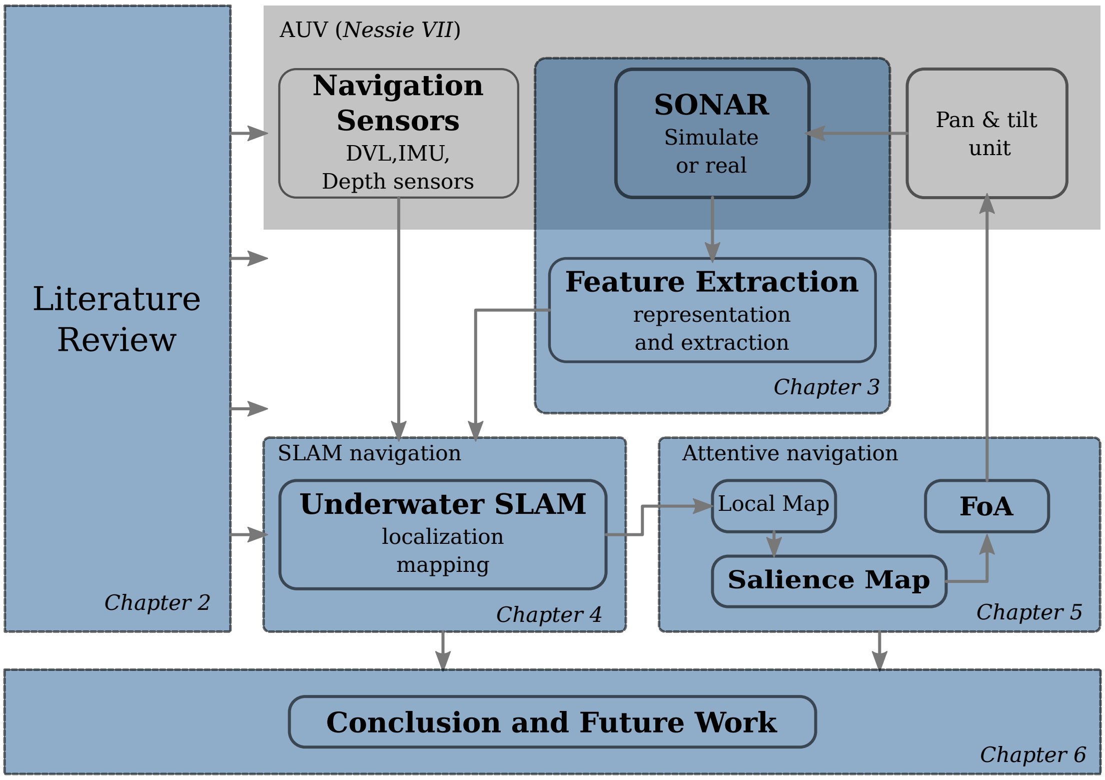


Figure 1.4: Thesis layout and proposed system architecture.

of this chapter is solving the problem associated with peaking the right measurement model for line feature based SLAM. Then Chapter 5, being the main contribution of this thesis, present a novel technique for computing sonar salience and improving a passive SLAM navigation by actively steering a sonar head mounted on a pan and tilt unit. We demonstrate how the use of focus of attention improves the frequency of landmark sighting which directly affect the performance of a feature based SLAM algorithm. Finally, Chapter 6 concludes the thesis by providing a summary of accomplished tasks and giving a direction for future improvements.

Chapter 2

Literature Review

In this chapter, we first provide a short background on currently available underwater navigation systems and sensors that they use. Then we discuses how these systems benefited from an underwater Simultaneous Localization and Mapping (SLAM) followed by a review on sonar feature extraction techniques in an underwater structured environment. Later in the chapter, a brief overview on the principle and applications of active vision in the context of robotics is presented. Then a background on focus of attention, as a type of active vision technique, is presented through a discussion on its importance for achieving primate like active perception. Finally, milestone works in attentive vision for robotic application are briefly presented.

2.1 Underwater Navigation

More than seventy percent of the earth's surface is covered by water, hence a lot has to be done to facilitate the exploitation and study of the deep ocean. Advances in technology have brought vehicles which can reduce the cost of surveying seabed or reduce the complexity of underwater structure inspection [5, 6]. Yet, Autonomous Underwater Vehicles (AUVs) require human intervention in order to accomplish persistent autonomy. One of the core tasks in underwater vehicles that requires such intervention is navigation. Unlike human counterpart, autonomous robots are not accurate enough in navigating through a new environment, mainly due to inevitable sensor limitations and measurement drifts. In a basic form, AUVs use on-board sensors to estimate the position of a vehicle. However, a more accurate navigation is achieved by incorporating external clues through other scientific sensors and informations from dynamic and kinematic models [5]. According to Kinesey *et al.* , current AUVs are able to use closed-loop feedback control system and provide a more detailed survey information as a result of improved precision and high update rate of navigation [5].

2.1.1 Common Underwater Navigation Sensors

AUVs use various sensors for providing navigation data. There is a variation in terms of the type and number of sensors that researchers mount on their vehicle. However, there are certain basic sensors which are common to most AUVs including *Nessie VII* (see Section 1.2). In this section we will discuss some of these sensors and their performance. Ultimately, this clarifies the review on navigation systems and the demand for further research on improving underwater navigation precision.

A. Depth Sensor

Depth sensors, also known as pressure sensors, compute the distance of a vehicle from the water surface using the ambient water pressure. There are two main categories of depth sensors: strain gauges and quartz crystals. Strain gauges based sensors use metallic alloys on an elastic pressure diaphragm in a Wheatstone Bridge while quartz crystal pressure sensors, as the name implies, utilize quartz crystals. The presence of ambient ocean pressure changes the resistance of the metallic alloy or causes a stress on quartz crystal which is translated into the strength of the water pressure. For both types of sensors the sensor accuracy can be improved by proper calibration and compensations for thermal variations and offset. In general quartz crystal based sensors are 10 times more accurate than that of strain gauge based sensors for full-scale [5]. A Keller Series 33X depth sensor on *Nessie VII*, which uses a piezoresistive transducer, has an absolute accuracy of 0.005 bar at fullscale, equivalent to 0.05 m, over a temperature range $+10$ to $+40$ °C and 0.01 bar for a temperature between -10 and $+80$ °C.

B. Magnetic Sensor - Compass

Two-axis and three-axis magnetic sensors are widely employed for measuring vehicle orientation. The technique employed in these sensors include flux-gate, magneto-inductive and magneto-resistive magnetic sensing methods. These methods use the presence of magnetic anomaly in order to determine the local north. Properly calibrated magnetic compasses can provide an accuracy on the order of $1°$. According to the datasheet a TCM 3 compass on our vehicle has $0.5°$ heading accuracy for a tilt angle below $70°$. Yet Kinesey *et al.* claim magnetic heading sensors are the main error sources in an overall navigation solutions [5]. This is mainly due to

- the magnetic disturbance from the vehicle,
- effect of vehicle acceleration on gravity based roll-pitch compensation methods,
- local magnetic and geographic anomalies, and
- poor alignment or mounting of the sensor on the vehicle.

C. Doppler Velocity Log (DVL) sensor

Doppler Velocity Log (DVL) sensors use beams of sonar pings to estimate 3D vehicle velocities based on frequency or Doppler shift. These velocities are normally taken relative to the reflection point on the bottom surface; however, most new generation sensors employ water-lock Doppler tracking when the vehicle altitude is out of the bottom-lock range. Teledyne Explorer DVL on our vehicle has a run time accuracy of 0.2 cm/sec or a relative accuracy of 0.4%. Like the angular rate sensor, DVL sensor accumulates velocities to estimate position and it require other sensors for estimating initial vehicle condition. Any error in these values or a wrong alignment between the DVL and the attitude sensor will result in poor absolute position estimation. Additionally, the position uncertainty of DVL dead reckoning navigation grows without bound in a long duration missions.

D. Inertial Measurement Unit (IMU)

Inertial Measurement Units (IMUs) are self-contained navigation systems that measure linear and angular vehicle motion with a triad of gyroscopes and triad of accelerometers (discussed below). The resulting velocities and accelerations are integrated to yield a dead reckoning position. Since these sensors use inertia of an inertial reference mass they are also known as inertial sensors. There are two types of IMUs: *inertial-frame* and *strap-down*. In inertial-frame IMUs both the gyroscope and the accelerometers are mechanically integrated. However, in case of strap-down IMUs the two sensors are directly attached to the vehicle and the information is integrated using microprocessors. The main drawback of IMUs is their power consumption. Their market price is also another challenge that limits their widespread use. Additionally, IMUs suffer from a growing and unbounded position error, hence in most applications these sensors are complemented using DVL or acoustic navigation sensors.

E. Gyroscope

Gyroscope is another important navigation sensor used for estimating orientation rate and this can be integrated to measure the robot orientation indirectly. First-generation gyroscopes were too expensive to be deployed on standard AUVs. Later vibrating mechanical gyroscopes become cheaply available though these sensors are less accurate (of the order of $1°$-$5°$ per second) to be used for orientation angle estimation. On the other hand, optical gyroscopes, though they are expensive, can provide a much better precision for orientation estimation. Fiber-optic gyroscope (FOG) and ring-laser gyroscope can detect an orientation rate in the order of $0.1°$-$10°$ per hour. In an optimal condition the uncertainty of KVH DSP-300 FOG on *Nessie VII* can be under $5°$ per hour. However this accuracy can be degraded by improper mounting on the vehicle like most other orientation sensors. Additionally, since gyroscopes are an incremental orientation sensors we need to provide initial estimation for an absolute angular measurement. This is achieved using a compass based initialization on *Nessie VII*.

F. Accelerometer

An accelerometer is used to measure the translational acceleration in the inertial frame of the sensor. It uses a mass attached to a spring to sense any acceleration which causes a displacement of the mass. Typical high end standalone underwater accelerometer can detect an acceleration up to ± 500 g where 1 g = 9.81 m/s² while low cost devices can measure acceleration with in the range of ± 50 g with a sensitivity range 10 to 100 mV/g.

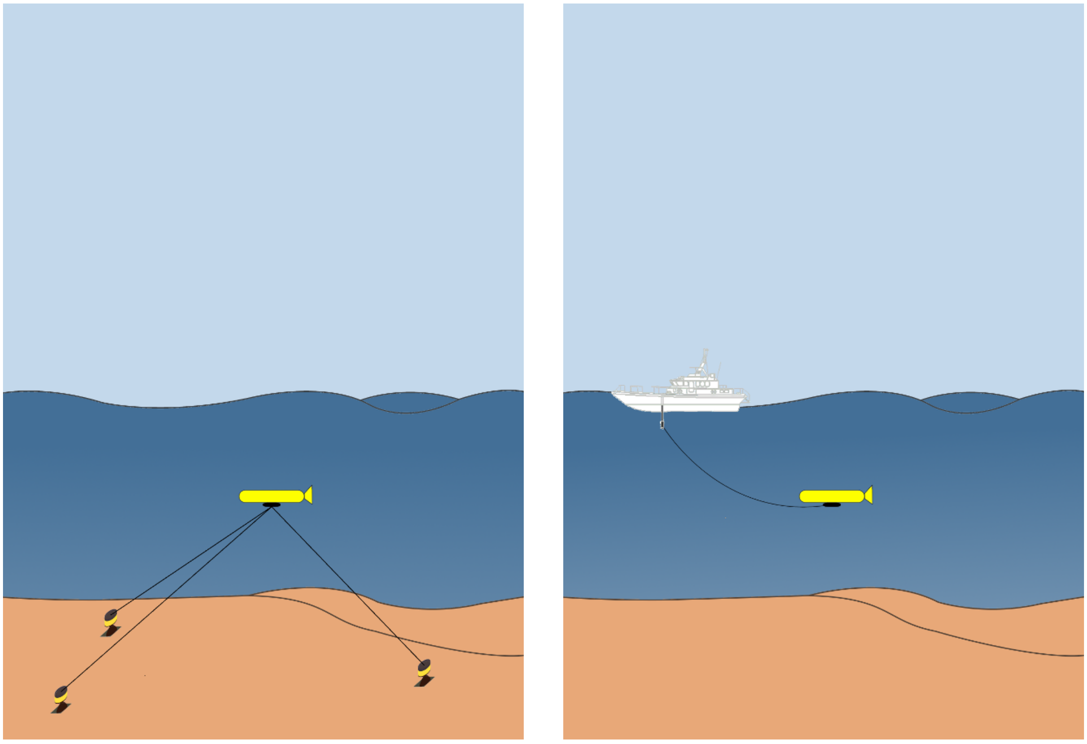
G. Time of Flight Acoustic Navigation Sensors

Sensors in this category use acoustic communication between the vehicle and sets of transponders in order to estimate the vehicle position. Early time of flight acoustic based navigation like Long Baseline (LBL) positioning systems employ acoustic transducer/rangers anchored to the seafloor (see Figure 2.1 on page 14) to compute the relative position of the AUV by triangulation. High frequency (around 300 KHz) LBL systems can achieve subcentimetre XY positioning; however, these sensors have a very limited maximum network range. Standard LBL systems at 12 KHz are reported to work in 10 Km range with a precision in the order of 1-10 m. Most LBL system use transducers moored on the seafloor, yet there are reports of transducer on the hull of a surface vehicle, sea ice or global positioning system (GPS) equipped surface buoys [5]. An alternative set-up that uses two or more closely positioned transponders is called Short Baseline(SBL) positioning system where the baseline distance ranges between 20 and 50 m. In case of SBL system the receiving hydrophones are usually mounted under a vessel floating on the surface hence the measurement accuracy is affected by the stability of the vessel. Further down when the system uses a precisely localised single hydrophones separated by an order less than 10 cm such system is called Ultra-Short Baseline (USBL) positioning system. Unlike LBL and SBL, USBL system uses signal run time and the phase shift of the reply signal to determine the distance and direction of the vehicle. USBL system can be deployed with a single transducer array on a surface vessel; however, its accuracy is an order of magnitude smaller than LBL system ([7] report 1 sigma circular positioning accuracy of 2.7-5.3 m). The main source of error in acoustic time of flight navigation sensors is imprecise transducer placement. Accurate knowledge of the sound velocity at different levels, ambient water temperature and density is also another source of error [5]. Another significant drawback of acoustic positioning system is their restriction of autonomy since the vehicle has to remain in an area covered by transponder network. Additionally, such systems are affected by multipath reflections, occlusion, interference from other beacons in the area, and physical phenomenon such as air bubbles due to USBL mounting ship's movement.

H. Scientific Sensors

Sensors in this category, also known as geophysical sensors, include visual cameras, sonar sensors (Section 2.3 discuss sonar sensors in detail) and magnetometers. These sensors are used to make an absolute observation of the external underwater environment. Geophysical sensors are capable of detecting, identifying and classifying environment features. Typically, the information from these sensors is used to correct navigation drift that arise through dead reckoning. These sensors can be integrated with other sensors in robot state estimators (more on this in Section 2.2).

A form of navigation where a vehicle uses only on-board sensors is called *dead reckoning* (DR). DR navigation systems integrate acceleration and velocity measurements to advance the vehicle position in a given orientation. DR navigations systems mostly use



(a) LBL acoustic navigation system.

(b) USBL navigation system.

Figure 2.1: Time of flight acoustic navigation sensors. Dark solid lines show communication between the transponders and the AUV. In LBL system (a) full deployment of transponder on the sea floor is required, while on (b) a USBL navigation system uses transponder attached to a ship.

sensors which complement one another in order to compute the full or partial vehicle pose [8]. Though this is the most widely used navigation system in AUVs, it suffers from a cumulative navigation drift through time or distance travelled [6]. Integration of expensive IMUs into DR navigation systems can mitigate this drift for a short or medium range expenditure; however, on the long run even these systems start drifting. An alternative approach for a more accurate navigation requires the use of vehicle kinematic or dynamic models in addition to information from some of the sensors mentioned above through analytical state estimation techniques. Some of these techniques will be discussed next.

2.1.2 Navigation and Localization Through State Estimation

Before we go any further, we would like to clarify the distinction between navigation and localization. When we talk about navigation it is all about how a vehicle can get from point A to point B. While localization involve localizing an AUV in a map. When the map is not available the problem turns into a SLAM, where the system construct a map while localizing itself in this map simultaneously. In this work we use the term navigation and localization interchangeably and our intention is to determine the robot position with respect to the starting point.

Most of the navigation sensors mentioned in previous section have limitations. The acceptability of their measurement depends on the application and duration of given AUV mission. A better navigation performance is achieved through the use of deterministic or stochastic state estimators which integrate these measurements with other knowledges about the robot or the surrounding environment.

Deterministic State Estimators

Deterministic state estimators use an exact model of the AUV and measurement models for computing the robot pose [5]. In [9], Jouffroy used contraction mapping for vehicle velocity tracking using an exact non linear vehicle dynamic model. Contraction theory is a non-linear analysis for studying the stability of non-linear systems which will result in a simpler observer design or tracking system [9]. Kinesey *et al.* also report a deterministic single degree-of-freedom (DOF) Remotely Operated Vehicle (ROV), that integrates high precision LBL position measurement with exact knowledge of the vehicle's dynamics, the force and moment acting on the vehicle [5]. Accordingly, the observer can provide a position estimation that can only be achieved using a high-end low frequency LBL system. Note that the requirement of an accurate observer is not limited to AUVs, it is also required for executing high precision tasks in ROV that require station keeping. In [7], a deterministic USBL based observer is used to provide a good enough accuracy for closed-loop control. This kinematic estimator has two stages of operation. First the DVL velocity measurement is integrated to yield a high frequency position estimation. Once in awhile this estimation is corrected using USBL position measurement. This estimator uses an Euclidean distance based outlier rejection technique to avoid erroneous USBL updates.

A principal drawback of deterministic state estimators is the lack of probabilistic concept when integrating noisy measurement. These techniques do not provide analytical methods for selecting the optimal gain, they rather use heuristic and simulation based techniques [7].

Stochastic State Estimators

By far the most common state estimation techniques in use are stochastic in nature which integrate noisy sensor measurements with dynamics and kinematics of the robot. Most stochastic state estimators do also integrate measurements from scientific sensors in order to provide a robust and a more reliable navigation information based on external features. Though the use of such sensors is optional in majority of such systems, Stutters *et al.* claim an accurate long mission navigation can only be achieved through the use of external references for resetting possible sensor drift [10]. The main advantage of stochastic techniques over deterministic state estimations is the possibility of quantifying the level of trust in each measurement or process model in order to provide an optimal gain.

The simplest form of stochastic estimator to-date is Kalman Filter (KF), which is one solution of a Bayesian filter (to be discussed in Chapter 4) [10, 11, 12]. The KF is a recursive two stage filter based on kinematic and measurement models. The kinematic model describe the vehicle state transition and in KF it is assumed to be linear while the observation model defines the relationship between the robot state and the measurement. In a KF both the state transition and the measurement models are supposed to be linear with an additive Gaussian noise for optimal performance. In [13], an aided Inertial Navigation System (INS) is develop using IMU and DVL water-track and DVL bottom-track velocity measurements. In an initial set-up they have only used DVL watar-track aided INS, later a complementary bottom-track DVL measurement is added. Having water-track mode in addition to the bottom-track DVL measurement makes their navigation more stable in a mid-water mission. Their Navigation system uses KF to fuse measurements from depth sensor, IMU, and DVL. Jalving *et al.* developed DVL aided INS that can integrate position estimations from various measurements [12]. Their core DVL aided INS consists DVL, IMU and kinematic model of the vehicle. Additional position estimations are obtained from the pressure sensor, optional compass sensor, GPS surface fix, acoustic navigation system and bathymetric terrain navigation system. Their bathymetric terrain navigation system uses a multibeam echo sounder to determine vehicle position over a bathymetric map. Some of these measurements are redundant which makes the KF estimation more robust.

Since almost all kinematic or observation models of AUVs are highly non-linear, most underwater navigation systems employ a non-linear variant of KF. An Extended Kalman

Filter (EKF) is the most successful variant of KF where non-linear process and observation models are linearized using first-order Taylor series approximation [6, 10, 14]. This approximation is handled using Jacobian matrices whose derivation is non trivial that makes the implementation complex compared to KF. In most EKF based navigations systems, the filter is used to fuse information coming from INS with DVL and/or GPS fixes. For instance, in [15] an EKF filter is used for GPS/INS integration, where a cheap IMU and GPS sensors are used to provide a reasonable position estimation. Their filter state representation is designed so that they can apply correction to the vehicle's full state and the IMU errors. Hence accelerometer bias, gyro drifts, GPS clock time and frequency bias are also compensated on the fly. Additionally, they have also done a comprehensive analysis on the noise representation and they recommend the use of small initial noise covariance for a slow but less noisy convergence. In [16] an EKF technique integrates a GPS receiver, strap down INS and DVL to achieve a high position accuracy of the order of 1 m, which increases to 3 m in the absence of the GPS. In another aided-INS work, Mongado *et al.* simulated an integration of USBL into a strap down INS system using EKF which allows a stochastic characterization of a round trip sound travel time [17]. They have compared their result with a typical loosely-coupled set-up, where triangulation and sensor fusion are handled separately. Accordingly, their system can provide a better estimation of vehicle position and orientation as well as sensor biases estimations.

Not all navigation problems can be solved using EKF. Some of the main shortcomings of EKF include its large estimation error when applied to strongly non linear systems which is expected as EKF is a first order approximation of Taylor series. An EKF filter also requires a very good initial position estimation for optimal performance and a poor dynamic and observation models might cause the filter to diverge quickly [18].

For applications which involve higher order kinematic models an Unscented Kalman Filter (UKF) is found to be much more robust than EKF for a slight degradation on computational performance. UKF uses unscented transform which approximates a distribution using few statistical sigma points [18]. These sigma points, which are typically distributed around the mean along the major and minor axis, can capture second and third order terms of Taylor expansions for non-linear models accurately [18]. The absence of Jacobians in UKF means it can be used for systems whose Jacobian is difficult to compute [18, 19]. Zhang *et al.* use an UKF to integrate strap-down IMU, inexpensive single GPS, and digital compass for autonomous ground vehicle navigation [20]. Their two stage implementation consists of establishing the complex vehicle dynamics while considering the earth's self orientation, sensor bias and system noise. In the second stage they use UKF for system state estimation. In [21], a comparison between EKF and UKF on a data collected using Typhoon-class AUVs shows that UKF filter performs better on compensating sensor drift. In their first experiment their AUV was equipped with IMU, pressure sensor, DVL and a GPS which was available at the starting and ending point of a mission. In their second experiment, they compared their result with a USBL navigation data which again shows the superiority UKF by providing a better mission trajectory.

In their basic form, these variants of KF use a prior knowledge of the system to model the system and measurement noise. Hence any error in this initial model or any subsequent change in this error noise estimations can highly degrade the performance of the filter. According to Chen *et al.* , it is common to experience a change in a system during a mission which could be caused by external force or sensor failures; hence, basic filters tend to diverge when the actual covariance is quite far from the initial estimate [18, 22]. An alternative solution requires the use of an adaptive Kalman filter where the process and measurement noise covariances are adaptively adjusted over the time span of a mission. Chen *et al.* classify adaptive Kalman filters in to two, multiple model adaptive estimation (MMAE) and innovation adaptive estimation (IAE) [18]. Techniques in both categories are based on an innovation sequence, which is the difference between the actual measurement and the filter prediction. MMAE based systems use a bank of linear Kalman filters that run in parallel with different models [18]. The final estimation is computed through a weighted sum based on the innovation in each filter. MMAE based terrain reference navigation has been successfully tested in a simulated underwater environment in [23]. The IAE approach uses a single filter where the noise covariance matrices are continuously adjusted based on the innovation sequence [18]. Tao *et al.* uses a time-varying noise estimation recursively to estimate the measurement noise based on the innovation [24]. Their simulated implementation of UD factorization-based adaptive Kalman filter integrates measurement from strapdown IMU, magnetic compass, DVL, depth sensor and underwater terrain map. However their result being a comparison of conventional Kalman filter and an adaptive UD factorized Kalman filter, it does not clearly show the contribution of the adaptive noise computation since the factorization has a part to play. Similarly Sun *et al.* used an adaptive Kalman filter for improving an integrated dead reckoning (DR) navigation system [25]. They integrated DVL, FOG, depth sensor, USBL, and GPS sensors into an adaptive Kalman filter based navigation system. Their limited knowledge of the system makes it difficult to model the noise a priori; hence, they used an adaptive Kalman filter that reduces the model error and subsequently suppresses filter divergence.

The recent advances in scan matching techniques for point cloud based ground vehicle localization has attracted some underwater researchers for a sonar aided localization [26]. Scan matching techniques estimate the relative displacement of a vehicle by maximizing the overlap between multiple range scans gathered either by a laser or a sonar sensor [26]. The most successful scan matching technique to-date is Iterative Closet Point (ICP) technique which minimizes the sum of the square error between corresponding points. Hernandez *et al.* used a probabilistic Iterative Correspondence (pIC) which takes in to account the sensor and displacement uncertainties [26]. In their implementation, an EKF filter is employed to provide an initial estimate of DVL-based navigation which is refined using pIC. This work has been extended to solve a full SLAM problem by Mallios *et al.* in [27] where locally an EKF filter is used to deal with motion-distortion that arise while completing scan formation from individual beams of a Mechanically Scanned Imaging Sonar (MSIS). This filter uses constant velocity model based on velocities coming from a DVL and heading from an attitude and heading reference system to provide an initial estimate for the state change which is used to undistorted the scan. Later a global trajectory of the vehicle is estimated using an augmented state EKF SLAM where a modified pIC algorithm is used to register each new scan with respect to all previous scans that are in the range. A similar technique is used by Burguera *et al.* for AUV self-localization through a probabilistic data assassination using an MSIS [28]. Their comparison with a non-probabilistic ICP shows the importance of stochastic characterization for reducing the average drift.

Most of the techniques discussed so far under a stochastic state estimation are limited to solving a localization problem without taking the demand to mapping into account. However, a proper autonomous vehicles need to understand its surroundings for effective navigation or any subsequent path planning requirement. Therefore, it is a common practice to handle the mapping as well as the localization problem at once in a technique called Simultaneous Localization and Mapping (SLAM). The SLAM algorithms allow an AUV to navigate through an unknown environment while incrementally building a map using scientific sensors and localizing itself within this map. Next, we discuses more about SLAM techniques for underwater applications which will be followed by a detailed discussion on sonar sensors in Section 2.3 as sonar sensors are the main data acquisition sensors for underwater SLAM applications.

2.2 Underwater SLAM

Ever since the introduction of SLAM technique by Hugh Durrant-Whyte and John J. Leonard [29] based on earlier work by Smith, Self and Cheeseman [30], several techniques has been proposed as a robust alternative for autonomous navigation. These techniques mainly vary based on the approach taken in modelling the motion as well as observation of features. Additionally, SLAM techniques also vary on how the system perceives environmental features or simply the type of the environment. Based on the navigation environment, underwater SLAM is a category of SLAM technique used by AUVs to concurrently map underwater feature while localizing the robot within this map.

Underwater SLAM is a challenging research topic for two main reasons. First, there is limitation on the type of localization sensors that can be used in underwater. As a result most of the high precision sensors, such as GPS, laser, do not function well in underwater environment, as discussed in Section 2.1.1. Second, commercially available AUVs use computationally limited processors inside a sealed compartment which needs to satisfy minimum heat exhaust requirement. Therefore a good underwater SLAM algorithm has to employ an acoustic sensor and should not consume much of the limited processing capability of the robot. In this section we present a review of the most influential underwater SLAM works.

2.2.1 State of the Art of Underwater SLAM

One of the earliest probabilistic SLAM algorithm proposed by Leonard *et al.* uses forward-looking sonar to complement navigation information from an INS and a DVL [31]. In this work they devised an off-line stochastic SLAM algorithm for post processing a forward-looking sonar dataset collected by Naval Undersea Warfare Center (NUWC) and Groupe d'Etudes Sous-Marines de l'Atlantique in Narragansett Bay area around NUWC. A single state vector is used to carry the both the vehicle state and the map features jointly. Con-

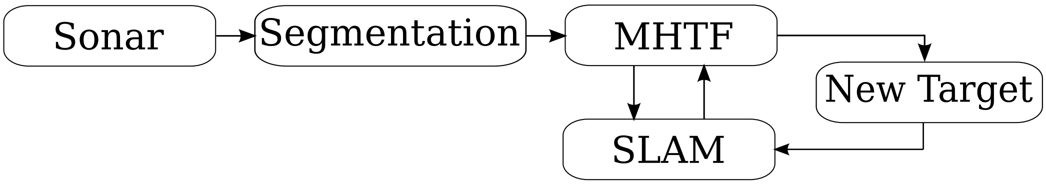


Figure 2.2: Underwater SLAM with MHTF data association. (Reproduced from [32])

trol commands are obtained out of INS readings by computing the velocity and orientation changes required to move the vehicle from a position in previous time to a new pose. Additionally, they used a logic-based track initiation and deletion strategy to address new feature appearances or clutter drop-outs. Assuming sonar returns originate from a single feature in a given area, before applying this logical decision any new measurements are compared to map features using nearest neighbour data association.

Underwater SLAM researchers have introduced improvements in various aspects of probabilistic SLAM for the past fifteen years. One of these improvements involve the use of a more robust data association as an alternative to an association merely based on feature location proximity. Rueiz *et al.* suggested the use of use of a point feature descriptors (see Section 2.3) through Multiple Hypothesis Tracking Filter (MHTF) to facilitate data association [32]. MHTF is one of the earliest visual tracking algorithm where a tree of potential track hypothesis is built for each candidate target. The likelihood of each track is used to determine the most likely combination of tracks, thereby solving the data association problem. As shown in Figure 2.2, an initial sonar data from SeaBat 6012 echo sounder is pre-processed and targets are extracted using a segmentation process. This targets are represented with geometric descriptors that are fed to the MHTF data association algorithm for computing track likelihoods. The Ocean System Lab (OSL) indoor tank and field experiments demonstrate the filter consistency and boundedness though there was no ground truth data available for the field experiment. Even though no computational report has been provided, having MHTF data association on top of an EKF-SLAM algorithm does not make it any simpler.

In another work Ruiz *et al.* proposed the use of SLAM for offline mosaicing sidescan sonar images [33]. In an initial attempt a forward-looking sonar is replaced by side scan sonar in EKF-SLAM navigation system. However, such a direct swap neglects the physical difference between forward-looking and sidescan sonars. The main reason behind this difference is lack of continuous target re-observation in sidescan sonar, unlike forward-looking sonar. Targets are observed again only when the AUV returns on an adjacent parallel track which results jerks in the trajectory due to delayed correction. In order to address this problem, they proposed the use of a smoothing post-process. Smoothing is a post-process operation where previous and future measurements are used to estimate the state of a system, while estimation is an on-line operation where a filter uses all previous information for determining the system state. Hence the addition of a smoothing backward filter will avoid jerks in trajectory through gradual and smooth application of measurement corrections earlier than an actual re-observation. However, unlike their previous work in [32], this technique relies on operators for selecting recognizable landmarks as well as matching landmarks for data-association.

One of the challenge in underwater SLAM is the transition from indoor tank based experiments to a real world application. Large scale SLAM researches are intended to fill this gap and any associated additional complexity arising from this transition as will be discussed next.

2.2.2 Large Scale Underwater SLAM

One way to handle the computational cost of large scale SLAM is through dividing the SLAM problem into clusters based on location. Aulinas *et al.* propose the use of independent local maps to bring down the complexity of EKF SLAM from $\mathcal{O}(n^2)$, where n is the number of landmarks [34]. However, such a decision comes at the expense of losing map consistency through landmark correlation. In order to avoid this, they proposed updates on old local maps whenever there is a possible loop closure. An EKF-SLAM algorithm is used to fuse measurements from DVL, IMU and salient features extracted from side-scan sonar. They used landmarks including objects, rocks and other detectable environmental features. The reference frame for a new local map is considered as the final robot pose in the previous map. Then, a global topological map is generated using this relation where edges are labelled with the relative transformation between maps. Information from this global map is used for verifying loop closures. Additionally, maps that share enough features are fused together to provide a better update. The size of the sub map is fixed based on maximum number of features and decided based on the computational requirement. Experiment on a REMUS-100 AUV demonstrates that sub-maps of size between ten and fifteen number of features provide a good compromise between computational cost and overall

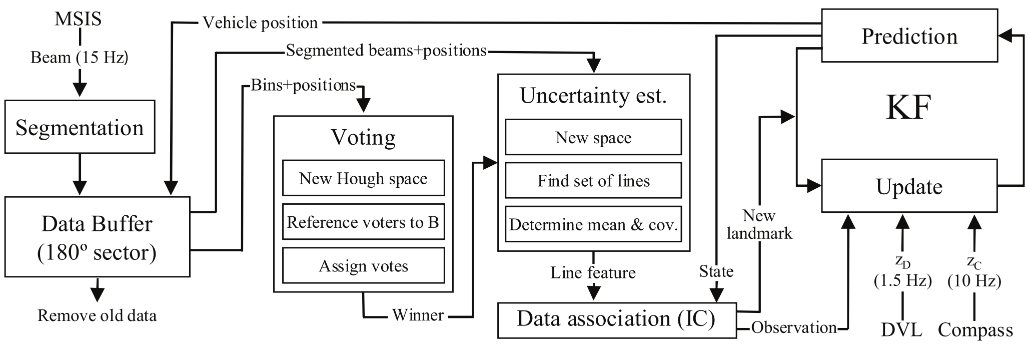


Figure 2.3: Underwater SLAM algorithm Ribas *et al.* [2].

map consistency.

In a work close to ours, Ribas *et al.* propose a SLAM solution for solving the navigation problem around structured environments [2]. Accordingly, landmarks around structures such as dams, harbours, marinas, and marine platforms are better represented using line features. A sonar based 4D EKF-SLAM algorithm is used to improve a DR navigation system composed of compass and DVL (see Figure 2.3). They use a novel sonar feature extraction technique (see Section 2.3.1) to detect walls using MSIS sonar. The robot state is composed of 3D position, horizontal heading angle and the rate of change in all the four dimensions. Every time a new line is detected it is augmented into the state to form a feature map. Separate updates are used each time a new measurement comes out of either DVL, compass or line extractor. The use of line based feature representations are attractive for reasons discussed Section 2.3.1. However, the downfall in using line features is a strong non-linearity of the observation model. In order to tackle this problem, they proposed the use of sub-map which, in the mean time, can enhance the capability of the system to handle long run missions. Unfortunately, there is a limited discussion on the choice of sub-map size which is critical in sub-map based SLAM. If a local map size is large, then the very purpose of using a sub-map is undermined. On the other hand, when a map is too small it will lead to loss of correlation among map features and lack of global landmark integrity. The later is caused mainly due to lack of enough transparency among sub-maps. From the result and discussion new sub-maps are initialized once the robot gets out of a circular region defined by maximum sub-map range, this robot pose is used as an initial position for the new map through which the two adjacent sub-maps are connected. However, after leaving the territory if the root comes back there is no means to alter the old map which contradicts with the concept of loop closure.

Underwater SLAM can also be used to assist mapping sea beds. In [35], Roman and Singh proposed SLAM methodology to improve sub-sea acoustic multibeam bottom mapping. They use a SLAM framework to associate sub-maps of point clouds generated from downward-looking multibeam sonar. The robot uses DR navigation based on FOG, depth sensor and DVL to generate sub-maps in a short time period. Each map has its own coordinate system at its centroid through which the sub-map is represented in the overall state vector. The size of these sub-maps, which directly affect the computational efficiency of the system, is determined based on two conditions. First, they have to be small enough to limit the over all DR drift. This is determined by comparing the growing uncertainty of the vehicle DR pose with an empirically determined threshold. Second, sub-maps need to be large enough to contain enough 3D information so that it can be registered with another sub-map. The information content of sub-map is measured using the Eigen structure of the principal component matrix. Hence, as a minimum sub-map size criteria a threshold is compared to the condition number of this matrix. Closure of sub-map is followed by a two step registration process so that the link between the current sub-map and other sub-map neighbours is generated and registered. The first step require a coarse alignment where overlapping maps are transformed into a common coordinate frames and then point clouds are matched using a 2D correlation algorithm. After applying this initial correction a fine tuning is achieved through *Iterative Closest Point* (ICP). To evaluate the method, an LBL ground truth navigation data is compared with the proposed technique which shows a better alignment in the later. The main challenge with such a proposal remain to be the computational cost, yet it is a suitable solution for offline data processing.

2.2.3 Robust SLAM and Outlier Rejection

Our main interest is to introduce focus of attention to the underwater domain; hence, the choice of SLAM framework is soley based on simplicity for implementation. However, we also want to try out a methodology which is becoming the dominant mathematical framework within the sensor fusion community for developing multi-target tracking algorithm: FInite Set STatistics (FISST) based multi-sensor multi-target filter [36]. These techniques provide a unified probabilistic framework to handle multiple moving targets in a highly cluttered environment. We will have more about these techniques especially on Probability Hypothesis Density (PHD) filter in Section 4.5. However, PHD-SLAM or FISST is not the only framework which can handle outliers in SLAM data association, in the remainder of this section we will briefly discuses some of the most widely used robust SLAM algorithms.

Currently graph-based SLAM is accepted as one of the state-of-the-art techniques and there is a growing trend to robustify its backend optimization [37, 38, 39, 40, 41, 42, 43]. Robust graph optimization or inference methods can be categorized in to two [38]. Researches in the first category are aimed to discount inconsistent graph constraints while those in the second allow multiple components per constraints [38]. Pfingthorn and Birk also propose a generalized graph SLAM framework that can handle arbitrary forms of outliers both locally (those that arise through sensor registration or odometry error) or globally (error due to wrong loop closures) [38]. They modelled uncertain loop closures using hyper edges while as in their previous work [37] they used a mixture of Gaussian to model constraints. This generalized graph SLAM introduces an intermediate stage between classical fronted and backend which makes the implementation efficient and robust against various noise in underwater visual SLAM.

An alternative way to handle outliers in pose-graph SLAM is presented in [39], which uses dynamic covariance scaling based backend for map optimization. This work is an extension of [44] which uses separate computations for determining switch status for switchable constraints. The main improvement is the ability to dynamically scale the covariance while keeping the computation burden down.

Classical graph SLAM algorithms avoid wrong data-association through the use of powerful frontend place recognition system. However, even for human beings there exist some environments where perceptual aliasing is almost impossible to avoid. Latif *et al.* through their work on robust SLAM propose a way of identifying outliers through general consensus of graph constraints [40]. Any constraint that a significant increase in the residual error is removed together with the associated link. Here, unlike switchable constraints in [39], individual loop closure assessment is not required rather a cluster based consensus test is used to identify wrong loop closures.

An alternative to categorizing loop closures as right and wrong is proposed by Olson and Agarwal in [41]. Loop closure errors are characterized in a more expressive manner that account for non Gaussian behaviours using a max-mixture of Gaussian error. Accordingly, a ‘slip or grip’ odometry error is represented by a multi-modal distribution while loop closures are accompanied by a ‘null’ hypothesis. As a main distinction from [40], this approach does not require approximating sum-mixture by identifying clusters of mutually consistent loop closures.

2.3 Sonar Sensors and Feature Extraction

Underwater transmission of high frequency electromagnetic spectrum suffers from high attenuation which limits the working range of most popular sensors such as optical cameras and lasers. However, these sensors are still in use for applications that are bounded within a smaller workspace. On the other hand sound, being a mechanical wave, propagates better in water than in air.

Sonar, which is an acronym for SOund Navigation And Ranging, is a principal data acquisition sensor in underwater. Though, the main purpose of sonar is for data collection, it can also be used as a navigation sensor which can be used to observe environmental landmark features. Broadly speaking there are two main category of sonar: *passive* and *active* sonars. *Passive* sonars are able to sense an acoustic energy originating from targets in deep water. Hence, they are mainly used for military applications. *Active* sonars, on the other hand, sense the environment by sending out pulses of acoustic signals and recording part of a reflected signal coming back to the sensor head. Active sonars are widely available for both military and civil use and they are our main interest in this section and the remainder of the thesis.

Passive sonar sensors return a huge amount of information as a sonar image. In order to extract useful features out of this information the image has to go through a feature extraction process. There are various techniques which can be used for image feature extraction across multiple domains. Next we will discuss some of the research work in this area.

2.3.1 Feature Extraction

Feature extraction is a well known concept in pattern recognition, machine learning or robotics. It can be considered as an important prepossessing stage which can bring a dimensionality reduction with the very minimal loss of useful information. These features could be global as in image retrieval applications where colour histograms and variations are used to represent an image, or local which could discriminate portion of an image from the rest. In this section our interest is limited to those local features which are introduced due to the appearance or absence of a physical structure in underwater environment. Generally these physical changes introduce a change in properties of the image such as: shape, texture *etc.* A typical feature extractor (also called detector) take a measurement from an image region as a descriptor. In the simplest form such a descriptor might carry the intensity of some bright pixels and/or their location in the image.

A feature can represent a set of well localized anchor points, which can be used to match consecutive images. In such set-up, it is not important to know what these features represent. However, image or object representation requires a feature that can assist in understanding the scene or recognition of an object. Ideally, a good feature corresponds to a semantically meaningful object part, is this all of the conditions for a good feature? Tuytelaars and Mikolajczyk devised six criteria that an ideal local feature descriptor need to satisfy [45].

Properties of a good local feature

According to Tuytelaars and Mikolajczyk, a good feature should have to satisfy:

a) *Repeatability*: most of the image processing applications in robotics involve matching a regions in consecutive frames. Additionally, frames taken from similar position are supposed to match closely in the absence of a moving object. However, for successful matching of two related frames, enough similar features must exist in both. This will only happen if we have an overlap among the frames, and the feature extractor manages to get similar descriptors in the overlapped regions. Challenges for feature repeatability include: presence of deformation, scaling, affine transform, noise and moving object.

b) *Distinctiveness*: features from different objects should be possible to distinguish and match, hence descriptors need to carry unique feature of a given object.

c) *Locality*: should be local in order to get persistent features even when there is a view point changes. Global features are most likely to be affected by occlusion.

d) *Quantity*: the number of detected features should satisfy the subsequent usage; it shouldn't be too small missing out some part of the scene, or too large which might make any subsequent task computationally expensive. Too many features is undesirable in our application of sonar based SLAM due to additional update and data association requirements.

e) *Accuracy*: accurate localization of features in an image is another important factor which translates into proper localization of targeted object or external landmark.

f) *Efficiency*: most robotic applications are time-critical; therefore, efficiency of feature extraction remains to be very important.

Currently available feature descriptors are able to satisfy one or more of these criteria. However, when we are dealing with sonar image, for instance images form *forward looking sonar* (FLS), the problem of feature extraction gets more complicated. This is mainly due to the lack of good spatial resolution in sonar images. Additionally, the low signal to noise ratio makes it difficult to use typical feature extractors that are used for visual images. Therefore there is a very limited range of choices when it comes to sonar feature extraction which are mainly based on segmentation and thresholding process. Some of the most commonly used feature representation in the literature include point, lines, blobs, and regions.

Types of feature representation

As we have mentioned it before, feature representation is a vast area of research across various disciplines; however in this section we will limit the review on techniques that can be applied on sonar images. These features mainly arise from intensity, since we don't have colour information in sonar images. Additionally, as we have discussed sonar images are highly corrupted by noise affecting the required feature accuracy prematurely. Yet it is sill possible to extract features from sonars. Next we will present some of these sonar feature extraction techniques.

A. Point and blob features

The most common point feature in use is corner. Corners can easily, and accurately be localized making them attractive sonar features. One of the early work by Wang *et al.* put corner detection techniques in to two classes: template based and geometry based [46]. Template based corner detection uses designed templates of a corners. Then a correlation measure is taken by moving the template window throughout an image to get areas where a corner exists. The main drawback of this method is the requirement of proper models of each and every type of corners that might exist in an image [46]. Geometry based corner detection depends on measuring the differential geometry features of corner points. Some methods in this category are called gradient based corner detection where corner point is defined as the junction of two or more edge lines which can be identified using gradients. There is also a second group of methods called topology corner detection which assumes a corner as the interior geometric feature of an image surface. The third category is auto-correlation based corner detection. An auto-correlation corner detection considers a local window in an image and shifts the window by a small amount in various directions. Then a decision on corner detection is given in places where there is a huge change in intensity.

One of most commonly exploited point feature descriptors is Harris corner [47]; it is also the most accurate corner detector [45]. Harris and Stephens, in [47], managed to represent an intensity variance within an image analytically using second moment (also called auto-correlation) matrix M. This is obtained by applying Sobel operator as a gradient function on an image I smoothed by Gaussian kernel $g(\sigma)$ as,

$$M = \sigma_D^2 g(\sigma_I) * \begin{bmatrix} I_x^2(x, \sigma_D) & I_x(x, \sigma_D) I_y(x, \sigma_D) \\ I_x(x, \sigma_D) I_y(x, \sigma_D) & I_y^2(x, \sigma_D) \end{bmatrix}, \tag{2.1}$$

where

$$I_x(x, \sigma_D) = \frac{\partial}{\partial x} g\left(\sigma_D\right) * I\left(x\right), \tag{2.2}$$

$$g(\sigma) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right). \tag{2.3}$$

The eigenvalues of this matrix represent the amount of signal change in the two principal axes. Since corners raise a significant number of variations in both directions which account for image location where both eigenvalues are large. Additionally, Harris in [47] proposes a cornerness measure R that is computed by,

$$R = \det(M) - \lambda \cdot \text{trace}(M), \tag{2.4}$$

where typical value of λ range between 0.04 and 0.06. It also possible to follow this detection by local maximum suppression or achieve sup-pixel localization of corner. Harris corner descriptors are invariant to translation and rotation. However, it was Mikolajczyk and Schmid who introduced a scale invariant version called Harris-Laplace/Affine [48].

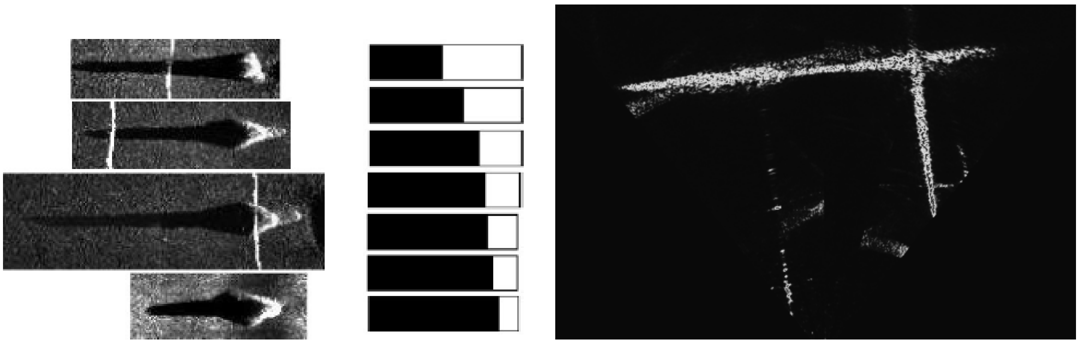
An other type corner or blob detector is Hessian matrix H. It is derived from Tylor expansion of an image intensity function $I(x)$[48]. The Hessian matrix is given by

$$H = \begin{bmatrix} I_{xx}(x, \sigma_D) & I_{xy}(x, \sigma_D) \\ I_{xy}(x, \sigma_D) & I_{yy}(x, \sigma_D) \end{bmatrix}, \tag{2.5}$$

where I_{ij} is second-order derivative of Gaussian smoothed image. The local maximum of the determinant and trace of Hessian matrix represent a blob like structure in an image. This determinant is also the core of well know Speeded Up Robust Features (SURF) [49]. The famous Scale-invariant Feature Transform (SIFT) is also an other blob detector [50]. It is based on an approximated Laplacean of Gaussian filter called Difference of Gaussian (DOG) filter. Additionally, it uses a pyramid of images to handle scale variations.

These point feature representation techniques are highly used in most image processing task mainly due to their accuracy. Moreover feature descriptors like SIFT and SURF are found to be rotation, translation as well as scale invariant and they can be extracted efficiently for real time applications. However, these features require the image to be rich in intensity variation which unfortunately is not the reality in sonar images.

An alternative convolution based feature descriptor called Haar feature, which is successfully used to represent facial features [51]. This involve the use of a cascade of rectangular Haar-like features to move around the image and compute the response. This response is encoded as the orientated contrast between regions in the image. Its simplicity and a possible fast integral image based implementation makes it attractive. Sawas *et al.* [52] successfully used these features for detection and classification of objects in side scan sonar. They use boosted classifier based on the work of the Viola and Jones, which learns the target and the noise classifier beforehand [51]. The major draw back of such system is their lack of adaptation to a new environment as it requires initial extensive training phase. Later Aulinas *et al.* also used similar features in their implementation of side scan sonar based simultaneous localization mapping (SLAM) [53]. The main reason for this success relies on the fact that objects in side scan sonar appear as a bright spot followed by a shadow, which makes them very effectively recognizable through Haar detector. However, in the case of forward looking sonar this advantage is lost since shadows in forward looking sonar



(a) Side-scan sonar → Harr-like feature [53].

(b) Object in typical FLS image.

Figure 2.4: Shadows in sidescan sonar and forward looking sonar images. a) Shadows in side scan sonar has a Haar-like structure with a portion of beright patch followed by a dark shadow. b) There is no direct association between a bright patch and a shadow in Forward looking sonar mainly due to the horizontal sonar set-up.

doesn't necessarily describe the type of object they are generated from (see Figure 2.4).

Even though the above mentioned feature descriptors are accurate and efficient in visual image processing, sonar images are too noisy and less rich with point features that can be localized accurately. From the trend most researchers in the field prefer to use segmentation to localize interesting features [32, 54, 55, 56]. These techniques describe a segmented image using its geometrical and intensity characteristics of segmented regions.

Tena *et al.*, in their early study to evaluate robustness of inter frame features aim to identify man-made objects consistently in a sequence of sonar images [55]. An input sonar image is initially smoothed using median filter which then passes through two levels of thresholding. The output of the second stage is used as a seed for region growing. They evaluate feature's robustness using a classifier that can identify objects based on their local features. Among the features that they used include: area A, perimeter P, compactness C, major-axis length, mean intensity, variance, contrast, first and second moment. These features are computed either from the geometry of segmented area image or from the intensity of input image masked by the segmentation output. Similarly, Ruiz *et al.* , in their underwater SLAM algorithm, use multiple stage segmentation statistics as their feature descriptors [32]. They have used adaptive double thresholding which is adapted from [56]. The first layer of segmentation is used to identify areas of the image with a potential target. This requires smoothing followed by adaptive thresholding based on intensity histogram. We have got an especial interest in such type of hierarchical ROI selectional mechanism, which is commonly known as focus of attention (to be discuss further in Section 2.4), mainly because it can reduce the required computational burden for executing the feature extraction task. The second layer requires segmentation with in the area of interest. A smart system in [56] predicts the location of these ROIs in subsequent frames using Kalman Filter. Finally, features are represented by the relative distance of centre of mass with respect to the vehicle, the areas of the segment, and the first invariant moment. Most of the these, a decade old and fundamental segmentation based techniques are still in use; for instance, a recent work from Aykin and Negahdaripour, in [54], uses k-means clustering to segment part of the sonar image which includes 1-2 % of the highest intensity pixels. The algorithm iteratively compute the value of k using a threshold on the minimum distance between cluster and minimum standard deviation with in a cluster.

There have been different application specific evaluation techniques to identify best corner, blob or region detectors. Tuytelaars and Mikolajczyk [45] talks about some of these technique. But the most common evaluation technique is a comparison of synthetic image with one perturbed in noise, varying angle, length, contrast, noise, blur etc. [45]. An alternative technique involves developing a ground truth metric representation of a scene before processing.

Gradient based techniques fail to provide meaningful features on forward-looking sonar images, mainly due to lack of enough sonar features from the open sea. Additionally, features from actual structures tend to vary tremendously with a slight change of view. When it comes to segmentation based features the major challenge tends to be their computational greediness. Practically speaking, in structured environment most of the visible feature in the acoustic image tend to be straight line which can be better expressed using line segments. Hence we have the other major category of sonar image feature, line features.

B. Line feature

Lines are the most apparent features in sonar images taken from man-made structure dominated environment. Ribas *et al.* uses a voting scheme to identify walls in a structured environment [2, 14, 57, 58]. A continuous beam of sonar data from a mechanically scanned imaging sonars (MSIS) first preprocessed, then part of each beams with echo-information is segmented. This is done by thresholding each beam followed by a min-max suppression around peaks. These pixels are assumed to be a representative part of an object in the scene. They use a model where a pixel in polar coordinate represents an arc in Cartesian coordinate system defined by angle and range resolution of the sonar. Figure 2.5 shows

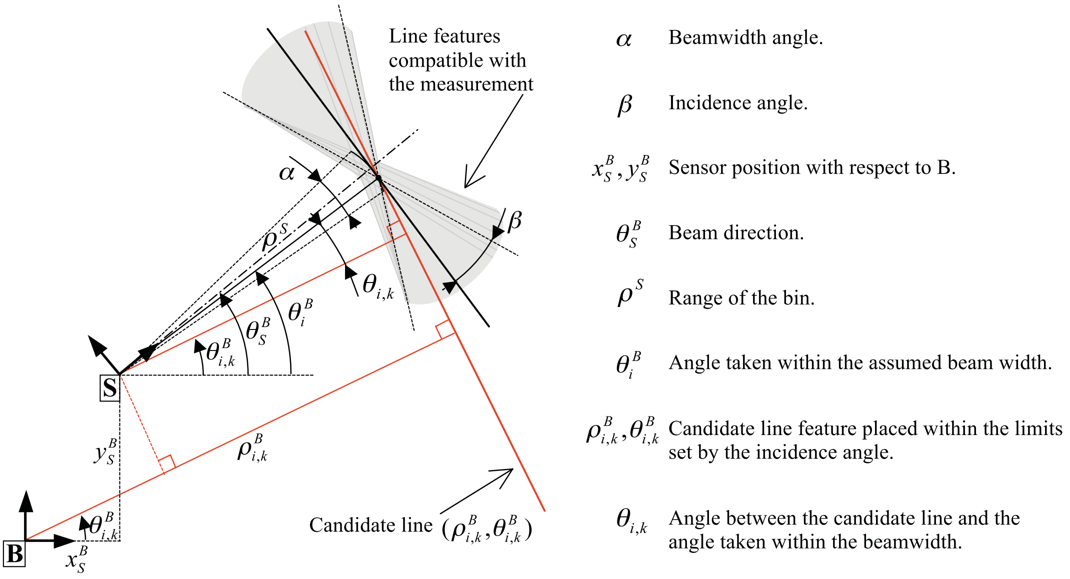

Line features compatible with the measurement
α
β
ρ^S
$\theta_{i,k}$
θ^B_i
θ^B_S
$\theta^B_{i,k}$
S
y^B_S
$\rho^B_{i,k}$
B
$\theta^B_{i,k}$
x^B_S
Candidate line $(\rho^B_{i,k}, \theta^B_{i,k})$
α Beamwidth angle.
β Incidence angle.
x^B_S, y^B_S Sensor position with respect to B.
θ^B_S Beam direction.
ρ^S Range of the bin.
θ^B_i Angle taken within the assumed beam width.
$\rho^B_{i,k}, \theta^B_{i,k}$ Candidate line feature placed within the limits set by the incidence angle.
$\theta_{i,k}$ Angle between the candidate line and the angle taken within the beamwidth.

Figure 2.5: Model of line feature in forward-looking sonar images [2]. Features are extracted relative to base frame **B**, while measurements are taken relative to sonar frame **s.** α represents the beam resolution, β represents possible incidence angle range of the line feature.

this model where line features are represented with respect to base frame **B** as the distance to the line and orientation (ρ^B_k, θ^B_k), which is similar to $(\rho^{S_j}_k, \theta^{S_j}_k)$ in the sonar frame **S**. A measurement is modelled by an arc of beam width α at an angle θ^B_S and distance ρ^S. However in addition to the line tangent to the arc all lines with a significant incidence β can contribute to this measurement. Therefore, a point in a sonar image contributes to the vote of all lines passing through the shaded region in Figure 2.5. This requires an adaptation of the Hough transform, which makes it slower for on-line applications. Ribas *et al.* also proposed a technique for estimating line uncertainty based on the thresholding confidence and the distributions of votes close to the line in the Hough space [2].

Ribas *et al.* also proposed a new voting based uncertainty computation method. One advantage of this covariance estimation technique is that it can adapt to image types without requiring change in parameters after initial set-up. This technique is discussed in depth in Section 3.2.3.

C. Other features

When it comes to working with 3D sonar data a much better compression can be achieved through the use of planes. Pathak *et al.* , in [59], uses a plane-based registration for underwater 3D mapping using sonar data. They use a minimum uncertainty maximum consensus plane matching technique presented in their previous work [60], which applies the technique for mapping indoor environments using land robot. Accordingly, two planes which are extracted from overlapping 3D point clouds are matched by maximizing the overall geometric consistency. The method computes the rotational and translational scan pose drift as well as providing the uncertainty of this pose-registration. Based on their result, this approach is reliable and fast for mapping underwater structures from sonar data.

Buelow and Birk claim the use of 3D mapping is a necessity for intelligent operation of AUVs in an environment full of complicated man-made structures [61]. It can also make piloting task easier for ROV operators in such environment. Three commonly used 3D mapping techniques (ICP, plane-based registration and spectral registration) are compared in their experiments. Their proposed spectral registration technique, which is based on phase only matched filtering, is fully implemented using Fourier transform. The method make use of the fact that two shifted signals share the same spectrum magnitude while the shift is carried through their phase difference. A simulated test and a run on real dataset collected from a flood gate lock structure in the river Lesum shows the superiority of spectral registration technique even in the presence of high noise or large spatial distance between scan positions [61]. Similarly, Hurtos *et al.* propose the use of Fourier-based registration technique for underwater mosaicing and motion estimation using forward-looking sonar [62]. Accordingly, the use of Fourier-based technique provides an improved performance in the alignment of both consecutive and non-consecutive views relative to state-of-the-art region based approaches [62]. Additionally, unlike most other registration methods Fourier-based sonar registration is capable of providing a good estimation around featureless regions of an environment. This technique uses Fourier-based image registration to determine local sonar movement in 2D plane which is used to generate a pose graph for global pose alignment and subsequent mosaicing. Hurtos *et al.* also propose an improved uncertainty estimation technique based on the work in [61] which makes convenient for use in a SLAM framework.

Among the techniques presented so far, line feature based techniques stands out for representing man-made structure from forward-looking or micro-bathymetry imaging sonar. When using such type of sensor around man-made structure one of the most visible features is the line feature. However, researches on range scan has shown the limitation of Hough transform when it comes to extraction speed [63]. Additionally, a voting based uncertainty computation could further add computational complexity making the system an unusable for online SLAM based navigation application. In Chapter 3 we will discuses more about alternative feature extraction techniques and do a comparison among these methods.

2.4 Active Vision

In addition to having a robust feature extraction technique, for a good SLAM navigation capability, a vehicle needs to observe those feature rich landmarks continuously. In all SLAM applications discussed previously the sensor configuration is passive where the vehicle observes anything that happens to land with in its field of view. However a good SLAM based navigation system need to deliberately look for those good features using active vision in order to achieve improved SLAM performance. Bajcsy [64], being one of the early researcher in the field, defined active vision as "a study of modelling and control strategies of perception." Accordingly, it is an application of intelligent control theory which includes reasoning, decision making and control. While Chen *et al.* [65] consider active vision as planned sensing or acquisition of perception on the operating environment, which requires strategies of sensor placement and sensor configuration.

One typical ambiguity that needs to be clarified before moving any further is the difference between active sensors and active sensing. An active sensor is a sensor that transmits, mostly electromagnetic radiation or sound wave, in to the surrounding and listen to a reflection to determine the proximity of obstacles in the environment[64], *e.g* radar, or microwave based sensors. On the other hand, active sensing tries to adjust sensor's state and parameters according to sensing strategies, and it doesn't necessarily require active sensor.

Typical image processing tasks (*e.g. object recognition, multidimensional segmentation*) can be improved by using images acquired from different vantage points. Robots with an active vision system are able to actively place sensors at several strategic viewpoints, yet the challenge becomes the decision on where to place them [65]. For instance in [64], efficient sensor placement reduces the number of range images required to reconstruct a 3D object from seventy down to ten. Additionally, problems that are ill-posed, non linear or unstable for a passive observer become well-posed, linear and stable for an active observer. Furthermore, the requirement for multiple views can be avoided by having some special view points as discussed in [65]. Chen *et al.* *[65]* provide an extensive survey on active vision covering the different methods and possible applications. Next we briefly discuss the various active vision applications in robotics and the main techniques available in the literature.

2.4.1 Applications of Active Vision

As mentioned in [64] early active vision works were targeted to find a better way of resolving typical computer vision challenges: shape from shading, shape from contour, shape from texture, structure from motion and optical flow. These works were only benefiting from plenty of data arriving from multiple views. Chen *et al.* [65] gives a good classification of contemporary active vision systems based on their application. We will use a similar approach and present the works done in application areas relevant to our proposed system: purposive sensing, robot localization mapping, navigation, exploration or tracking [65].

Purposive sensing and object modelling

One area of active vision application is purposive sensing which is the acquisition of image for better robot understanding. Robots understanding of an object could be improved by having more images of the object; however, a non complementary image addition will result in unnecessary redundant information which will only add computational burden. Therefore, a choice has to be made on the type of new image and a decision to stop after collecting enough number of images.

Li and Liu [66] propose an active vision system that can determine a good image acquisition point for efficient measurement and reconstruction where the result can be used for reverse engineering of CAD models from existing objects, computer animation and virtual and augmented reality. On the other hand, Tsai and Fan [67], used a CCD camera with zoom control and a line laser in order to measure a large surface or local steep profile. The camera zoom is actively controlled according to the slope distribution of the object. In [68], a study on the number of possible vintage points of image acquisition for 3D reconstruction demonstrated that there are some specific viewpoints which gives a better reconstruction. Additionally, it is shown that regions close to these points give similar performance. A non model-based system by Jonnalagadda *et al.* [69] defined four stages for viewpoint selection for 3D reconstructions: local surface feature extraction, shape classification, viewpoint selection and global reconstruction. Accordingly, an active vision system called biclops with two cameras mounted on independent pan & tilt unit is used to extract 2D and 3D features. These features are assembled in to a geometrical primitives, which are classified in to local shapes and later used to develop a hypothesis for the global shape of the object. Then the next view point is selected to verify this hypothesis and if successful, the information about the global shape is saved otherwise a new hypothesis is generated from old data. As stated in [65] non-model based 3D reconstructions are automated using incremental model of the object. An optimization strategy is applied on such models to select the best scanning viewpoints and generate collision-free scanning trajectories[65]. Most non model based 3D reconstruction systems are a result of multiple partial models, and the combination does require a registration which is mostly done offline[65]. Therefore, this causes missing information that in turn creates a hole on the reconstructed object[65]. Some of the works, including one by Chang and Park [70], are intended to determine the additional scanning orientation which covers the missing areas of the 3D object.

Imaging systems for purposive-sensing are mostly formed from multiple types of camera system for instance, MacKinnon *et al.* [71] uses a camera composed of touch probes and laser scanner for photogrammetry, while Huang and Qian [72] used tactile point sensor instead of touch probes to improve the measurement speed and efficiency. Treuillent *et al.* [68] combined structural geometry and texture for 3D reconstruction.

Hollinger *et al.* [73] uses a sonar based active view planning in underwater environment. Their work aims to adaptively plan views of an AUV to improve the quality of object inspection rather than maximizing the accuracy of a given data stream as in most active vision systems.

In summary, purposive sensing, which provides view points that complement and complete the view, is less applicable for a SLAM navigation system that does not require to explore the surrounding. However, in cases where the outcomes of the SLAM map is significant it can provide a complete map for other robotic tasks.

Site modelling

Site modelling is an other application area of active vision where the goal is to minimize unobserved part of an environment [65]. Static scenes are handled using perception/action cycle which involves outlining perception strategies for scene exploration and generation of camera motion using visual servoing [65]. Se and Jasiobedzki [74] devised a system called Instant Scene Modeller (iSM) for creating photo realistic 3D models of environments using stereo pairs. It uses an iterative sensor planning and sensor redundancy to enable robots to efficiently position their cameras with respect to task [65, 74].

Site modelling is ideal when a complete exploration of a site is required. The complexity of site modelling grows as the size of the site, which makes it difficult to employ these techniques for a large underwater environments without a prior knowledge about the environment. Additionally, to employ an active site modeller for navigation the goal has to be modified to improved navigation instead of covering the whole area.

Object search

In object search, it could also be landmark search, the task is to find an object in a known or an unknown environment under partial or full model based vision framework [65]. This operation mostly arises in order to assist other applications, for instance, in site modelling to fill a missing part of a site it might be handily to search for some feature which is in proximity that part of the environment. Shubina and Tsotsos [75] argue attentive systems (to be discussed later in the chapter) require the search for object/features in order to accomplish top-down attention. They address the problem of finding objects in an unknown space as an optimization problem. They optimize the probability of finding the object at a fixed cost of total number of robot actions required using the joint probability of object detection given an operation, and the probability of target distribution thought the whole region.

Object tracking

Active tracking requires continuous adjustment of visual systems in order to obtain extra information or be able to perform tasks efficiently [65, 76]. The main task of most active tracking systems is fixation, where gaze of vision system is dictated in a way that can keep a target in field of view as long as possible. Barreto *et al.* [77] used an active stereo rig to simultaneously track multiple moving targets (up to 3 targets). Their system managed to track points with unconstrained motion in a plane by adjusting the camera setting using consistent homography between stereo images. A fixed homography is specified during the hardware design, and later this 4 Degrees of freedom (DoF) camera rig is adjusted to keep this value. Similarly Zhou and Sakane [78] defined the baseline of the stereo rig based on the target size and the distance at which the object of interest is existing. Chen *et al.* [65] also mention works that managed to move an imaging system in proportion to an object moving at linear velocity, keeping a fixed distance between the observer and the target. In some other works, the task of active tracking in an unknown environment involve: selecting the object of interest, building a 3D model and using the model to track the object [65].

Mobile robotics

The last active vision application area that we want to discuss here is mobile robotics. Mobile robotics requires the deployment of autonomous robots in to partially or fully unknown environment, where a robust navigation system is crucial. Mobile robots are equipped with various localization sensors, however, these sensors might not be accurate enough or efficient in all time and in every condition. To alleviate this problem most navigation systems are coupled with vision system and active vision can add more autonomy to this systems and the robots.

One of the fundamental problem in mobile robotics, as discussed in Section 2.2, is localization and mapping. It involves determining the position of the robot and/or objects in the environment based on sensor data. Other robot actions are planned according to this location. Typical localization and mapping systems do not integrate sensor alignment or placement to improve the efficiency of the system or provide an accurate map. To fill this gap Gonzalez-Banos and Latombe [79] managed to include a scheme for driving the robot in to the next good position based on the expected amount and quality of information to be revealed in each new location. Their work is intended on improving the quality of a global map generated from a local measurement using modified SLAM algorithm. In particular the selection of the next good robot pose requires addressing the concept of safe regions (to identify regions free from any obstacle based on historic sensor reading) and sufficient overlap among successive local models (in order to allow registration of successive views under position uncertainties). Not all active explorations are efficient, a brute force way of improving a map requires an extensive exploration following paths as simple as lawn mower patterns. However the rate of convergence of a map is highly dependent on the order of the observation [80]. Sim and Roy [80] argue that conventional Kalman filter based SLAM converges in the limit of infinite time and data. They proposed a technique that can lead to global optimization in a way that results in a more accurate map. The global planning strategy that they use tries to enforce a loop closure whenever required resulting in faster convergence to the correct map [80]. Chen *et al.* [65] mention some of the works that generate semantic maps of an environment using active vision based SLAM.

Leuang *et al.* [81] use trajectory planning Smooth and Mapping (SAM) based SLAM, in order to minimize uncertainty of map estimate and to maximize coverage. Similarly, Zhang *et al.* [82] criticize most robot path planning techniques for just being a problem of minimum distance optimization, even though the primary objective of exploration is gathering as much information as possible. These are just few of the works that employ active vision for mobile robotics. Later we will discuses some more works that employ a specific type of active vision technique called focus of attention but first let us introduce some of the most widely used active vision techniques.

2.4.2 Techniques of Active Vision

There are a wide range of active vision techniques in the literature ranging from statistical approach to constraint based viewpoint selection. Some are general while other are tailored for the specific application. Here we discuss some of the basic approaches in active sensing.

Expected model based approaches

Techniques in this category try to develop an expected geometrical model of a scene to decide on the next best view. As in [65], the first step is 2D or 3D surface feature extraction which is assembled in to geometric primitives. Then local geometries are used for shape classification which is later used to determine viewpoints based on a hypothesized model. Once the model generation is complete the next best view is determined using a visibility map. It describes the possible set of global visible view directions for a surface [65]. Liu *et al.* [83] treats the problem of building visibility map as an inverse of hidden surface/line removal problem, where the later is given a view direction finding the set of geometric primitives visible along that direction. Hence, the visibility map problem is defined as given an entity in the scene it is finding the set of viewpoints from where the given entity is completely visible [83]. A projection of this viewing direction on the surface of a unit sphere is called global visibility map (GVM).

An other way of solving model-based view planning involves formulations developed for "minimum wall guard problem" [65]. In this definition, the problem is finding the minimum set of points on a plane, so that every wall is guarded (satisfying the constraint of visibility, resolution and field of view) at least by one of these points in the presence of an obstacle. Generally, to solve the minimum wall guard problem a first candidate guard set S are identified. Then the number of walls that can be guarded using a subset of guard points $s \in S$ is determined. The final decision is to determine the minimum subset of S, that satisfies the three main constraints mentioned above [65].

Yet another model-based approach uses spatial tessellation around an object [65]. A spherical or cylindrical region around the object is tessellated and each grid point is considered as a possible sensor pose for viewing the object. Then these grid points are classified as either viewing or void volume.

A model based approach can be a good alternative in environment where there exist a prior knowledge on the type of object available. Additionally, the goal of model based active vision is to provide a full account of an object rather than selecting a view for a high quality information that can improve SLAM navigation, for instance.

Graph based approach

An alternative technique for sensor placement uses graph theory in a scenario where an object's size and distance are previously unknown or variable [65]. In such method viewpoints are computed based on visibility and reliability according to an offline process on features extracted from object [65]. Accordingly, this viewpoints are connected using the shortest path computed using techniques that solve the travelling salesman problem as shown in Figure 2.6. As discussed in [84], a graph $G = (V(G), E(G), w_E)$, is constricted using viewpoints V_i, shortest path E_{ij}, with a weight w_{ij} proportional to its length. Hence, such sensor placement plan provides possible viewpoints and all the minimum cost collision free paths among this viewpoints. Finally, the objective is to achieve the lowest travelling cost T_{cost}, required to complete the whole procedure of planning, and execution of robot action. Such offline pre-processing is undesirable in application such as ours which requires an online reactive planning.

Statistical approaches

A wider range of recent active vision systems are developed based on statistical techniques, this goes in parallel with other robotic interests including localization and mapping. Farshidi *et al.* [85] approach the object recognition problem using an active sensing camera set-up. They consider the task of object recognition as state estimation, where the identity and pose of the object is the unknown states. Hence, they manage to define active object

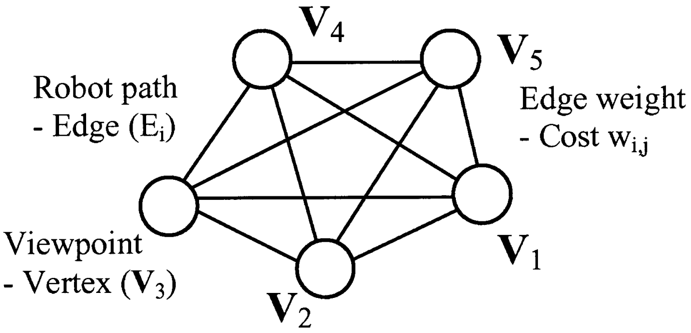


Figure 2.6: Sensor placement graph [84].

recognition in a probabilistic framework under the assumption of Markov processes. This framework allows the incorporation of uncertainty arising from the camera noise, illumination variations, object and camera position [85]. Such a system uses observations $g_{0,1,\ldots,n}$ and sensor parameters $a_{0,1,\ldots,n}$ up to the current time in order to estimate the unknown state s^n using Bayes' formula as:

$$p(s^n \mid g_n, a_n, \ldots, g_0, a_0) = \frac{p(g_n \mid s^n, a_n) p(s^n \mid g_{n-1}, a_{n-1}, \ldots, g_0, a_0)}{p(g_n \mid a_n, g_{n-1}, a_{n-1}, \ldots, g_0, a_0)}. \tag{2.6}$$

The state $s^n_{i,j,k,l}$ is composed of object class o^n_i, object pose number φ^n_j, k^{th} occlusion level in the view of camera-1 ξ^{1n}_k and that of camera-2 ξ^{2n}_l. The first two are time-invariant *i.e.*, no transition occurs between states with different object class and/or pose. The observation vector g is formed using coefficients extracted by application of principle component analysis (PCA) on the images from both camera. The camera control vector a is defined as $a = [\theta_1, \gamma_1, \theta_2]$, where θ and γ are the pan and tilt for either camera. The key part of such type of active sensing is the evaluation measure employed to select the sensor control command. Farshidi *et al.* [85] discuss two indices of observation quality metrics to determine the next best view, Mutual Information (MI) and Cramer-Rao Lower Bound (CRLB). MI is a measure of the reduction in uncertainty in s^n due to the observation g and maximizing its value yields a good camera position. On the other hand, CRLB is a measure for estimation variance which is minimized to provide a good quality observation.

Constraint based cost function

Constraint based cost functions are one of the early techniques used for active vision where the aim is to satisfy constraints like focus, resolution field of view, visibility *etc.* Chen and Li [84] in their proposal for automatic sensor placement in model-based robot vision, they summarized and analysed these constraints. Accordingly, constraints can be categorized in to geometrical, optical, re-constructive and environmental constraints. In this set up each constraint limit the viewpoint space to some smaller region, and a combination of multiple constraint give rise to an intersection of this regions. As in [65] a good sensor placement space that satisfy visibility (g_1), viewing angle (g_2), field of view (g_3) and resolution (g_4) constraints is given as:

$$V_{placement} = V_{g_1} \bigcap V_{g_2} \bigcap V_{g_3} \bigcap V_{g_4}, \tag{2.7}$$

where, V_{g_i} is a volume of the space that satisfy the criteria $g_i \geq 0$, which is mostly a Boolean function; for instance, the visibility of a point on an object depends on incidence and the surface normal and can be defined as Boolean function by putting some threshold on these measures. The aim of such techniques is to maximize an optimization function formed from these constraints (as in equation 2.8), or to minimize a cost function involving the cost of sensor movement. One of the cost functions mentioned in [65], to determine the next best view, tries to minimize a cost by reducing the total area observed or camera displacement while avoiding unreachable viewpoints.

$$h = \max(\alpha_1 g_1 + \alpha_2 g_2 + \alpha_3 g_3). \tag{2.8}$$

In an other model based approach, Chen and Li [84] defined their model-based sensor path planning using the following procedures:

- generate a number of viewpoints;
- reduce redundant viewpoints, which involve discarding viewpoints with no extra information;
- if the placement constraints are not satisfied, increase the number of viewpoints;
- construct a graph corresponding to the space distribution of the viewpoints; then
- find a shortest path to optimize robot operations.

Yet another category of active vision technique is focus of attention which is motivated from primates cognitive based vision system. Due to our special interest, a more detailed review of these techniques is presented next.

2.5 Focus of Attention

An information management behaviour of primates that gives them the capability to focus on the most relevant pieces of an information is called attention [86]. Even though attention could be associated with different sense organs, here it means visual attention and its counter part image based robotic visual[1] attention. Researchers in this field are motivated by either the need to understand and hypothesize the principle for this behaviour in human, and later test this hypothesis, or the potential application of this behaviour in artificial systems [86]. We belong to the second category, we intend to introduce similar capability in underwater robotics. In this section of our review, we cover some the cornerstone techniques and applications of visual attention starting with a brief explanation on the different categories of attention, followed by an explanation on the approach that researchers from different disciplines (psychology, computational neuroscience or computer vision) take towards the subject. Then we base our discussion on a commonly accepted structure of attentive system and some of the works that use attention for mobile robotics applications.

2.5.1 Categories of Attention

Human visual attention mechanisms can take either a *bottom-up* or *top-down* approach, and these dictate trends in computational models of attention. Similarly the concept of *overt* and *covert* attention is another key based on which computational models can be categorized. Understanding the distinction among this categories is key for identifying the possible technique for a given application.

Bottom-up vs top-down attention

Bottom-up attention refers to an attention triggered by stimuli in a visual field, a salient feature attract our attention for further inspection. On the other hand, top-down attention is started with a prior know-how on the object that we are looking for, this could be a knowledge on it's position, characteristic feature *etc*. The former is like searching for something surprising, without having a defined description of the target. However, the latter is like searching for a phone in a messy tabletop.

[1] The term visual is used to describe image based sensing, either from visual camera or sonar sensor.

Overt vs covert attention

A control mechanism of an attentive system can be either overt or covert. Overt visual attention is common interpretation of attention in primates, and it involve a physical movement of the imaging system; for instance, in humans it involves the movement of eye, head and/or body. Covert visual attention, however, require mentally focusing attention on a particular stimulus in a scene. Most artificial attentive systems reside in the later category, where their main purpose is similar to selecting a region of interest (ROI) in a given scene [86]. This is mainly due to the additional challenge that an overt attention introduces in construction of a salience map. As discussed in [86] this includes change of reference due to a moving camera which forces a fresh salience computation after each camera movement, complexity of bookkeeping of previously attended subjects giving a room for new stimuli, and handling a complete or partial disappearance of objects caused by camera movement.

Space vs object based salience

A key element of most attentive systems, whether behavioural or computational, is the salience map. A salience map defines the level of focus a region or an object require. The former imply each space in a scene is assigned specific salience which is effective in static environment. However, in case of a dynamic scene this will not work since movement of an object might alter salience of a given space. Yu *et al.* [87] argue that object-based salience is much more robust due to the contribution of each component in an object. It also mention that information regarding an object (e.g., size, shape) are more likely to appear in an object-based attention than space-based attention. Additionally, hierarchical selectivity of attention in the presence of structured object, groups of objects or overlapped objects is better handled using object-based attention [88]. On the other hand, object based salience becomes challenging when there exist multiple objects with similar salience causing an oscillation [65]. Therefore, a good salience map need to balance between both space and object based representations.

2.5.2 Computational Models of Visual Attention

Begum and Karray, in their comprehensive survey on Visual Attention for Robotic Cognition [86], suggested a common model of visual attention architecture as shown in Fig-

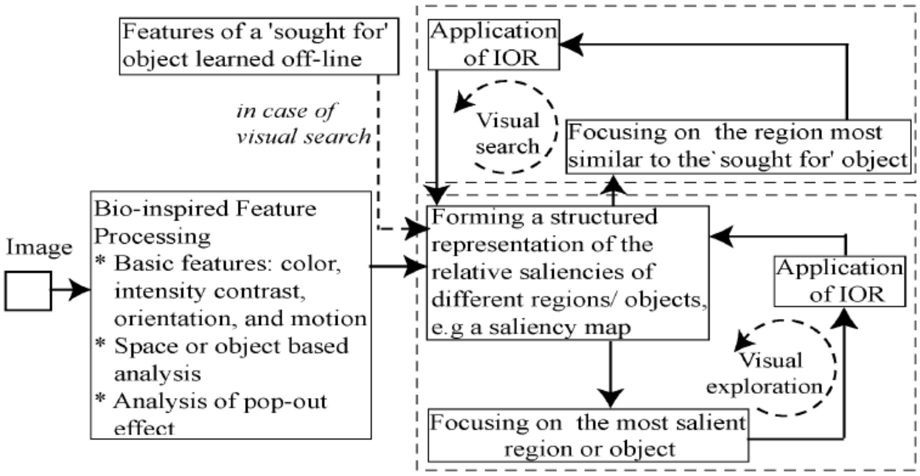

Features of a 'sought for' object learned off-line
in case of visual search
Image
Bio-inspired Feature Processing
* Basic features: color, intensity contrast, orientation, and motion
* Space or object based analysis
* Analysis of pop-out effect
Application of IOR
Visual search
Focusing on the region most similar to the`sought for' object
Forming a structured representation of the relative saliencies of different regions/ objects, e.g a saliency map
Application of IOR
Visual exploration
Focusing on the most salient region or object

Figure 2.7: General architecture of visual attention models in computer vision [86].

ure 2.7. Variations among techniques arise either by addressing individual components using different models or excluding some of the components. The diagram also show two sub sections, first visual search which is activated in case of top-down attention, and then visual exploration which is a bottom-up attention. This later section is responsible for developing either an object or a space based salience map in most applications.

Even though there exist a tremendous variation on the implementations, there are some milestone computational models such as *feature integration theory* [89], *neuromorphic vision toolkit* (NVT) [90], and *visual object detection with a computational attention system* (VOCUS) [91]. These models are considered as the most influential models and are incorporated as part of most other models. Next a brief discussion on these basic technique is presented.

Feature integration theory [89]

Koch and Ullman explained the psychology of human perception and cognition using a feature integration theory, where perception of feature is necessary condition to perceive an object [89]. A recent work by Kootstra [92] adopted this technique for artificial visual attention, where features (see Figure 2.8) including colour, shape, orientation and direction of motion register their salience in separate feature maps. These feature maps include the physical location of features representing objects. Individual feature maps are superimposed to from a single master map, where attention is directed to the physical location with a higher salience using a winner-takes-all (WTA) criteria. Though progress has been

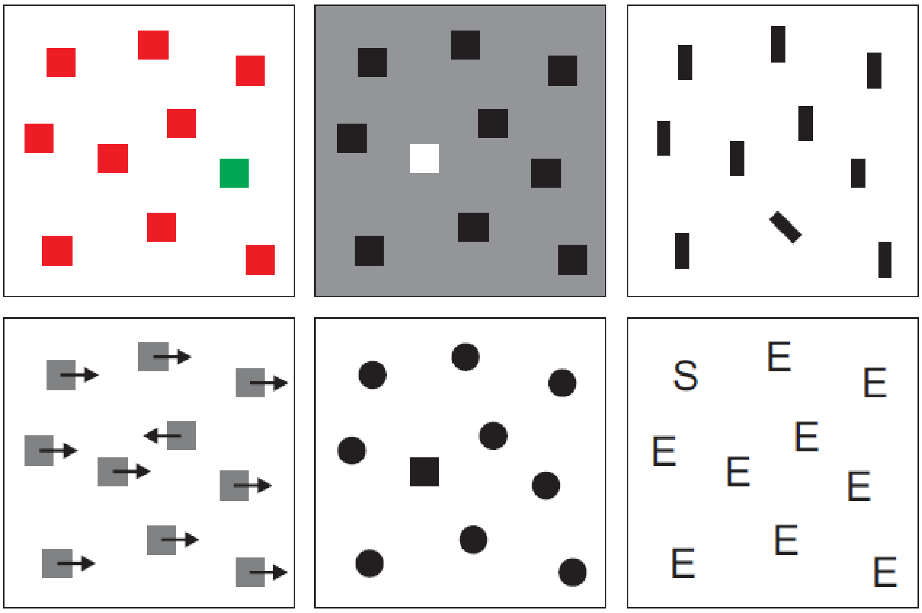


Figure 2.8: A list of basic feature. From top-left to right-bottom: colour, intensity, orientation, motion, and shape feature inconsistencies popping out for focus attention [92].

made in feature based attentive system, it is predominately confined to systems that use a bottom-up approach. Yet, an alternative top-down attention model, *guided search,* has been found to be effective to explain visual search [86].

***Neuromorphic vision toolkit* (NVT) [90]**

Itti *et al.* [90] introduce one of the widely used bottom-up computational model *Neuromorphic vision toolkit* (NVT), which is an extension of Koch and Ullman's [89] feature based approach. One of the main reasons for their wide acceptance is their proposed simple and elegant way of computing salience which they made it available for testing and reuse. Figure 2.9 shows the structure of this model. An intensity normalization is applied on each of the three colour channels of input image to decouple it from intensity, resulting in four tuned colour channels: red, green, blue and yellow. A dyadic Gaussian pyramid is generated for each of these channels. They managed to incorporate a primate like cognition, *centre-surrounded differences* between a 'centre' fine scale and a 'surrounded' coarser scale, to generate intensity based feature maps. This is similar to the way ganglion cells in the visual receptive of human visual system respond. Additionally, colour feature maps are created by applying chromatic opponency to excite one colour by inhibiting an other, while an oriented Gabor pyramid is used to determine local orientation information in orientation based feature map. This map representation result in 42 feature maps: 6 intensity, 12 colour, and

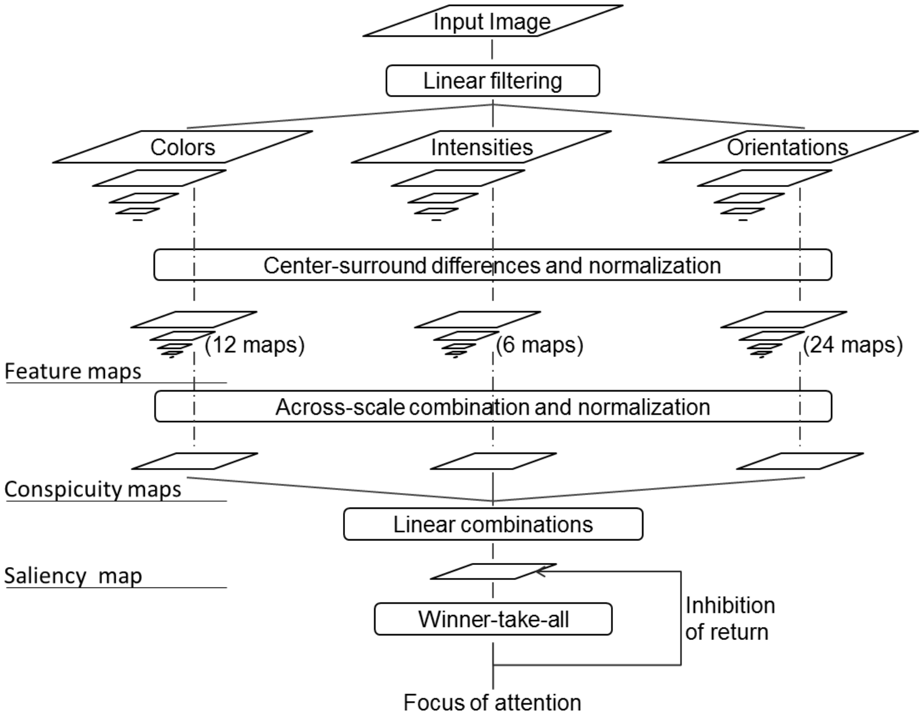
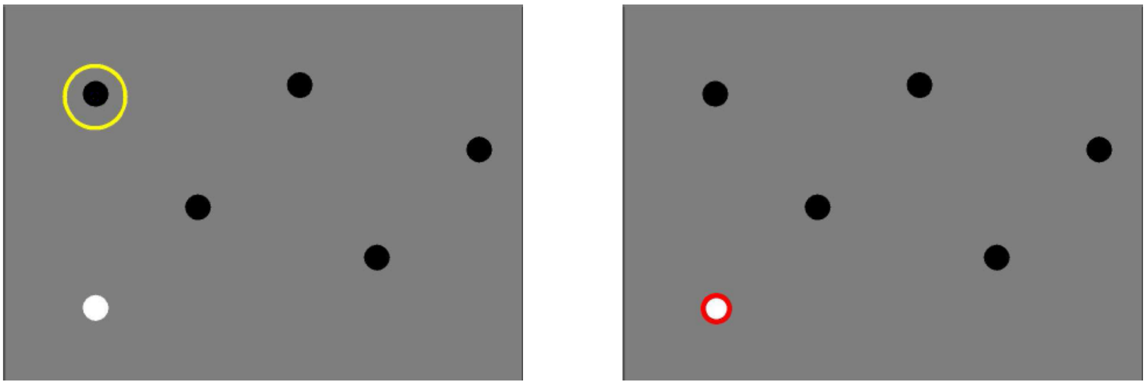


Figure 2.9: Architecture of the *Neuromorphic Vision Toolkit* (NVT) [90]. Multi-scale representation of three features maps are linearly combined to form a salience map which determines the focus of attention.

24 orientation. Then feature maps are normalized in order to compensate amplitude differences among different modalities. This normalized pyramid of feature maps are scaled in to an intermediate level of the pyramid then superposed to form three “conspicuous maps”, each for intensity, colour and orientation. Finally, these three maps are normalized to the same range and linearly added together to form the salience map.

Commonly, after developing a salience map the most salient location is determined using a strategy known as winner takes all. Then the focus is shifted towards this maximum salience location. In order to avoid a refocus on the same space, this location is inhibited for a while using a technique called inhibition of return (IoR). This gives the system a chance to attend a new silence point.

NVT architecture is strongly influenced by primates early vision which makes it a good model as to understand attention in primates. Therefore when used as a computational model it lacks robustness in practical environment as most of the component modules were selected for mere computational efficiency. Though NVT still acts as a major breakthrough as a computational attention model many drawbacks have been picked out by different authors. Among these, Draper and Lionelle [93] argue that NVT lacks robustness to handle



(a) NVT first focus of attention.

(b) VOCUS first focus of attention.

Figure 2.10: A separate on-center and off-center differences in VOCUS result in human like pop out effect, which is difficult to replicate using NVT's single center-surrounding model (Image source [91]).

2D similarity transformations like translation, rotation, and reflection. They blamed the implementation rather than the architecture, proposing an improved system for such scenario which favours precision against speed of computation. However, it is still a fact that the efficiency of this technique has made possible systems such as attention based object grasping system proposed by Rasolzadeh *et al.* [94]. Miau *et al.* [95] combine NVT with support vector machine based object recognition to form biologically-inspired recognition system. They argue their set-up can speed up recognition while preserving the detection rate. Walter *et al.* [96] uses NVT to demonstrate how attentional region selection can improve the robustness of Lowe's *scale-invariant feature transform* (SIFT) based object recognition [50]. On their other work [97], they demonstrated the successful integration of NVT based attention system in assisting manual annotation of underwater video images for objects with a potential interest for human annotator. Some of the NVT implementation expanded the type of feature that can be used for computing salience; for instance Lee *et al.* [98] uses aspect ratio and symmetry together with colour to generate feature maps. Since the work in [98] involves direction of attention towards human faces, they suggested aspect ratio of face and ellipse map formed from outer contour of faces as a good feature to pop-out faces in an image. This indicates that the type of feature map is mainly dependent on the application of an attentive system.

Visual object detection with a computational attention system (VOCUS) [91]

Frintrop [91] proposed an extension on NVT architecture that has an additional feature to include a top-down attention, known as *Visual Object detection with a Computational attention System* (VOCUS). VOCUS improves some of the structural design components and implementation details of NVT. VOCUS as in NVT uses a Gaussian pyramid to compute feature maps at different scale. It also employs *centre-surround* algorithm for computing intensity scale maps and across scale addition for generating feature map. However, the map reduction is done at higher scale than NVT which prevents the loss of information. Unlike NVT, VOCUS computes both on-centre and off-centre differences separately which guarantee a pop out of single white object in the middle of multiple black object on a grey background as shown in Figure 2.10. This also has a similar advantage on the top-down extension, making it possible to find either dark-on-bright or bright-on-dark object. Additionally, VOCUS implementation showed the use of CIE-LAB colour space instead of HSV colour space can improve the performance.

In fusing maps, VOCUS introduced the use of uniqueness weight before summation. Uniqueness weight determines the influence that a given map can have in the combined map. Maps with one strong peak are emphasized while those with multiple equivalent peaks are suppressed. An alternative weight determination technique called *Composite Saliency Indicator* (CSI) is proposed in [99]. CSI measure the contribution of each feature map to a salient region based on spatial compactness of saliency and salience density. Following weight normalization all feature maps of a given class are combined to give three conspicuous maps; each for: colour, intensity and orientation. These maps are weighted and normalized before a final summation to create salience map, which conclude the bottom-up attention of VOCUS. Frintrop also proved that as long as the difference between salience values is high enough, the focus of attention is not affected by translation, rotation or scale variations.

Figure 2.11 shows an extension of bottom up attention for visual search application. This extension includes two new maps: excitation and inhibition maps. These maps play a roll in creating a top-down salience map, which competes with the bottom-up attention salience in the formation of final map. An excitation map is weighted sum of all feature and conspicuity maps which are important constituent of the target feature, while inhibition map is formed from those maps which have weight smaller than one [91]. The global salience S

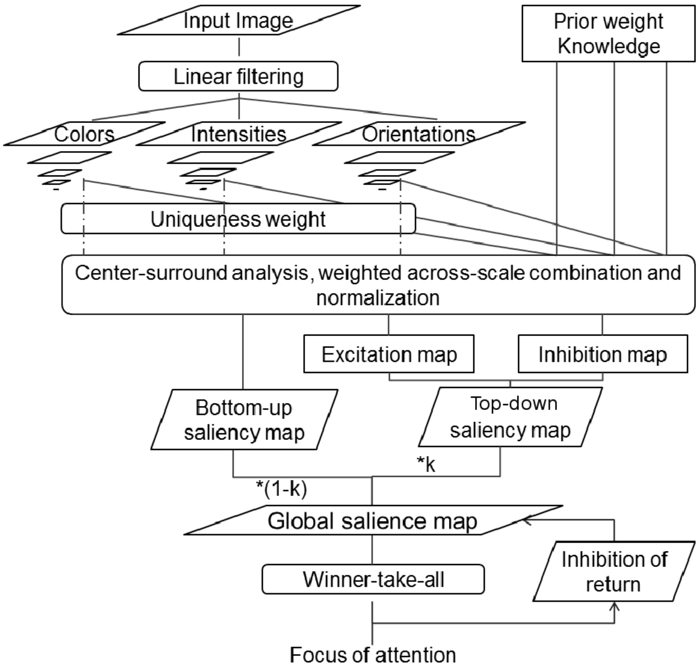


Figure 2.11: An extension of NVT based bottom-up attention architecture with a competitive top-down attention for visual search in VOCUS [91].

is computed as a weighted sum of top-down salience S_{td} and bottom-up salience S_{bu}. For a weight k determining the importance of these two maps the global salience is give by:

$$S = (1 - k) * S_{bu} + k * S_{td}. \tag{2.9}$$

Both NVT and VOCUS are used as a building blocks in most other attentive systems with a little modification on the implementation detail. Kootstra *et al.* [100] argue the use of local symmetry for salience map synthesis is an important part of human gaze control. Hence, they tested the benefit of such feature in artificial attention. As a result they proposed MUlti-scale Symmetry Transform (MUST) to detect symmetrical interest points and Symmetrical Region-of-Interest Detector (SymRoID) to detect symmetrical regions of interest. An other improvement over VOCUS is brought up by Rasolzadeh *et al.* [94] using Artificial Neural Network (ANN) for their dynamic bottom-up and top-down attentive grasping system. Their architecture, shown in Figure 2.12, uses an interactive spiking Neural Network (NN) to bias the bottom-up processing towards a task. In top-down attention context and task dependent, optimal set of weights are learned through NN.

Yu *et al.* [101, 102] proposed a combined bottom-up and top-down object based attention mechanism to introduce goal directed behaviour to control gaze based on current

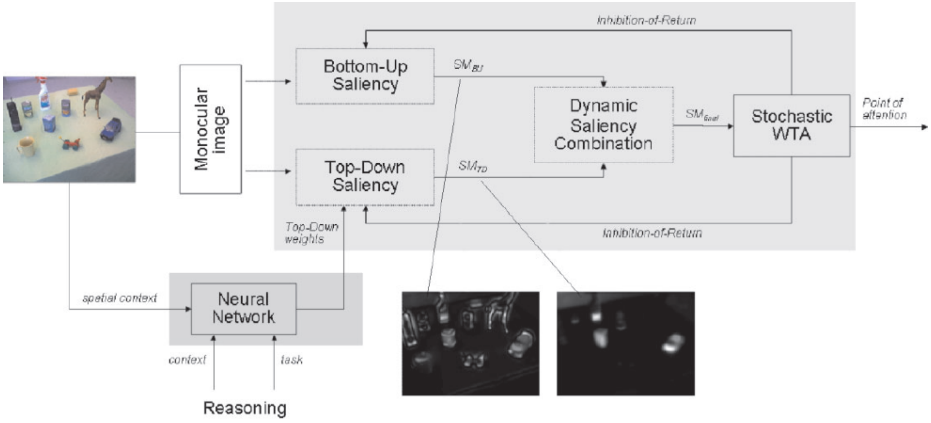


Figure 2.12: Rasolzadeh *et al.* model of combined bottom-up and top-down attention, with top-down task and context dependent weight [94].

task and learned knowledge. The bottom-up attention is based on NVT for automatic detection of unexpected or unusual situations while the top-down attention is done in three stages: pre-attentive segmentation, top-down attention selection, post-attentive perception. The use pre-attentive segmentation leads to proto-objects which are formed by clustering similar features as a possible components of objects or group of objects. In the top-down attentional selection stage, task relevant objects (objects that are useful to accomplish a given task) are determined based on long term memory (LTM) task representation which is used to deduce task-relevant features for building attentional template in the working memory (WM). Finally a probabilistic location-based salience map is estimated by comparing attentional template and pre-attentive feature. The third stage (post-attentive perception) involves detailed interpretation of attended object, in order to produce an appropriate action and learn LTM task representation for guiding subsequent attention. The top-down section of their architecture is shown in Figure 2.13.

A distinct behaviour of object-based saliency requires the use of segmentation or grouping mechanism to identify possible object structure [101, 102]. Though this might add computational burden for an online application, Yu *et al.* claim that it is more robust than space based saliency. Additionally, in applications that can use object template information for other robotic tasks, the computational burden can be ignored.

2.5.3 Focus of Attention for Mobile Robotics

Our interest here is to use focus of attention for the benefit of SLAM navigation. Probably a closely related work available in literature is the one by Frintop and Jenseflt [103]

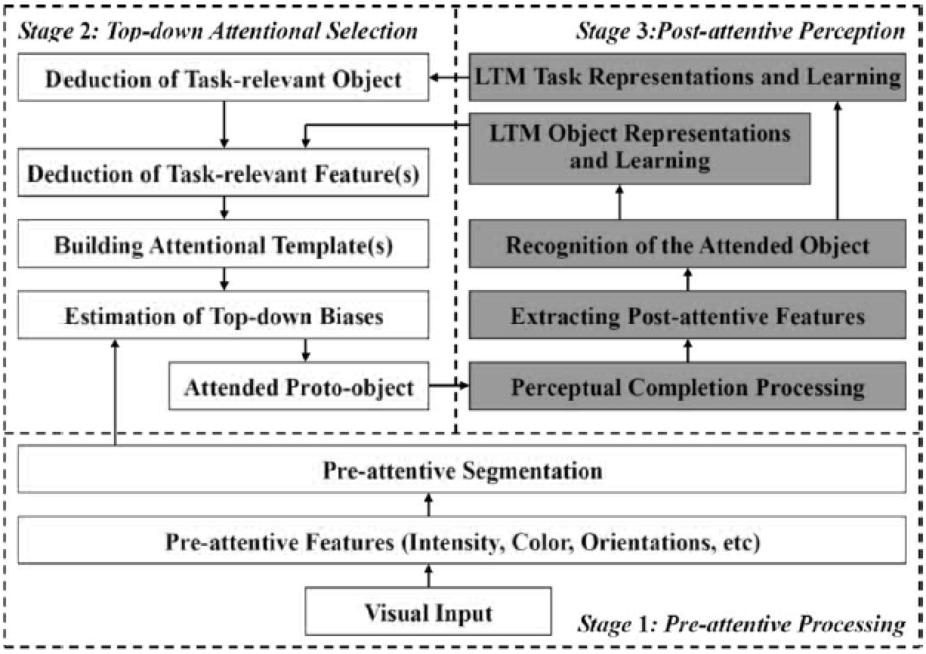

Stage 2: Top-down Attentional Selection
Stage 3:Post-attentive Perception
Deduction of Task-relevant Object
LTM Task Representations and Learning
LTM Object Representations and Learning
Deduction of Task-relevant Feature(s)
Building Attentional Template(s)
Recognition of the Attended Object
Estimation of Top-down Biases
Extracting Post-attentive Features
Attended Proto-object
Perceptual Completion Processing
Pre-attentive Segmentation
Pre-attentive Features (Intensity, Color, Orientations, etc)
Visual Input
Stage 1: Pre-attentive Processing

Figure 2.13: A goal-directed visual attention based on object-based salience [101].

which uses a visual attention scheme to assist their simultaneous localization and mapping algorithm. Figure 2.14 shows this architecture where VOCUS is used to select the most discriminative region of an image. Accordingly, the repeatability that can be obtained using attention-driven ROIs is much better than that from non-salient region or feature selected by conventional detectors like SIFT [50] and Harris corner detector [47]. Predicted landmarks from SLAM system initializes a top-down attention for either loop closure or exploration, while any surprising new landmark is used to dictate the bottom-up attention. One advantage of using VOCUS in this type of framework is its computational simplicity which does not add much on the overall SLAM performance.

Another application of attention for mobile robotics is shown in [104] which proposed attention-driven feature selection for obstacle avoidance and navigation. Einhorn *et al.* [104] were inspired by looking in at primates obstacle perception where animals usually turn their gaze toward areas of possible obstacle. In their architecture, see Figure 2.15, new features are selected in an image areas which are more relevant for obstacle detection and preventing collision. Rectangular regions of interest R are determined by maximizing a weighted sum of objective function o_i thought an image I_o as:

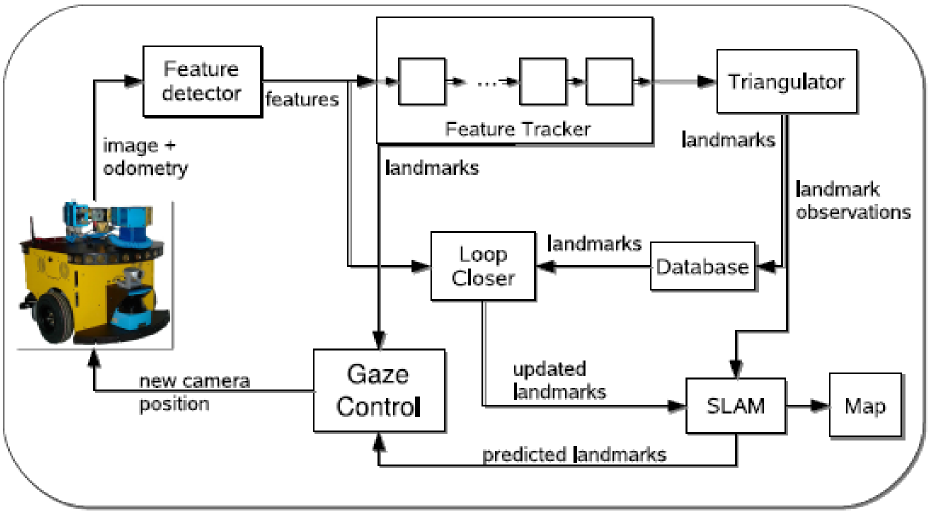


Figure 2.14: Attention driven active visual SLAM system. VOCUS [91] is used to decide on "where to gaze at?" (image source [103]).

$$R = \underset{R' \subset I_0}{\arg\max} \sum_{x' \in R'} a(x'), \tag{2.10}$$

$$a(x') = \sum_i (w_i o_i(x')), \tag{2.11}$$

where x' is pixel location, while $a(x')$ is attention value computed from objective function. These objective functions are similar to saliency maps; however, unlike a salience map their computation is based on obstacle uncertainty objective and an inhibitory objective function. The obstacle uncertainty objective is used to direct the focus in to areas where the presence of obstacle is unclear based on entropy of voxels in 3D grid map. An ambiguous voxel with an occupancy of 0.5 gets the maximum entropy while a free or occupied voxel gets the lowest entropy. While inhibition of return objective carries previous region of interests with a decaying coefficient.

A more recent work by Kim *et al.* might also require a mention here since it uses visual salience for underwater hull inspection [105]. However, their use of salience is slightly different from the goal of a typical computational visual salience map. The salience map is not used to direct attention rather it is used for measuring the registration utility of a key-frame in graph based SLAM. Accordingly, image registration refers to feature richness of an image which can be used to generate nodes in graph SLAM. Through the use of salience,

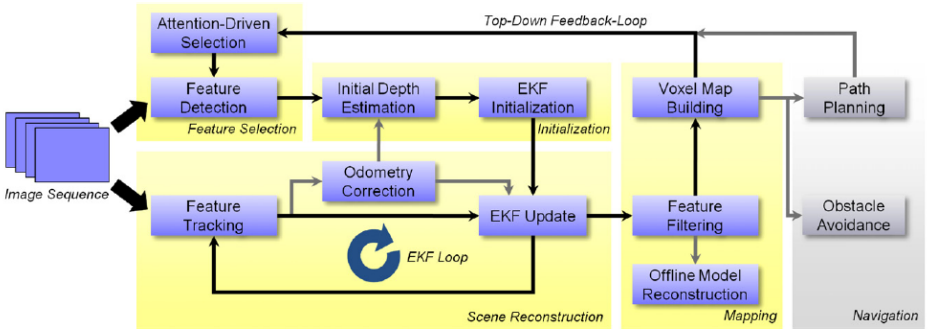


Figure 2.15: Architecture of attention-derived feature based obstacle avoidance and navigation (source [104]).

uninformative key-frames are removed to sparsify the pose-graph for computational efficiency. Additionally, as demonstrated in their other work [106], global salience value can be used to direct an AUV towards distinct region in search of loop-closure while exploring an area.

Sonar images are limited to intensity based features, making it impossible to gain most of the benefit that could be obtained using colour image in feature map computation. Additionally, the noisiness of sonar images can possibly make it difficult to synthesize any biologically inspired orientation maps. Hence, generation of sonar salience has to address these problems without introducing significant computational complexity. It also has to align properly with the line feature based sonar underwater SLAM algorithm that we are proposing in Chapter 4.

2.6 Summary

In this chapter we have provided a background information on commonly used underwater navigation sensors and their limitation included. This discussion included scientific sensors which make use of environmental landmarks for an absolute referencing. A robust underwater navigation system is developed by fusing sensor measurements with a kinematic or dynamic model of a vehicle involved. As we have discussed this can be achieved either using a deterministic model or stochastic state estimation technique where the later is preferred due to its probabilistic concepts for incorporating measurement noises.

SLAM is one type of stochastic state estimator which uses information coming form the surrounding environment on top of dead reckoning to generate a navigation map with in which the vehicle is localized. Among the various SLAM feature choices, we shared the success of line feature based techniques in underwater structured environments.

In the last sections of the chapter, we discussed the rising interest in using active vision for various robotic applications. This discussion later narrowed down on focus of attention which is a reactive process for selecting the most informative pieces of information. It uses a salience map to measure the importance of features in an environment.

Based on the background information as well as the literature review provided in this chapter, subsequent chapters will describe the various stages involved in devising a sonar salience based top-down over attentive SLAM navigation system. First lets start by a detailed discussion on sonar feature extraction in the next chapter.

Chapter 3

Sonar Feature Extraction

This chapter presents three types of line feature extraction techniques: Hough line extraction, split & merge and incremental line fitting. These techniques are mostly used for extracting lines from range measurements using laser scanners [63], however some of the works that we mentioned in Chapter 2.3.1 directly use these techniques on sonar images. However, such a direct usage of these techniques on sonar images does result in poor features, as partially demonstrated by Ribas *et al.* [57], which then hamper the success of feature matching procedure in SLAM algorithm. Here, further adaptation of these techniques for multibeam sonar will be discussed with the intention of improving the sonar feature quality taking the real-time requirement into consideration. The chapter also compare the slow Hough accumulator based uncertainty estimation technique, proposed in [57], with a traditional approach which assumes a constant deviation in each measurement direction.

3.1 Multibeam Sonar Characteristics and Simulation

Multibeam sonar is a type of sonar where multiple transducers are used to send out a series of sonar beams and receive any incoming reflection. Each acoustic beam propagates through the environment until it hits an object surface, then part of this signal is echoed back to multiple transducers. As in most time-of-flight systems, the time it takes to receive the signal is used to compute the range of the object based on the speed of sound in the water. The transducers wait until all possible echoes are heard before sending out the next acoustic signal which limits the overall speed of such sensor. These signals produce not just one but a series of periodic echoes each one is called sonar *bin*. An array of bins resulting from a single acoustic signal forms a sonar *beam*. Typically, a beam is fan shaped with a vertical beam width and a multiple of such beams are arranged horizontally to form a wedge that define the sonar field of view. For this work, we use two types of multibeam sonar sensors: BlueView P900-2250 dual frequency imaging sonar and MB2250 multibeam profiler (see Figure 3.1).

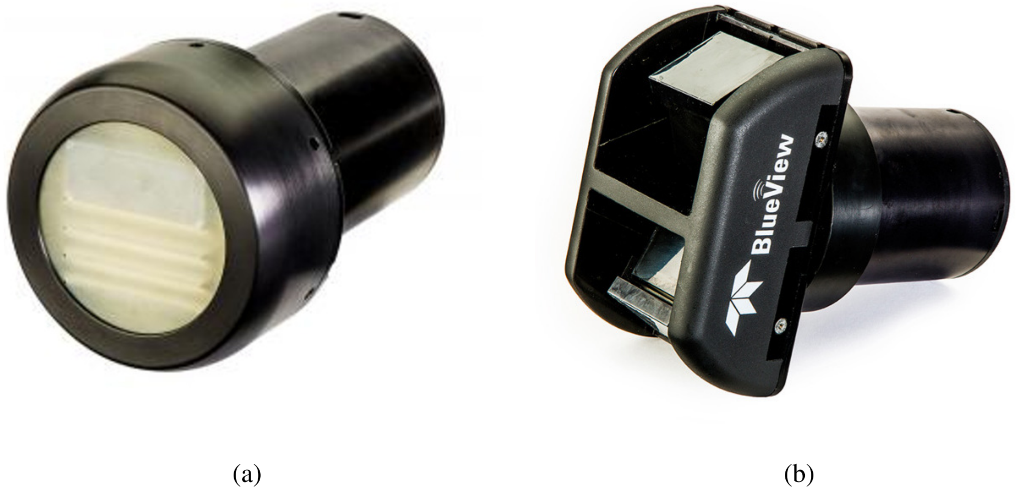
3.1.1 BlueView Multibeam Sonars

The P900-2250 is a dual frequency imaging sonar where a 900 KHz head provides a medium range capability while a short range head operating at 2.25 MHz provides a detailed view. The second sensor, MB2250, is a micro-bathymetry multibeam profiler which works only at 2.25 MHz with a short range response similar to a typical laser scanner. Detailed specifications of these two sensors is presented in Table 3.1. As shown in the table, there are few important differences between the two sensors. First the P900-2250 has a narrow field of view, which is limited to 45°, while MB2250 has 90°horizontal field of view. The P900-2250 has a much wider vertical beam width, 20°, making it slightly ambiguous to determine the vertical position of targets in an image. Another, yet less important, advantage of MB2250 over P900-2250 is the update rate, but due to the processing load on the driver typical sonar rates are way below 40 Hz. One key benefit of using P900-2550 is that it provides an option to switch between the long range head for coarse view and a short range head for detailed inspection.

Other than these differences, however, the two sonars work based on the same principle resulting in closely related images as show in Figure 3.2. Both images are taken in underwater tank where the sonar is facing the tank wall. However, image from MB2250 is more sharp and have a wider horizontal span due to its narrow beam width and wide field of view. The significance of these differences is studied in the next section while investigating the formation of sonar image for simulating these sensors.

3.1.2 Characteristics of Sonar Images

The performance of the proposed system in this document is required to be tested using a simulated environment, which involve the use of simulated sonar. This simulated sonar needs to have a very good resemblance with the actual sensor to achieve a smooth transition



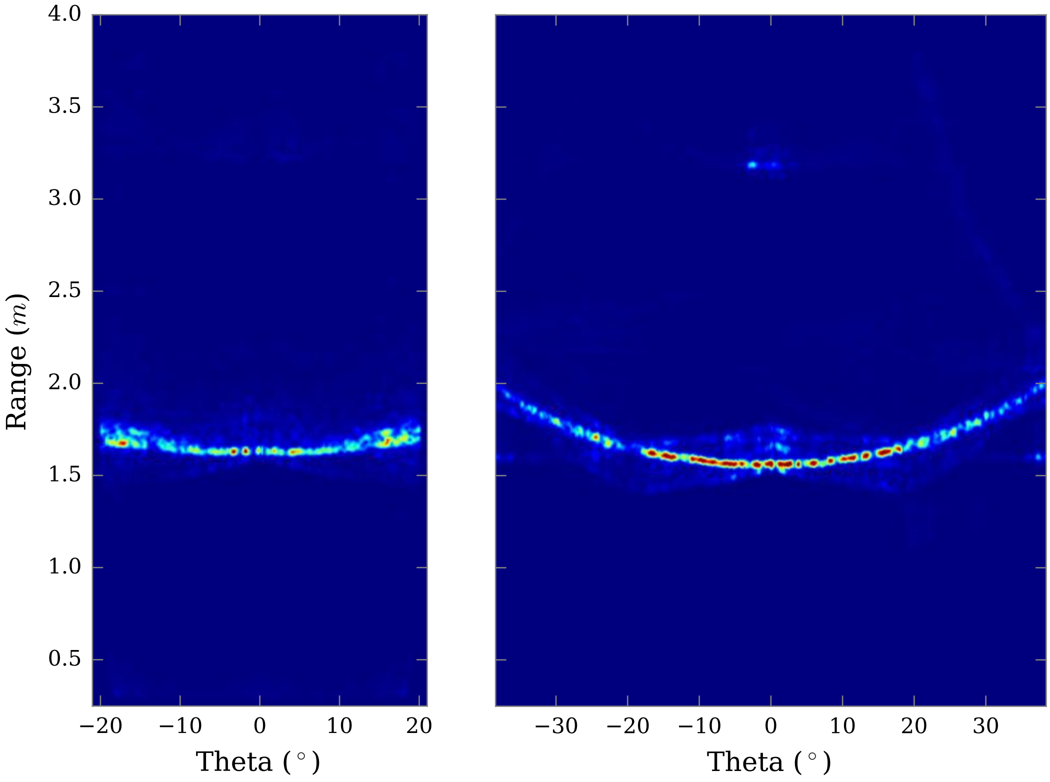
(a)

(b)

Figure 3.1: BlueView multibeam sonar sensors: (a) BlueView P900-2250 forward looking sonar, (b) BlueView MB2250 multibeam profiler.

Specifications	**P900-2250**		**MB2250**
	900KHz head	2.25MHz head	
Frequency	900 KHz	2.25 MHz	2.25 MHz
Field of view	45° × 20°	45° × 20°	90° × 1°
Max range	100 m	8 m	10 m
Beam width	1° × 20°	1° × 20°	1° × 1°
Number of beams	256	256	512
Beam spacing	0.18°	0.18°	0.18°
Range resolution	0.025 m	0.01 m	0.01 m
Max update rate	15 Hz	15 Hz	40 Hz

Table 3.1: BlueView Multibeam sonar specifications [107].



(a)

(b)

Figure 3.2: BlueView multibeam example polar images. (a) image from P900-2250 sensor (setting: field of view $42°$, maximum range $4\,\mathrm{m}$), (b) polar image from MB2250 (setting: field of view $77°$, maximum range $4\,\mathrm{m}$). Both images are taken in underwater tank where the sensor is facing a wall from $1.5\,\mathrm{m}$ range.

from simulation to an actual real world application. Here lets characterize the actual sonar images which in turn will make the simulation realistic.

The strength of beam reflection from a given ensonified surface depends on the surface's reflection coefficient which characterizes acoustic impedance difference between the water column and the surface. This is the reason why the visibility of underwater objects vary according to the material they are made of, which can be exploited for target detection. Furthermore echo strength is also affected by surface roughness as a function of acoustic wavelength. Hence the intensity of sonar reflection depends on the incidence angle of an acoustic signal where a high intensity is achieved when the signal has a normal incidence. For an ideally smooth surface a slight deviation from the normal orientation would result all the signal to be reflected away from the transducer; however, typical underwater structures have a significant surface roughness which makes them visible for wider range of incidence angles. An other factor that affects the acoustic backscatter is the volume of reverberation as a function of signal wavelength. Additionally, as in most time-of-flight cameras it is affected by the range of reflection due to signal attenuation; however, this is less of a problem while operating with in the optimal sensor range. Lastly, even though beams in multibeam sonar are supposed to be independent, it is important to note that in most practical situations they are actually correlated due to beam overlap.

In order to demonstrate the effect of these factors in sonar image formation a series of tests were executed. In each test either of the sonar (*i.e.* P900-2250 or MB2250) is set-up facing a wall at a right angle and a sequence of images are taken, the number varying between 10 and 50. These sequences of sonar frames are then averaged to remove salt and pepper noise and the processes results in a mean image. Then individual beams are characterized and modelled for simulation. One example of such an average image is shown in Figure 3.3. In this case MB2250 sonar is set-up to face the wall at normal angle and 10 subsequent images are taken from a single sensor position. The mean of these 10 sonar frames is shown in Figure 3.3 (a). This image shows an abundant amount of low intensity sensor noise throughout and particularly around the actual wall. Such type of noise can be removed using an initial preprocessing step which involve thresholding and low pass filtering. For this particular test, initially the image is smoothed using a Gaussian filter with a window size of 3, then intensity values below 0.01 are suppressed.

Another important feature to notice is the presence of features coming from a *ghost*

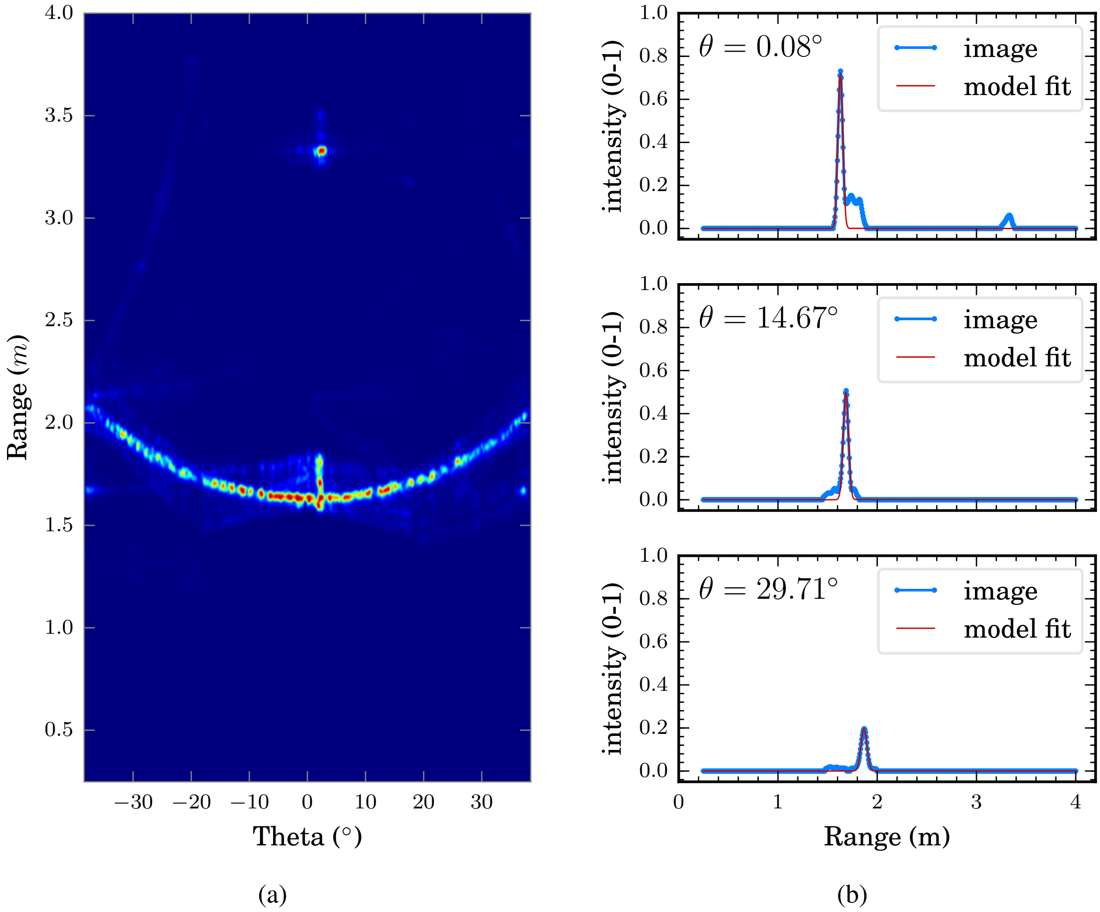


Figure 3.3: Average MB2250 sonar image and intensity profile in selected beams. In a) average MB2250 sonar image, with max-range 10 m and FoV 76.8°, and b) intensity profile of selected beams and Gaussian model fitting on the main peak.

reflection. A ghost reflection happen when operating in a confined environment where a sound wave reflects of multiple surfaces and then returns to the transducer giving a false impression regarding object of reflection. For instance, the bright sport at the top of the image is the sonar sensor itself seen after a multiple reflection between the front and back walls. To avoid confusions arising from such features only the first peaks of each beams are considered for feature extraction as well as sensor simulation.

Lastly, if we ignore the discontinuity of the wall intensity due to dark spots while traversing across the beams, there is a trend on the variation of the peak intensity and the corresponding peak width. The wall appears to be more bright at the centre of the image with a narrower peak width than at the edges. This can be seen more clearly by analysing the profile of selected beams shown in Figure 3.3 (b). The figure shows three smoothed beams with orientations angles 0.08°, 14.67° and 29.71°. As shown in the figure the height

Beam Angle (°)	**Intensity** (0-1)	**Mean** (m)	**STD** (cm)
0.08	0.72	1.62	2.4
14.67	0.50	1.69	2.57
29.71	0.20	1.88	2.94

Table 3.2: Model parameter for sonar beams shown in Figure 3.3b.

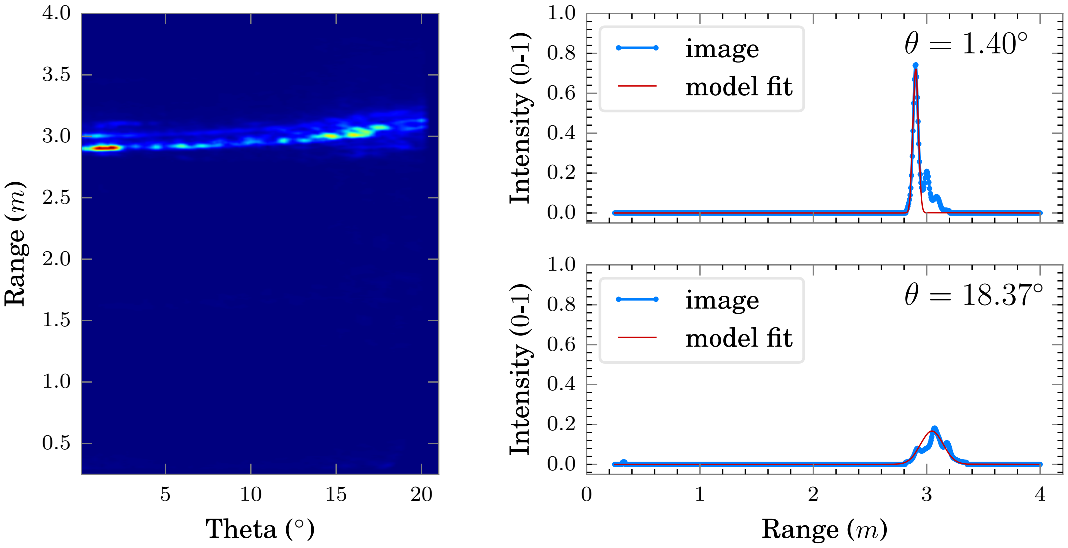
of the peak intensity is inversely proportionally to the beam orientation. Additionally, it can be seen that at $0.08°$ the sonar beam has two peaks where the second one is a ghost reflection and we will ignore it in our sonar simulation. However, the change in peak width with respect to beam orientation is difficult to notice from the figure. It will become apparent next when modelling each peak with a normal distribution.

In order to simulate each beam more realistically, bins within 10 pixel distance from the first peak are used to fit the actual sonar beam intensity profile with a normal distribution. The mathematical representation of this model for the j^{th} sonar beam intensity $I_j\left(\rho\right)$ is given by

$$I_j\left(\rho\right) = \mathcal{A}\exp\left(-\frac{\left(\rho - \rho_{peak}\right)^2}{\sigma^2}\right), \tag{3.1}$$

where ρ is range and ρ_{peak} range of maximum reflection for model parameters: scale $\mathcal{A}$ and standard deviation σ. For each beam the model parameters are computed by minimizing the sum-of-squared-error over the actual sonar peak profile. The resulting parameters for the three selected beams in Figure 3.3 are presented in Table 3.2. Accordingly, the model standard deviation increases with an increasing angle of beam orientation, which in this set-up is equivalent to sonar angle of incidence β, while model scale $\mathcal{A}$ is inversely proportional to beam orientation angle. A similar test on P900-2250 is shown in Figure 3.4, which shows only the right section of an average sonar image. The images are taken from $2.9\,\mathrm{m}$ range while the sonar is facing the wall at normal angle. As in MB2250 two representative beams are selected in the figure with beam orientation $1.4°$ and $18.37°$. The peak intensity is higher for beam with lower angle of incidence, while the peak is widely spread for beams with a higher angle of beam orientation.

Comparing Figure 3.3 and 3.4 we can see the peaks in P900-2250 are more spread than that of MB2250 this is mainly due to the difference in the vertical beam width. On the other hand, there are regular darker spots across the wall profile in both images, this is mainly due



(a)

(b)

Figure 3.4: Average P900-2250 image and selected beam intensity profile (only the right half of the sonar image is used). In a) average P900-2250 sonar image, with max-range 10 m and FoV $42.0°$, and b) intensity profile of selected beams and Gaussian model fitting on the main peak.

to the roughness of the wall surface which inevitably has small patches which refract the signal away from the transducer. This will be more clear in the next section, as the concept of sonar characterization is used to sensitize sonar beams that matches the average sonar image.

3.1.3 Simulation of Sonar Images

The simplest way to simulate a multibeam sonar is through the use of a *ray casting* technique. Ray casting is a range estimation technique where the range of object of reflection is obtained by determining the intersection distance of a ray with object's boundary. In case of multibeam sonar each beam is represented by individual ray. A similar set-up is used for simulating multibeam sonar sensor in UWSim simulator through a multiple range sensors arranged with a given beam spacing [4]. The raw output of a ray casting based sensor is a list of ranges in each beam corresponding to points of reflection which, if converted into an image, will have a single pixel peak corresponding to every response. However, without further processing this data is a bit unrealistic lacking most of the sonar characteristics mentioned in previous section. From the previous discussion, a real beam profile is more realistically represented using a normally distributed signal around each peaks.

A better sonar simulation requires a ray casting operation to be followed by a model fitting process where every peak is replaced by a Gaussian model using Equation 3.1. As discussed before the scale and standard deviation of each peaks in every beam are dependent on the angle of incidence β. Here other factors are not considered because either their impact is insignificant or their inclusion will make the simulation process unnecessarily complex. Now it is important to introduce a mathematical expression that relates the model parameters in each beam with their corresponding angle of incidence. In order to determine this expression, next we will further study the average images shown in Figure 3.3 (a) and Figure 3.4 (a).

Figure 3.5 show the relationship between the peak intensity and angle of incidence for both sonar sensors. Considering the left and right half of the average images, the incidence angle is symmetric and repetitive. The plots in the figure show the average peak intensity for every incidence angle. Ideally the sonar reflection is assumed to follow Lambert's cosine law where the ratio of transmitted sonar intensity to the received sonar intensity is given by the cosine of angle of incidence. However, in practical applications this doesn't hold mainly due to a significant contribution from a specular reflection in addition to isotropic diffusion that a Lambertian surface assumption requires. This is clearly seen in Figure 3.5 where a least square minimization of a Lambertian model is far from the actual measurement. It is rather more convincing to approximate the relationship using a linear model $\mathcal{A} = m_{\mathcal{A}}\beta + b_{\mathcal{A}}$. Furthermore, the use of a simple linear model will make the simulation faster. Both sensors are better approximated using a linear model with a negative slope yet the P900-2250 image has a steeper slope with a lower intensity value for a given incidence angle compared to image from MB2250 sonar.

Similarly, the relationship between peak standard deviation and angle of incidence is approximated using linear model $\sigma = m_\sigma\beta + b_\sigma$ as shown in Figure 3.6. In this case the slope is positive and higher for images taken by the P900-2250 sonar sensor.

As a result of this realization, an output of ray casting technique can be converted into a more realistic sonar image as shown in Figure 3.7. In Figure 3.7 (a), a polar image generated from a row range measurement obtained from a ray casting technique. The high peaks in each beam are one pixel wide and their position is determined by discretizing the range corresponding to each beam based on a given range resolution (0.62 cm/pixel is

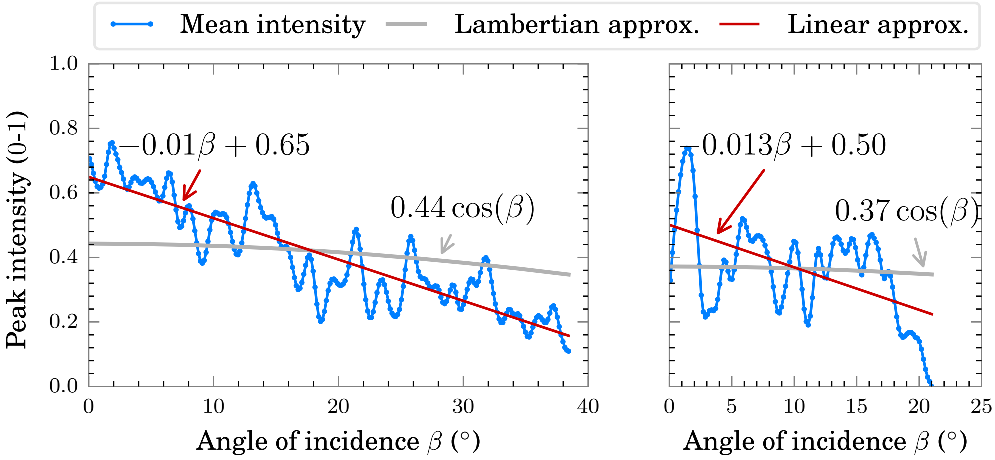


Figure 3.5: Relationship between angle of incidence and peak intensity in each beam. MB2250 (left) and P900-2250 (right) sonar image first peak intensity profile.

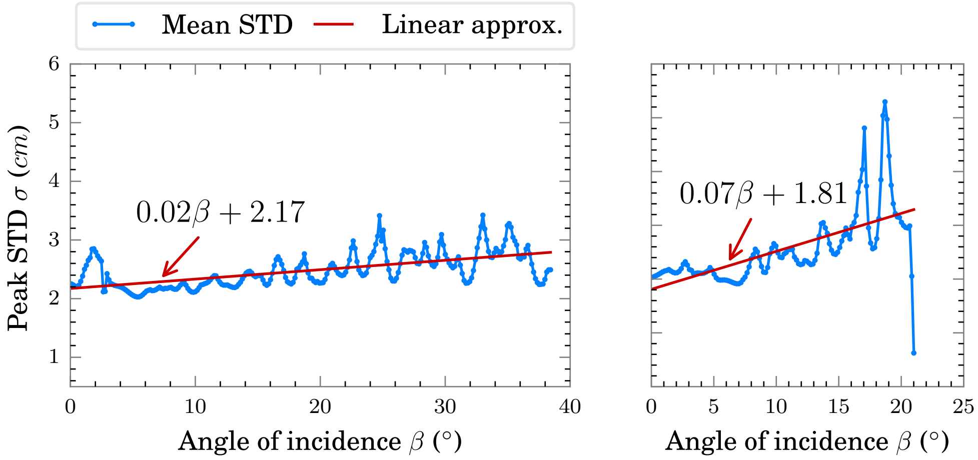


Figure 3.6: Relationship between angle of incidence and peak spread in each beam.

used for this example). The next stage of this simulation is controlled by four simulation parameters: peak intensity slope $m_{\mathcal{A}}$, peak intercept $b_{\mathcal{A}}$, peak standard deviation slope m_{σ}, and peak deviation intercept b_{σ}. For instance, the simulation of MB2250 sonar image in Figure 3.7 (b) is achieved using the following sets of parameters $[-0.01, 0.65, 0.02, 2.17]$ respectively. However changing the parameter values to $[-0.013, 0.5, 0.07, 1.81]$ produce a simulated image that resembles image from P900-2250 sonar (see Figure 3.7 (c)).

A ray casting based sonar simulation is simple and computationally efficient. Additionally the technique can be used in virtual environment where long missions are executed while testing a navigation system. However, the method does not provide any control over

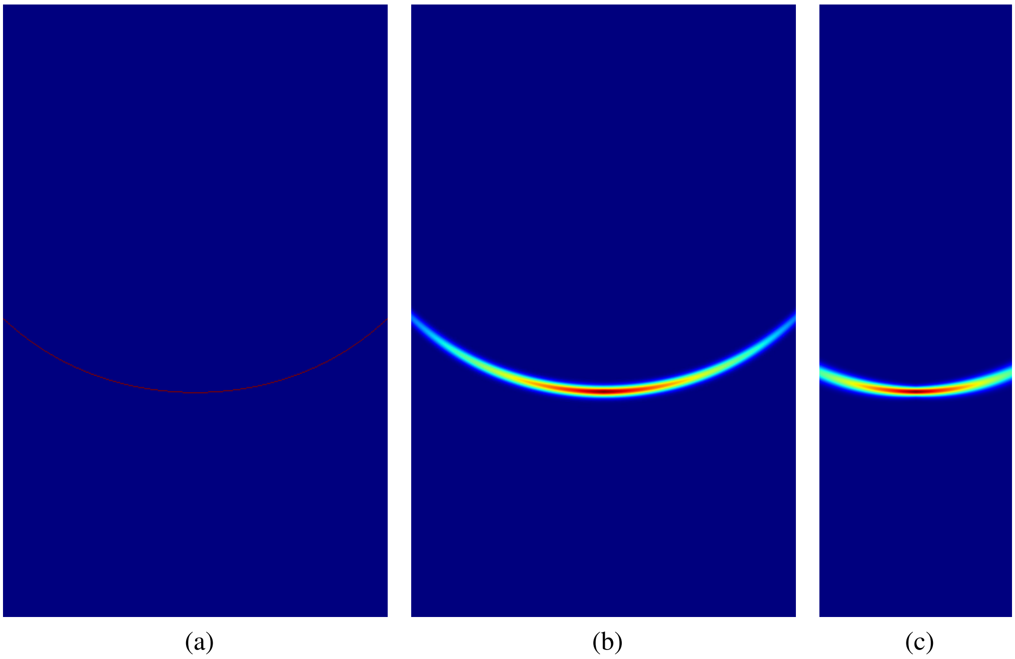


Figure 3.7: A ray casting based sonar image simulation. In (a) a row ray casting range measurement turned into a polar image, (b) simulation of MB2250 sonar, (c) simulation of P900-2250 sonar.

the uncertainty associated with lines which are later extracted from the simulated sonar image. The knowledge of the noise level is highly essential in order to compare line feature extraction algorithms. Hence an alternative Hough space based simulation technique is adopted from the work of Ribas *et al.* [57].

The process of turning a given line segment into a sonar image starts by creating a set of ρ-θ pairs based on the line parameters and a corresponding Gaussian noise covariance matrix. Then these lines, represented by their ρ-θ pairs, are projected over a polar image using a predefined sonar sensor parameters (for this illustration we used the parameters of the sonar image shown in Figure 3.3). Echo intensity of an image pixel is determined by the number of lines that gets projected over that pixel. This procedure is show in Figure 3.8 where a line with parameters $\rho = 1.5\,\text{m}$ and $\theta = 22.5°$ is projected over a sonar image. In Figure 3.8 (a) set of ρ-θ pairs are generated around the line based on a covariance matrix whose 3-σ limit is shown by the bounding ellipse. Some of the lines represented by these parameters are shown in Figure 3.8 (b).

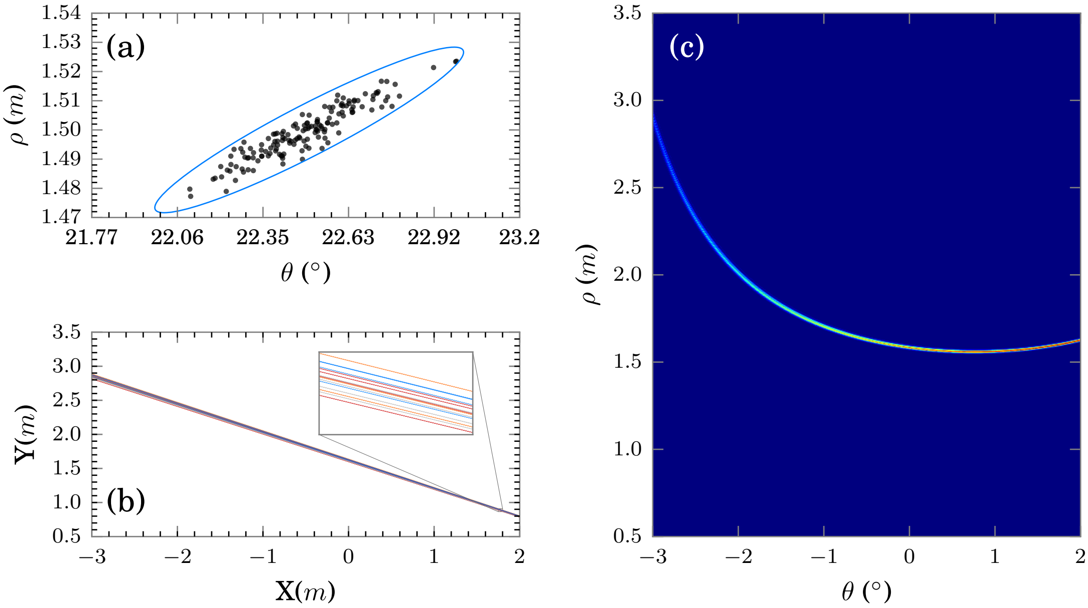


Figure 3.8: Sonar line segment simulation. (a) a set of ρ-θ pairs generated around the actual line based on the covariance matrix shown by the ellipse. (b) sample line whose parameters are displayed in (a). (c) polar image of a simulated line.

3.2 Sonar Line Feature Extraction

The aim of using a sonar sensor in this application is to capture a target's position information so that it can be considered as a landmark in order to enhance the accuracy of the AUV navigation system. As mentioned in Chapter 2, the most significant information in a sonar image is extracted as a feature. Additionally, the chapter also discussed sonar features in man made environments are better described using line segments. The next key questions that need to be addressed are: how to represent these line features? what is the best technique to extract features accurately with less computational cost? and how to determine the uncertainty of these line features? These three questions are discussed in the remainder of this section.

3.2.1 Line Features Representation

Line segments in a 2D plan can be represented in various ways, all with their pros and cons. For instance, a line segment, whose endpoints are at $P_0 = (x_0, y_0)$ and $P_1 = (x_1, y_1)$, can be defined using a slope intercept form as $y = m \cdot x + b$, where the parameters of the line segments are the slope m, the y-intercept b and either the x or y co-ordinates of the endpoints as a limit (*i.e.* $[x_0, x_1]$ or $[y_0, y_1]$ respectively). However, such type of line representation fails for a vertical line where the slope becomes infinity. In order to avoid such kind of singularity, a vectorial representation of a line segment can be used as $P\left(s\right) = P_0 + s \cdot \vec{u}$, where $\vec{u}$ is a line direction vector which is given by $\vec{u} = P_1 - P_0$, and s is a real number between 0 and 1. Alternatively, the four parameters of the two endpoints can be used to avoid the singularity in slope intercept form. However, both of these techniques lack feature compression capability because the line segment does not give any meaning in the absence of two of the four parameters. Additionally, an endpoint based representation will make it difficult to identifying colinear line which might need to be joined after a multiple view of a line. A much better option for our work is a *Hessian normal* form of a line segment, which is discussed next.

In a typical man-made structure, there is a redundancy of features mainly due to the symmetrical and parallel nature of the environment. As a result of this colinear lines are a common occurrence. This can happen either from an interrupted view of a continuous wall or viewing a wall with opening or due to two separate structures put over the same line. To use the feature compression effectively, such types of line segments can be represented using a single pair of line parameters and an additional pairs of parameters to indicate the starting and ending point of each individual line segment. Hessian normal form of a line segment can provide this; the first two parameters are used to identify the line vector while the last two indicate the position of the two end points on this line. Given the two endpoints of the line segment p_0 and p_1, Hessian normal form of a line segment is given by: its normal distance from the origin ρ, the orientation of the line normal θ, the length of the line segment l, and the distance from the centre of the line segment to a point on the line where the line intersect its normal passing through the origin d as shown in Figure 3.9. The figure also shows equations for computing the four parameters given the two endpoints of the line segment. Both the measurement and its uncertainties are computed per line rather than every line segment, while the remaining two parameters are carried aside while doing SLAM (see Chapter 4). The covariance matrix of a line is given by,

$$R = \begin{pmatrix} \sigma_\rho^2 & \sigma_{\rho\theta} \\ \sigma_{\rho\theta} & \sigma_\theta^2 \end{pmatrix}, \tag{3.2}$$

where σ_ρ^2 and σ_θ^2 are the covariance in ρ and θ respectively, while $\sigma_{\rho\theta}$ is their cross covariance.

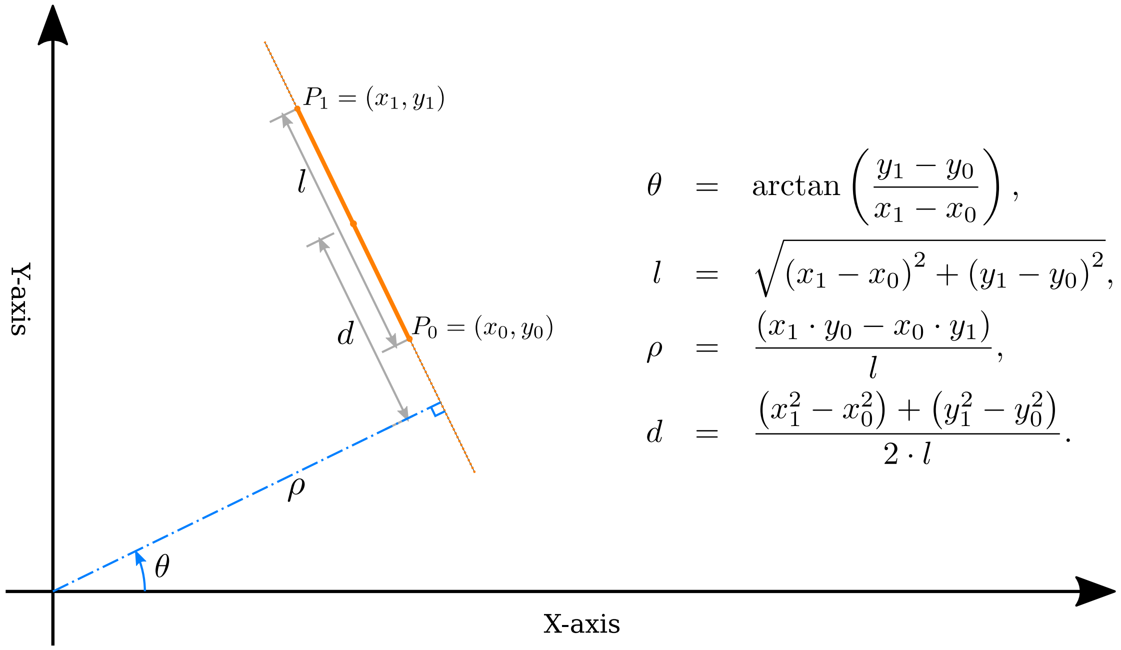

$P_1 = (x_1, y_1)$
l
d
$P_0 = (x_0, y_0)$
ρ
θ
Y-axis
X-axis
$\theta = \arctan\left(\frac{y_1 - y_0}{x_1 - x_0}\right),$
$l = \sqrt{(x_1 - x_0)^2 + (y_1 - y_0)^2},$
$\rho = \frac{(x_1 \cdot y_0 - x_0 \cdot y_1)}{l},$
$d = \frac{(x_1^2 - x_0^2) + (y_1^2 - y_0^2)}{2 \cdot l}.$

Figure 3.9: Hessian-normal form of a 2D line segment. The four parameters can be computed from the endpoints of the line segment.

3.2.2 Sonar Image Processing

Both MB2250 and P900-2250 are shipped with a common SDK (Software Development Kit). For this work, this SDK is warped into a ROS device driver that publish sonar images with their corresponding parameters. This list of parameters include: image dimension, minimum and maximum range, and Field of View (FoV) limits. Though the driver can provide an image in Cartesian (X-Y) co-ordinate, polar (ρ-θ) images are preferred in order to avoid the computational burden of a full image co-ordinate transform, then an X-Y points can be generated using a lookup table whenever needed.

Up on receiving a sonar image, the feature extraction node first applies a Gaussian filter over the image to remove salt and pepper noises. After this an empirically defined threshold is used to reset those pixels with a lower intensity value which are not associated with any physical feature in the environment. Then from each beam all the ghost reflections are removed based on their distance from the first peak in that beam. The resulting set of ρ-θ points are processed using a line feature extraction algorithm in order to identify the line segments in the image with their corresponding covariance relative to the sonar frame.

Most line extraction algorithms, including those techniques discussed in [63], are developed for 2D laser scan data. Even though these techniques can directly be applied for sonar data, this will not provide the optimal functionality. Mainly because none of these techniques make use of the intensity value which might improve the accuracy of the line extraction process. Hence our proposed line extraction algorithms use both the location and intensity of peaks in each beam. However, choosing the best line extraction algorithm is puzzling and this choice varies according to the required accuracy and computational speed. Therefore, in the next sections we will discuses three well-known line extraction algorithms and their adaptation for sonar data.

3.2.3 Hough-Transform Line Extraction

Hough-transform is a mapping of pixels from an image plane to a curve in a Hough-space. The Hough-space is an accumulator grid where each cell corresponds to a discrete ρ-θ pair. The resulting curve defines the set of all possible line parameter pairs representing line segments passing through the point in the image. The standard Hough-transform requires the use of X-Y points, however, using the polar points directly from the sonar image can avoid unnecessary computational burden. Hence the proposed technique directly uses the polar image for filling the Hough accumulator.

Given a pre-processed sonar point $P = (\rho_i, \theta_i)$, there are possibly infinite number of lines that can pass through this point (see Figure 3.10). However, assuming a discrete line orientation and a maximum angle of sonar beam incidence to be $\beta_{\max}$, the parameters of possible lines can be determined. Based on the second condition the orientation θ_{ij} of a line through P, with incidence angle β_j, should satisfy the following criteria,

$$\theta_i - \beta_{\max} \leq \theta_{ij} \leq \theta_i + \beta_{\max}. \tag{3.3}$$

The parameters of this line are computed as,

$$\theta_{ij} = \theta_i + \beta_j, \tag{3.4}$$

$$\rho_{ij} = \rho_i \cdot \cos\left(\theta_i - \theta_{ij}\right). \tag{3.5}$$

The result from Equation 3.4 and 3.5 are binned to the nearest Hough-accumulator cells. Then these cells are incremented: normally by constant value which is one; however, in our implementation the intensity of pixel P is used instead. The effect of this change is discussed later in Section 3.4. Figure 3.10 shows the procedure of mapping a single sonar

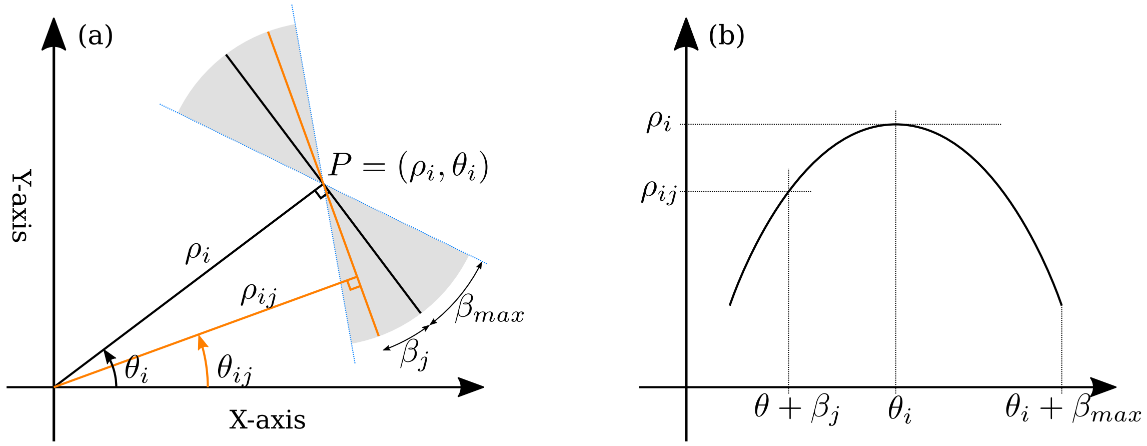


Figure 3.10: Hough-transform of a polar point in to Hough-space. In (a) a point P is shown together with a shaded region representing all possible lines that passes through the point, these lines have an angle of incidence below β_{max}. (b) an example line with parameters (ρ_{ij}, θ_{ij}) is mapped into the Hough-space.

image pixel into the parameter space. This procedure is repeated for every pixel which passed the pre-processing stage discussed in Section 3.2.2. A cell with a sufficient amount of vote is considered as a winner and the cell parameter values are considered as the mean ρ-θ values of the line. Before searching for the next line, every vote that correspond to the first line is masked in order to avoid detection of the same line multiple times. This is done by resetting cells neighbour to the winning cell. Then a similar procedure is followed until all the lines with a significant vote are extracted or a maximum number of line features are obtained.

The first step of Hough transform does only provide the two basic line parameters of each line segments. Then for each line segment its length and mid point parameter d are determined using the input polar point distribution. This procedure is started by classifying each point to the nearest line. Then in each set points which are far from the line represented by the peak are removed while the remaining points are projected over the line. Next this points are sorted and traversed from one end to the other looking for a gap between point distributions. When a gap arises a line segment is returned with all the four parameters if it satisfy minimum segment length criteria. The process continues with the remaining set of points.

3.2.4 Incremental Line Fitting

Incremental line fitting technique is one of the fastest line extraction algorithm for ordered set of points, as demonstrated in [108]. Typically such a technique starts from two adjacent points and keeps on adding one or more adjacent points at a time based on a distance criteria. According to this criteria, a new point has to be close to a point already in the line set and the distance from the point the line has to be below a maximum threshold d_l. This is demonstrated in Algorithm 3.1. In our implementation the presence of noisy point is considered. Hence line 10 is modified with a counter that tells how many times the condition in line 7 fails, if it fails above certain number of times then it is executed otherwise the new point is skipped and the process continue from line 4.

Algorithm 3.1 Incremental line fitting algorithm.

```
 1: start with point sets $S$
 2: while $S$ not empty do
 3:     initialize a line $l$ with 2 points from $S$
 4:     for a new point $p_j$ form $S$ do
 5:         compute distance $d_l$ to $l$
 6:         compute distance $d_p$ to closest point in $l$
 7:         if $d_l$ and $d_p$ satsfiy minimum condition then
 8:             go to $4$
 9:         else
10:             go to $3$
11: return all the line segments
```

Incremental line fitting is probably the simplest line extraction algorithm available, however adding a point a time is time consuming. As an alternative multiple points can be added at a time, making the implementation slightly complex and forcing a back and forth execution whenever the new set of point fails to match with a given line.

3.2.5 Split and Merge Line Fitting

By far the most widely used line feature extraction technique is called *split and merge* line fitting. Split and merge is also shown to be the fastest algorithm for 2D laser scan data fitting [108]. There are various implementations of this technique, the first being an *iterative-end-point-fit*, which uses the first and the last points of a set to fit a line. Most of the other split and merge implementations make use of the whole points in a set to fit a line which makes them significantly slower. Additionally, unless an extra validation step is incorporated, in these case the splitting point can end up to be at either end of the point set. One down side, iterative-end-point-fit depends only on the two end points of a set making the final line segment less optimal in terms of minimizing error.

The split and merge technique assumes an ordered point distribution for efficient implementation; hence, in our implementation an initial line segment identification procedure uses only the peak intensity points of each beam from the pre-processed image. Then the remaining points are classified in to one of these line segments. In order to increase the accuracy of the fit, the final line parameters are determined using a weighted least square technique. This process is shown in Algorithm 3.2. The advantage of using all the points in an image together with their intensity weight will be further discussed in Section 3.4.

3.2.6 Least Square Line Fitting

Although the initial split and merge line fitting operation can be done using end-point-iterative technique, the final refinement or parameter fitting for incremental line extraction technique require the use of least square approximation. The least square technique minimizes the sum of square difference between each point and the nearest point on the line. For a set of points $[x_i, y_i]$ and their corresponding intensity weight w_i, the line parameters $[\rho, \theta]$ are computed by minimizing,

$$\min \sum r_i^2 = \min \sum \left(\rho - \left(x_i \cos(\theta) + y_i \sin(\theta)\right)\right)^2 \cdot w_i. \tag{3.6}$$

The minimization of Equation 3.6 can be done either using linear algebra or closed form statistical expression. The later is recommended whenever the number of points involved is higher. As it will be shown in Section 3.4, the line parameters obtained using least square minimization are much closer to a point distribution than that of end point fitting with significantly higher computational cost. Hence, this work introduces a two stage split and merge technique which uses iterative end-point fitting for initial rough estimate which is then refined using a weighted least square.

Algorithm 3.2 Split and merge line fitting algorithm.

```
 1: start with all pre-processed point set $S_{all} = \{P_i | \, 0 \leq \; i < N\}$
        and corresponding intensity $W = \{I\left(P_i\right)\}$
 2: Generate list of peak intensity points
        $S_{peak} = \{P_j | \, 0 \leq \; j < M, P_j \in \; S_{all}, I\left(P_j\right) \Rightarrow$ max. intesnity of the $j^{th}$ beam$\}$
        assuming $M$ number of beams
                          split based on continuity
 3: $d_p = \{\| \, P_{j+1} - P_j \, \| \; where \;\; 0 \leq \; j < M - 1\}$        ▷ distance between adjacent points
 4: Smooth the list of distance $d_p$                                  ▷ using a moving average filter
 5: Split $S_{peak}$ into a set of $\{S_k\}$ for every $d_p > threshold$       ▷ split based on continuity
                          Iterative-end-point-fit
 6: for $S_k$ in $S_{peak}$ do
 7:     Determine the line $L_k$ using iterative-end-point-fit          ▷ use first and last points
 8:     Compute $d_l$ distance between points in $S_k$ and $L_k$
 9:     Detect a point with a maximum $d_l$
10:     if $\max \; d_l > \; threshold$ then
11:         Split the segment into $S_{k1}$, and $S_{k2}$
12:         append $S_{k1}$, and $S_{k2}$ to $S_{peak} = \{S_k\}$
13:     go to $6$
14: When all sets in $S_{peak}$ are checked return $\{L_k\}$
                          Classification of points
15: For all $P_i \in \; S_{all}$ compute the distance to the lines in $\{L_k\}$
16: Cluster the points in $S_{all}$ as well as their intensity $W$ to the nearest line
17: return $\{S_i\}$, $\{L_i\}$, $\{W_i\}$
                          Refine line parameters
18: for $S_i$ in $\{S_i\}$ do
19:     $L_i \leftarrow weighted\_least\_square\left(S_i, W_i\right)$
20:     complete the line segment parameter by calculating $[l, d]$ parameter
                          Merging and covariance computation
21: Merge colinear lines
22: Compute the covariance of each line segment
23: return all the line segments parameters and thier covariance
```

3.3 Uncertainty Estimation

Most line feature extraction implementations assume constant radial and angular deviations for representing feature uncertainty. This makes the feature extraction process easier with less computation burden. On the other hand, as shown in [57] a Hough accumulator can be used to determine the uncertainty of lines, in addition to their parameters.

Hough-line Uncertainty

Depending on the voting-space parameter, the vote for a given line is distributed across multiple cells around a peak. Identifying these cells can help to estimate the confidence level of a given line. As discussed in Section 3.1.3 (specifically in Figure 3.8), once the ρ-θ distribution is found, it is even possible to recover the sonar image. Hence, the first step in computing the line uncertainty is the search for these neighbouring line parameters. For this search, the winning line is considered as seed for a 4-connected neighbourhood region growing. One possible criteria for region growing is the use of overlap ratio as proposed by Ribas *et al.* in [57]. The overlap ratio is a measure of similarity between the sonar image and a line parameter. For a line (ρ_i, θ_i), the overlap ratio is computed first by projecting the line over a pre-processed (segmented) sonar image for all possible sonar beam angles. Then the overlap ratio η_i is given by the number of projections that overlap with a non-zero echo intensity divided by the total number of projections which is equivalent to the sonar image width. Next this ratio is compared to that of the winning line, if it is above a threshold then the line parameter is added to the line neighbour set and the process continue until all possible neighbours are visited. Finally the line uncertainty is given by the covariance of this neighbouring line parameter sets (ρ_i, θ_i) as,

$$R = s \cdot \begin{pmatrix} \mathrm{E}\left[(\rho_i - \mathrm{E}(\rho_i))(\rho_i - \mathrm{E}(\rho_i))\right] & \mathrm{E}\left[(\rho_i - \mathrm{E}(\rho_i))(\theta_i - \mathrm{E}(\theta_i))\right] \\ \mathrm{E}\left[(\rho_i - \mathrm{E}(\rho_i))(\theta_i - \mathrm{E}(\theta_i))\right] & \mathrm{E}\left[(\theta_i - \mathrm{E}(\theta_i))(\theta_i - \mathrm{E}(\theta_i))\right] \end{pmatrix} \tag{3.7}$$

where s is a scale factor to account for the change due to a pre-processing threshold value.

Hough based line extraction does not require an ordered point distribution which is one of the main advantage. However this is less significant for sonar images where points are ordered according to their beam orientation anyway. Additionally, filling an accumulator and searching for a peak is a time consuming process. This gets even worse when considering the uncertainty estimation, yet we will examine the performance of this system since it is the only non-empirical and unsupervised way of computing line uncertainty to date.

3.4 Comparison of Line Extraction Algorithms

The extraction of good features is a multiple stage process that requires the use of a good sensor together with a robust feature extraction technique. However, there are certain constraints, such as time and computational power, that restricts the selection of some techniques. In a more relaxed application, for instance in our work of evaluating a sonar servo control system we found out that a Random Sample Consensus (RANSAC) algorithm does provide a more accurate and stable wall line estimation [109]. However, in this application it is known before hand that the number of line segment involved is just one making the whole process computationally less expensive. Such assumption does not fit a typical underwater navigation scenario where the number of line segments can be significantly higher. Hence computational cost becomes an important criteria together with the accuracy of the line extraction. In this section we compare the performance of the three line extraction algorithms discussed so far.

3.4.1 Experiment Set-up

Performance comparison of the three line extraction algorithms is done using simulated environments show in Figure 3.11. Each test arena is filled with lines representing either walls or edges of columns with some extra random lines representing abandoned structures or floating objects. All the arenas are confined with in $10\,m \times 10\,m$ area through which a robot navigates to collect sonar measurements. The solid coloured lines in the maps indicate walls while the black dotted lines are robot trajectories from where the sonar measurements are taken. This robot paths are generated manually by tracing a mouse movement over each map. The test environment is used to validate two things: the first is the accuracy and performance of line extraction techniques while the second test compares the performance of covariance estimation.

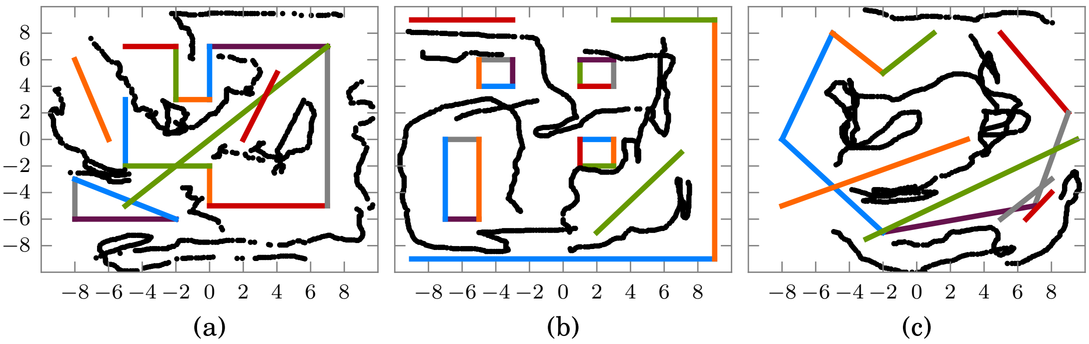


Figure 3.11: Simulated maps for line extraction comparison.

Setting up the sonar for Test-1

In order to simulate the sonar at each position, first the global map is transformed into a local version. Then, using a ray casting technique, a trace of the closest reflection points are generated through the full FoV of the sonar. Each ray intersection point is assigned an incidence angle which is the angle between the ray and normal of the reflection line. Rays represent sonar beams and intersection points are assumed to be peaks of a particular beam. Hence using the incidence angle and range of the intersection point a sonar profile of every beam is generated using the method described in Section 3.1.3. On each map on average 1000 robot positions are considered from which a total of 5340 line observation are made. A line is said to be observed if there exists an overlap between the line segment and the sonar. However, not all these observed lines are supposed to be detected, since some of the line segments are very short to pass the minimum line length requirement.

Setting up the sonar for Test-2

This set-up is intended to compare the two uncertainty assignment methods discuses in Section 3.3, where one is assigning a constant deviation in ρ and θ, while the second option is to use Hough accumulator based uncertainty. In this case, before detecting sonar intersection points all the visible lines are replaced by a hundred other lines which are generated using an assumed covariance and the initial line using random normal 2D distribution. Then an image is generated using all the intersection points between multiple lines and sonar rays. To allow the change in sonar uncertainty from one image to another, the assumed uncertainty is randomly chosen between extreme bounds. Accordingly the deviations in ρ and θ are chosen between $\sigma_{\rho\rho}$=[2 cm, 10 cm] and $\sigma_{\theta\theta} = [0.3°, 1.5°]$. The Hough accumulator similarity measure is set at 0.52. This value is determined by manually matching a single line uncertainty with a training example line. A higher value can result with empty neighbouring cell sets, hence it is better not to narrow the neighbourhood size too much though this might improve the computation efficiency.

Line extraction parameter assignment

Choosing the right parameter for comparing the three line extraction techniques discussed is not straight forward. As much as possible similar parameters are used whenever the three techniques share a parameter. For instance, the minimum length of a line segment d_l is set to be $10\,\mathrm{cm}$, where the minimum number of points in a segment N_p is 10 and the maximum distance between to adjacent points within a segment d_p is $10\,\mathrm{cm}$. The Hough accumulator space is set up so that each cell is spaced $15\,\mathrm{cm}$ by $2°$ and the minimum distance allowed between two Hough peaks is $1\,\mathrm{m}$ and $20°$. For each line extraction technique, three categories are considered based on the type of sonar points used for line extraction:

a) In this type of experiment only one pixel per beam is used to extract the line, the pixel corresponds to the first maximum intensity above a threshold in a given beam. Hence for a simulated MB2250 sonar image only 512 pixels are used for line extraction.

b) In this case all the points around a maximum peak with intensity above a threshold are used for incremental line fitting and Hough line extraction while in case of our split and merge implementation the initial line estimate is always determined using the high intensity pixels however for this category the final refinement will use all the points.

c) In this final category all lines are used together with their corresponding intensity weight, *i.e.*Hough accumulation is incremented by the actual intensity instead of a constant, while standard least square refinements are replaced by weighted least square approximation as given by 3.6.

The number of points and the values of votes in the Hough accumulator will vary according to the category. Hence the threshold for maximum peaks are 80, 1900 and 600 for category a, b, and c respectively.

3.4.2 Result and Discussion

Result for Test-1

Quantitative comparison of the three line extraction methods and the three different category of point selection is show in Table 3.3. One of the criteria for the best line fit is the distance of the line $[\rho_{i,}\theta_i]$ from a matched ground truth line $[\rho_{Ti}, \theta_{Ti}]$. This distance is computed as a normalized Euclidean distance d_h (as proposed by Lakaemper in [110]), which is given by,

$$d_h = \sqrt{(\rho_i - \rho_{Ti})^2 + \left(\frac{\min\left(|\theta_i - \theta_{Ti}|, |\theta_i - \theta_{Ti}| \pm 360^\circ\right)}{h}\right)^2}. \tag{3.8}$$

Where the normalization factor h is assumed so that a 25 degree difference in angle corresponds to $1\,\mathrm{m}$ distance. As a quality measure, a good technique is supposed to extract a maximum number of true lines with a minimum number of segments, *i.e.* extracting all possible lines intact. The accuracy of the line extraction is measured using Root Mean Square Error (RMSE) which, for N number of distance d_i, is given by,

$$RMSE = \sqrt{\frac{\sum_{i=0}^{n} (d_i)^2}{n}}, \tag{3.9}$$

where d_i corresponds to either d_h or error in range $\Delta\rho = \rho_i - \rho_{Ti}$ or error in angle $\Delta\theta = \theta_i - \theta_{Ti}$ as shown in the table. The precision is measured with the corresponding standard deviation (STD).

The computational comparison is done on a laptop with processor Intel® Core™ i5-3360M CPU @ 2.80GHz × 4 and 6GB RAM. *Split-and-Merge* line fitting is unarguably the fastest at $1180\,\mathrm{Hz}$, which runs slightly slower when all the points are used rather than peaks of each sonar beam. On the other hand, *Incremental* line fitting ended up being the slowest. This is due to python's slow looping where the technique has to loop over every point. Additionally, *Incremental* line fitting uses least square approximation to update the line parameter after each new point addition unlike that of our *Split and Merge* implementation which uses end point based fitting until the final refinement. In case of *Split and Merge* technique lines are divided to a fewer number of small segments compared to the output from *Incremental* line-fitting, hence it is considered as a better detector for complete lines.

The accuracy and precision of *Incremental* line fitting is much better than the other two

Method & Category		Speed Hz	No. of segm.	No. of lines	Matched lines (%)	RMSE (cm/deg)			STD (cm/deg)		
						d_h	$\Delta\rho$	$\Delta\theta$	d_h	$\Delta\rho$	$\Delta\theta$
HL	a	106	10265	4189	77.58	17.77	3.50	9.46	10.91	9.14	3.50
	b	12	6052	4335	80.89	16.76	3.11	10.33	11.22	9.95	3.11
	c	11	8039	4405	82.13	16.70	3.12	10.26	11.08	9.74	3.12
INC	a	11	6191	4822	86.15	10.65	1.82	6.70	7.74	6.61	1.82
	b	11	6207	4792	84.92	10.20	1.92	8.48	6.65	6.61	1.91
	c	11	6207	4791	84.98	10.07	1.90	8.61	6.55	6.47	1.89
SM	a	1180	5466	4813	86.07	9.95	1.66	5.77	7.39	6.09	1.66
	b	481	5465	4843	87.25	7.97	1.54	6.48	5.01	4.98	1.54
	c	475	5465	4841	87.18	7.77	1.55	6.52	4.63	4.58	1.54

a, b and c are categories in terms of the points used as discussed in Section 3.4.1
HL-Hough line fitting, INC-Incremental line fitting, SM-Split and merge line fitting

Table 3.3: Results of line extraction validation experiment.

and it improves with the use off all the sonar points above a threshold. In actual sonar implementation the rate of sonar acquisition is under $30\,\text{Hz}$, hence an extraction as high as $400\,\text{Hz}$ does not constrain the overall performance of the navigation system. However, it is important to note that the performance of all the three methods increase with the use of sonar intensity weight without causing too much degradation on the computational speed.

Result for Test-2

For comparison of techniques for assigning line uncertainty to measurement two metrics are used. The first is the Frobenius norm where the sum of the square difference of an estimated covariance P_i and the corresponding ground truth covariance P_{Ti}. As in Test-1 the discrepancy between angle and range is normalized using a factor h. Hence the Frobenius distance d_F between P_i and P_{Ti} is given by,

$$d_F = \sqrt{\left(\left(P_d\left(0,0\right)\right)^2 + \left(\frac{\left(P_d\left(0,1\right) + P_d\left(1,0\right)\right)}{h}\right)^2 + \left(\frac{P_d\left(1,1\right)}{h^2}\right)^2 \right)} \tag{3.10}$$

where $P_d = P_i - P_{Ti}$. The second metric used is Bhattacharyya distance d_B which is normally used to determine the similarity between two discrete or continuous probability distributions. In this case to facilitate a pure covariance comparison the mean of the lines assumed to be the same, hence the a simplified version of the general Bhattacharyya dis-

Method & Category		RMSE		STD	
		$d_F\ (m)$	d_B	$d_F\ (m)$	d_B
Hough uncertainty	a	0.007680	0.263300	0.004812	0.173254
	b,c	0.011031	0.286885	0.002429	0.148899
Constant	a,b,c	0.002867	0.165336	0.001042	0.117617

Table 3.4: Uncertainty estimation technique comparison.

tance is used as,

$$d_B = \frac{1}{2}\ln\left(\frac{\det P}{\det P_i \cdot \det P_{Ti}}\right)$$

where $P = {}^{(P_i+P_{Ti})}/_{2}$ for estimated covariance P_i and ground truth P_{Ti}. Based on these measures the accuracy and precision of the two uncertainty estimation techniques for various method categories is shown in Table 3.4. Both the Frobenius distance and Bhattacharyya similarity indicate a good constant uncertainty provide a better covariance estimation. From the table, it is clear that the addition of more points does not help the performance of Hough uncertainty. Additionally, a Hough based uncertainty has an added computational cost more than tenth of a second, unlike a constant deviation assignment which does not add anything. Hence, though the initial tuning of a constant deviation uncertainty assignment is complex, it is a better option for an online estimation.

3.5 Summary

In this chapter we presented details of the two main sonar sensors that we intend to use for the rest of this work. Then we proposed two ways of simulating these sonar sensors for validating the proposed line extraction algorithms and latter for navigation. One approach is to fit every beam with normal distribution while the second is to use a line uncertainty estimate to compute multiple line parameters and their corresponding sonar return points. Later we presented three well known line extraction algorithms together with a novel approach to add a sonar intensity weight for a more accurate line extraction. The experimental results show that a *Split and Merge* line fitting technique with an initial end-point fitting and a final weighted least square refinement has a superior speed and accuracy. The weights are obtained from raw sonar intensities which are typically discarded in similar applications. A constant covariance assignment does not only avoid adding a computational burden, it can also provide a good estimate of line uncertainty.

Chapter 4

Underwater SLAM Navigation

This chapter address the problem of underwater SLAM. Solving the SLAM problem involves the use of vehicle control commands and landmark feature based measurements to concurrently localise an AUV while generating an environmental map which subsequently can be used for localization. We have already established that line features are good for representing landmarks in a man made underwater structures. This chapter focuses on two of the most commonly used SLAM filters which are simple to implement: Extended Kalman Filter (EKF) and Unscented Kalman Filter (UKF). Later we will discuses the possible use of FInite Set STatistics (FISST) for representation of sonar measurement through Probability Hypothesis Density (PHD) filter. This chapter is a foundation for the next chapter which propose an attentive active SLAM for improved underwater navigation. This novel attentive active vision system is an added functionality on top of SLAM and can be integrated with any type of SLAM algorithm.

4.1 Bayesian Framework of SLAM

There are two main forms of SLAM: *online* and *full* SLAM. In an online SLAM a momentary estimation of the vehicle pose and map feature position is developed using a history of control commands and measurements up until that moment. A full SLAM problem, on the other hand, requires the estimation of posterior over the entire robot trajectory. However, in this work we are interested in the former which can be used to improve the vehicle navigation using external landmark features. Additionally, since most of the dead-reckoning drift in a typical AUV mission is constrained within the horizontal plane, we concentrate our effort on improving the 3 Degrees of freedom (DoF) robot state, X-Y position and heading. This simplification is brought by the level of accuracy that an altimeter or pressure sensor provide for depth estimation and the fact that in most applications the vehicle neither rolls nor pitches.

4.1.1 State Representation

Before going any further lets introduce the state representation adopted here and in subsequent chapters. At a time step k the robot state X_k is composed of the two-dimensional vehicle state X_v and line feature map M. The AUV state is given by positions (x, y) and heading ψ; hence

$$X_v = [x, y, \psi]^\intercal \,, \tag{4.1}$$

while the map comprises a set of N line features

$$M = [\rho_0, \theta_0, \dots, \rho_{N-1}, \theta_{N-1}]^\intercal \,, \tag{4.2}$$

where (ρ_i, θ_i) is the $i^{\{th\}}$ line feature with respect to the map origin $\{W\}$ which we assume to be the initial vehicle position. Hence the overall state is given as,

$$X_k = [X_v, M] \,. \tag{4.3}$$

The corresponding state uncertainty is given by a covariance matrix $\mathbf{P}_k$ which comprises covariance of the vehicle $\mathbf{P}_v$ and map features $\mathbf{P}_m$ as well as the cross covariance between them $\mathbf{P}_{vm}$:

$$\mathbf{P}_k = \left[\begin{array}{c|c} \mathbf{P}_v & \mathbf{P}_{vm} \\ \hline \mathbf{P}_{vm}^\intercal & \mathbf{P}_m \end{array}\right] \,. \tag{4.4}$$

4.1.2 Probabilistic SLAM Recursion

The propagation of robot state or new measurement acquisition is governed by probabilistic laws [19]. From the AUV point of view neither of these are deterministic, hence both measurements and location are represented using a stochastic distribution. The probabilistic definition of SLAM is represented as,

$$p\left(X_k|z_{1:k},u_{1:k}\right)=p\left(X_v,m|z_{1:k},u_{1:k}\right), \tag{4.5}$$

which is a joint probability density of vehicle pose X_v and landmark locations m for a given history of measurements $z_{1:k}$ and control command sequences $u_{1:k}$. In Bayesian representation the SLAM problem is solved in a two stage recursion: motion prediction and measurement update. In the prediction step a motion model $p\left(X_k|X_{k-1},u_k\right)$ is used to estimate the position of vehicle based on applied control u_k and previous state $p\left(X_{k-1}|z_{1:k-1},u_{1:k-1}\right)$. Hence the predicted robot state density is given by,

$$p\left(X_k|z_{1:k-1},u_{1:k}\right) \quad = \quad \int p\left(X_k|X_{k-1},u_k\right)\times p\left(X_{k-1}|z_{1:k-1},u_{1:k-1}\right)d_{x_{k-1}}. \tag{4.6}$$

On the other hand, the update step is used to correct any drift from this prediction posterior using new measurement z_k based on a measurement model $P(z_k|X_k)$. The corrected vehicle position is given by,

$$p\left(X_k|z_{1:k},u_{1:k}\right)=\frac{p\left(z_k|X_k\right)\times p\left(X_k,m|z_{1:k-1},u_{1:k}\right)}{p(z_k|z_{1:k-1},u_{1:k})}. \tag{4.7}$$

A close form solution for these recursive formulae were proposed by applying various approximations. For instance, in one of the earliest probabilistic SLAM solution by Durrant-Whyte and Leonard [29] an EKF filter is used with a strong linearity assumption. Subsequent research in the area have introduced various techniques which can perform well in a more non linear systems. However, this comes with an added computation complexity, which makes them less attractive for applications where the robot has a limited resource as in most AUVs. Hence selecting the right technique is a key in developing practical underwater SLAM algorithm. The remainder of this chapter presents some of the works we have done using various SLAM algorithms.

4.2 Extended Kalman Filter SLAM

The earliest SLAM algorithm [111] uses extended Kalman filter (EKF) which is based on a non-linear extension of Kalman filter. An EKF-SLAM is a closed form SLAM solution where the a state transition and measurement models are approximated using a first order Tylor expansions. To simplify the discussion on EKF-SLAM lets put an alternative representation of the motion and observation models, discussed in 4.1.2, using two function $f(\cdot)$ and $h(\cdot)$ which represent vehicle kinematics and geometry of observation, respectively. In the derivation of EKF-SLAM algorithm these two functions are assumed to be slightly non-linear which could easily be approximated using first order Taylor expansions. Using these two functions the state transition and measurements can be re-written as

$$X_k = f\left(X_{k-1}, u_k + w_k\right), \tag{4.8}$$

$$z_k = h\left(X_v, m\right) + v_k, \tag{4.9}$$

where w_k is Gaussian additive velocity noise with zero mean and a covariance of Q_k, and v_k is zero mean Gaussian observation error with a covariance of R_k. Assumptions of linearity and additive Gaussian noise are key in EKF-SLAM which uses a normal distribution to approximate the probability densities in Equation (4.6) and Equation (4.7). One advantage of a normal distribution is it can be easily represented using two variables the mean and the variance as long as the system is linear or in this case turned in to a linear one. Therefore, in an EKF-SLAM the vehicle state is represented as a normal distribution $\mathcal{N}\left(;\hat{X}_k, P_k\right)$ where only the mean $\hat{X}_k$ and the system covariance P_k are propagated through time. Hence the standard EKF-SLAM recursion can be defined as follow:

a) Filter initialization

The SLAM recursion is started by assuming the initial vehicle position to be the map origin. At this stage the vehicle position is assumed to be well known with zero uncertainty, or practically with a small non zero uncertainty for a reason to be discussed in Section 4.3. Initial robot state $\hat{X}_0$ and P_0 are given by:

$$\hat{X}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}; P_0 = \begin{bmatrix} 1. & 0 & 0 \\ 0 & 1. & 0 \\ 0 & 0 & 1. \end{bmatrix} * \epsilon$$

where ϵ is a small uncertainty (*i.e.* $\epsilon = 1e - 4$).

b) EKF-SLAM time update

After an initial pose assignment, consecutive vehicle states are predicted using previous robot state and current control command as:

$$\hat{X}_{k|k-1} = f\left(\hat{X}_{k-1}, u_k\right) + w_k, \tag{4.10}$$

$$P_{k|k-1} = F_x P_{k-1} F_x^{\intercal} + F_w Q_k F_w^{\intercal}, \tag{4.11}$$

where F_x and F_w are Jacobians of $f(\cdot)$ with respect to the state and the velocity noise evaluated at the estimated state $\hat{X}_{k|k-1}$.

c) EKF-SLAM measurement update

For a given line measurement z_k and its corresponding i^{th} map feature observation $h_i\left(\hat{X}_{k|k-1}\right)$ the SLAM update is given by,

$$\hat{X}_k = \hat{X}_{k|k-1} + \mathcal{K}_k v_k, \tag{4.12}$$

$$P_k = \left[I - K_k H_k\right] P_{k|k-1}, \tag{4.13}$$

for an innovation vector v_k and its covariance S_k with a Kalman gain $\mathcal{K}_k$ which are given by,

$$\begin{aligned} v_k &= z_k - h_i\left(\hat{X}_{k|k-1}\right), && (4.14) \\ S_k &= H_k P_{k|k-1} H_k^{\intercal} + R_k, && (4.15) \\ \mathcal{K}_k &= P_{k|k-1} H_k^{\intercal} S_k^{-1}, && (4.16) \end{aligned}$$

here H_k is the Jacobian of $h_i(\cdot)$ evaluated at the estimated state $\hat{X}_{k|k-1}$.

A line feature from a sonar measurement can be either an introduction of a new landmark or a revisit for an older one or even a noise. This identification and determining matching landmarks across a given mission is handled by a process called data association. There are various techniques for solving the data association problem, however here a simple nearest neighbour (NN) criteria is used to determine the correspondence. Such technique is sufficient as long as the line features are well apart to avoid confusion between two different landmarks. After identifying the correspondence, it is validated by individual compatibility (IC) test using the Mahalanobis distance as

$$D_{ij}^2 = v_k^\intercal \left(S_k\right)^{-1} v_k < \chi_\alpha^2 \tag{4.17}$$

where α is the desired confidence level for 2-DoF Chi-squared statistics.

d) EKF-SLAM state augmentation

Any measurement that fails to satisfy the criteria in Equation 4.17 is considered as a new feature and appended to the current vehicle state. However since a new measurement $z_k = [\rho, \theta]^{X_v}$ is given in the robot frame, first it need to be transformed to map frame before adding it to the current state. The transformation of line feature from robot frame to map frame is done using the function $g(\cdot)$, which at z_k is given by,

$$g(z_k) = \begin{pmatrix} \rho + x\cos(\theta + \psi) + y\sin(\theta + \psi) \\ \theta + \psi \end{pmatrix}. \tag{4.18}$$

After appending this transformed feature into the state the new state covariance becomes,

$$P_k = \left[\begin{array}{c|c} P_{k|k-1} & \left(G_x P_{k|k-1}\right)^\intercal \\ \hline G_x P_{k|k-1} & G_x P_{k|k-1} G_x^\intercal + G_z R G_z^\intercal \end{array}\right] \tag{4.19}$$

where G_x and G_z are the Jacobian of the transformation function $g(\cdot)$ with respect to the robot state and the new measurement respectively.

4.2.1 State Transition Model

The control command u is obtained from a DVL and FOG/compass reading. It is composed of forward velocity v_x, lateral velocity v_y and rate of rotation ω which, in case of compass, is calculated based on the rate of heading change between two consecutive acquisition. These velocities are assumed to have added white Gaussian noise with a covariance Q. Using these velocities the motion model $f\left(\hat{X}_{k-1}, u_k\right)$ can be represented by a 2D constant velocity model as:

$$\begin{bmatrix} x \\ y \\ \psi \\ \rho_i \\ \theta_i \end{bmatrix}_{k|k-1} = \begin{bmatrix} x + \frac{v_x}{\omega}\left(\sin\left(\psi + \omega T\right) - \sin\left(\psi\right)\right) + \frac{v_y}{\omega}\left(\cos\left(\psi + \omega T\right) - \cos\left(\psi\right)\right) \\ y + \frac{v_x}{\omega}\left(\cos\left(\psi\right) - \cos\left(\psi + \omega T\right)\right) + \frac{v_y}{\omega}\left(\sin\left(\psi + \omega T\right) - \sin\left(\psi\right)\right) \\ \psi + \omega T \\ \rho_i + 0 \\ \theta_i + 0 \end{bmatrix}_{k-1} \tag{4.20}$$

and as $\omega \to 0$ Equation (4.20) becomes,

$$\begin{bmatrix} x \\ y \\ \psi \\ \rho_i \\ \theta_i \end{bmatrix}_{k|k-1} = \begin{bmatrix} x + v_x T\cos\left(\psi\right) - v_y T\sin\left(\psi\right) \\ y + v_x T\sin\left(\psi\right) + v_y T\cos\left(\psi\right) \\ \psi + \omega T \\ \rho_i + 0 \\ \theta_i + 0 \end{bmatrix}_{k-1} . \tag{4.21}$$

The difference between Equation (4.20) and (4.21) is limited for cases where the time step T is small given a typical maximum AUV velocity is under 0.5 m/sec. In a typical SLAM missions Equation (4.21) is good enough because most of the discrepancy between the two models can be corrected in the EKF-SLAM update stage. However in the absence of measurements, we stick with the motion model in Equation (4.20) for subsequent prediction to avoid any accumulation of drift.

4.2.2 Sonar Observation Model

An important element of SLAM algorithm is the measurement update which adjusts navigation drift based on landmark error. This correction is done based on an observation model. In order for the SLAM algorithm to succeed the observation model needs to be as accurate as possible. In this section we discuss a piecewise-defined observation model for a line segment feature based sonar measurement.

In Section 3.2.1 a Hessian normal form of a line segment is proposed as the best alternative for representing sonar line features. Since angles are periodic in polar representation, a single line can be expressed in multiple forms. For instance, a polar line given by (ρ, θ) can also be represented as$(-\rho, \theta + \pi)$. For much of the SLAM algorithm this does not cause problem or can be controlled; however, when it comes to computing innovation vector, this can make filters diverge. Hence, as discussed in [3] line segment observation requires two models. For the i^{th} map line $m_i = [\rho_i, \theta_i]$, the two observation model are given by,

$$h\left(\hat{\mathbf{x}}_k, m_i\right) = \begin{bmatrix} \rho_i - x\cos\theta_i - y\sin\theta_i \\ \theta_i - \psi \end{bmatrix} \tag{4.22}$$

and

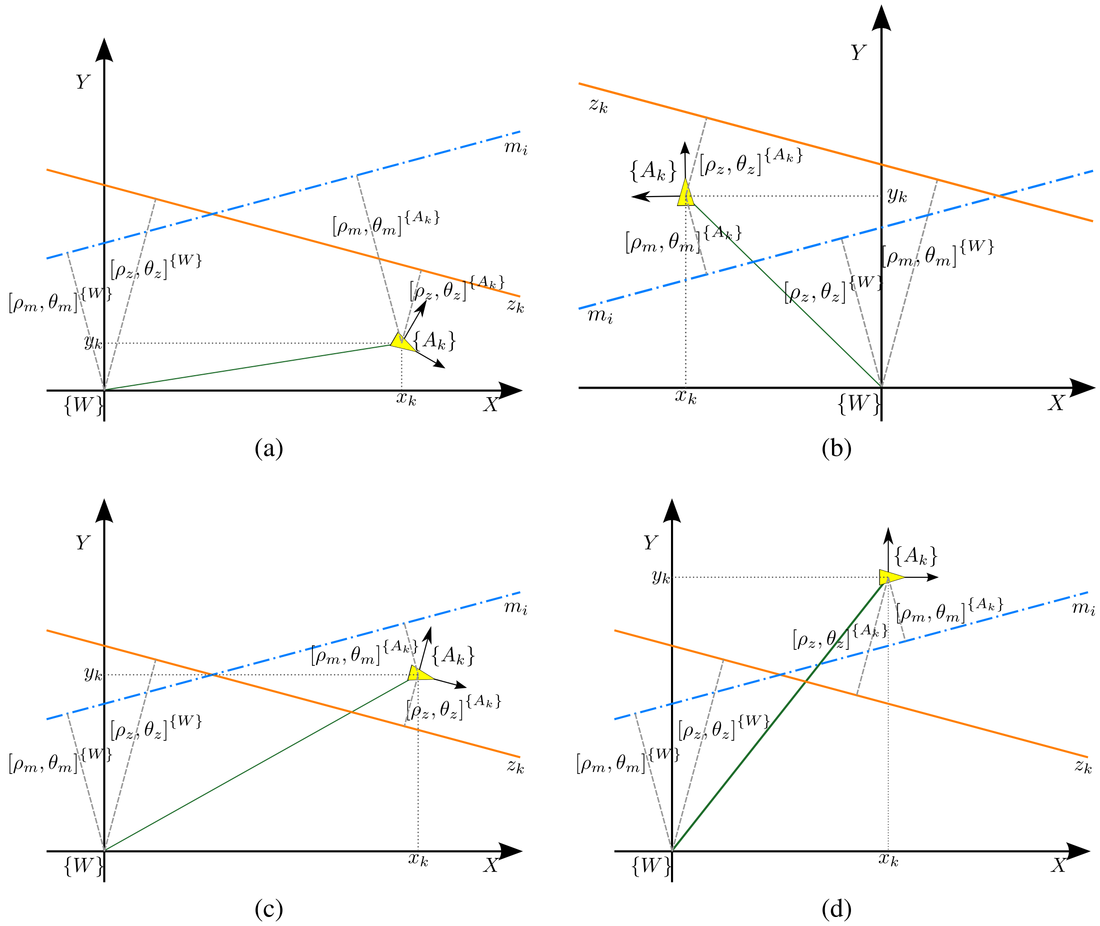
$$h\left(\hat{\mathbf{x}}_k, m_i\right) = \begin{bmatrix} -\rho_i + x\cos\theta_i + y\sin\theta_i \\ \theta_i - \psi + \pi \end{bmatrix}. \tag{4.23}$$

Probably the most important question now is how to choose between Equation (4.22) and Equation (4.23). In [3], this choice is made based on whether the line segment joining current robot position with the map origin intersect with the line m_i or not. On the other hand, in [2] only Equation (4.22) is used and all the problems with the observation model are claimed to be attributed to the increasing non linearity of the model as the robot goes far out from the map origin. Obviously, both the non linearity as well as a wrong observation model play a role for degrading the performance of SLAM navigation, yet it is important to identify which issue is causing a particular error.

A wrong observation model can easily cause filter divergence which is caused by an exaggerated innovation in Equation 4.12. However, the innovation vector can also go wrong if the sonar measurement is given in a wrong polar form. For clear understanding of the problem consider the example shown in Figure 4.1. A line is observed from different AUV positions $\{A_k\}$ at time k, based on the position of measurement line z_k and map line m_i with respect to the map origin $\{W\}$ and $\{A_k\}$ there are four possible scenarios.

Scenario 1: This the most common robot pose configuration which is applicable for most indoor robotics where a vehicle does not go past a boundary line. In such configuration the line joining current robot pose with the map origin does not intersect with either the new measurement line z_k or the associated map feature m_i as shown in Figure 4.1a. In this case the observation model in Equation 4.22 is sufficient.

Scenario 2: This is less common robot pose configuration which typically happens when the AUV observes a line which is close to it or when the vehicle pass by a convex corner. In such scenario, although the measurement line z_k lie on one side of both the



Y
m_i
$[\rho_m,\theta_m]^{\{A_k\}}$
$[\rho_z,\theta_z]^{\{W\}}$
$[\rho_z,\theta_z]^{\{A_k\}}$
$[\rho_m,\theta_m]^{\{W\}}$
z_k
y_k
$\{A_k\}$
$\{W\}$
x_k
X
(a)
Y
z_k
$\{A_k\}$
$[\rho_z,\theta_z]^{\{A_k\}}$
y_k
$[\rho_m,\theta_m]^{\{A_k\}}$
$[\rho_m,\theta_m]^{\{W\}}$
$[\rho_z,\theta_z]^{\{W\}}$
m_i
x_k
$\{W\}$
X
(b)
Y
m_i
y_k
$[\rho_m,\theta_m]^{\{A_k\}}$
$\{A_k\}$
$[\rho_z,\theta_z]^{\{W\}}$
$[\rho_z,\theta_z]^{\{A_k\}}$
$[\rho_m,\theta_m]^{\{W\}}$
z_k
$\{W\}$
x_k
X
(c)
Y
$\{A_k\}$
y_k
$[\rho_m,\theta_m]^{\{A_k\}}$
m_i
$[\rho_z,\theta_z]^{\{A_k\}}$
$[\rho_z,\theta_z]^{\{W\}}$
$[\rho_m,\theta_m]^{\{W\}}$
z_k
$\{W\}$
x_k
X
(d)

Figure 4.1: Possible arrangements of measurement line z_k and i^{th} map feature m_i with respect to the global frame $\{W\}$ and robot frame $\{A_k\}$. Lines are represented using polar form $[\rho, \theta]$ with respect to either $\{W\}$ or $\{A_k\}$ at time k.

robot and the map origin, the associated line m_i lie between the two (see Figure 4.1b). For such robot pose configuration a correct observation can be obtained by using Equation 4.23.

Scenario 3: Opposite to scenario 2 in this scenario the measurement line is between $\{W\}$ and $\{A_k\}$, while the observed map feature does not (see Figure 4.1c). For a robot in such configuration both Equation 4.22 and 4.23 will cause divergence. In stead first the measurement line need to be inverted, *i.e.*, $[\rho_z, \theta_z]$ becomes $[-\rho_z, \theta_z + \pi]$, then Equation 4.22 can be used.

Scenario 4: In the last scenario (see Figure 4.1d) both the observed feature and the measurement line intersect with the line segment joining $\{W\}$ and $\{A_k\}$. Here the measurement vector need to be inverted then Equation 4.23 can be used for correct innovation vector computation.

In a typical run the observed feature and measurement pairs alternate among the four possible configuration. Just modifying the observation model does not guarantee a correct innovation computation for all possible scenarios, even though it is possible to get a successful result for most of the time in a typical trail. For a complete remedy both the observation and the map feature need to be checked and modified accordingly.

The next question will be how to make this checks. For instance in Figure 4.2, to determine if the line $l^{\{A_k\}} = [\rho, \theta]$ intersect with the line segment joining $\{W\}$ to the current robot pose $\{A_k\} = [x, y, \phi]^{\{W\}}$, consider a point p on the line l which is a new frame on the line where the parallel component is along the line while the perpendicular component is along the line normal l_n. From this new co-ordinate frame the robot pose and the map origin are given by $\overline{pp_A}$ and $\overline{pp_0}$ respectively. The projection of these two vectors on the line normal gives $\overline{pp_A}_\perp$ and $\overline{pp_0}_\perp$ which mathematically can be expressed as,

$$\overline{pp_0}_\perp = \rho + x\cos(\theta - \psi) + y\sin(\theta - \psi), \tag{4.24}$$

$$\overline{pp_A}_\perp = \rho. \tag{4.25}$$

For the line to intersect the line segment joining the map origin with the robot origin the sign of the two projection has to be different. This check is done both for the measurement and observed line, adding a slight computation cost on the overall SLAM performance which will tackle a possible filter divergence arising from wrong innovation vector as it will be illustrated in Section 4.4.

4.3 Unscented Kalman Filter SLAM

As long as an AUV navigates along a straight path and it remains close to the starting position an EKF-SLAM can provide an optimal navigation performance. However, any vehicle rotation or an observation of a line segment far from the starting position will introduce a higher non-linearity to the motion model and observation model respectively. This strong non-linearity will eventually degrade the performance of an EKF based SLAM algorithm. The magnitude of this degradation depends on the angular speed of the rotation or the distance from the starting position to observation pose. As some of the works dis-

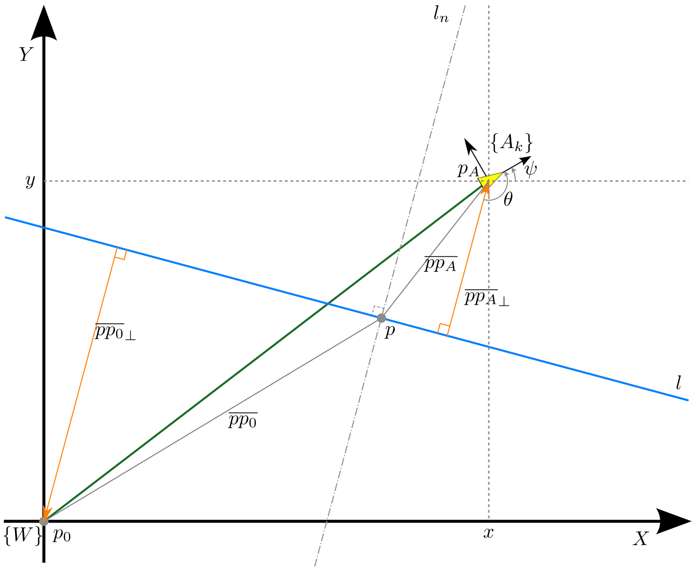


Figure 4.2: Intersection between a line and a line segment joining the map origin with current robot frame, which is used to determine the correct innovation distance.

cussed in Chapter 2 proposed, in applications where the vehicle covers a large area the use of sub-map is one option, yet care has to be taken to avoid a disconnection of adjacent or overlapping maps which result in missing loop-closure opportunities. This will be further demonstrated using an example in Section 4.4 while recreating Ribas *et al.* [2] result on abandoned marina dataset from St. Pere Pescador, Spain. Another option to resolve the shortcoming of EKF-SLAM is to use an alternative filter. One alternative would be using Unscented Kalman Filter (UKF). In this section, the theory and the implantation detail of UKF-SLAM is discussed with respect to our previous description of an EKF-SLAM.

As noted earlier, an extended Kalman filter propagates the robot state posterior, which is approximated by the mean and covariance of Gaussian probability distribution, using linear models. The UKF uses the original non-linear state transition or observation model to propagate multiple points in addition to the mean and covariance. These multiple points are generated through unscented transform using the posterior mean and covariance. Basic implementation of UKF has a slightly higher computational complexity than EKF, however recent work in [112] propose a square root UKF with complexity $O\left(N^2\right)$ as in EKF. Hence, the computational cost of UKF-SLAM can be be considered as that of EKF based SLAM.

4.3.1 Unscented Transform and Sigma Points

Unscented transform is a method that approximates a random variable for a non linear transformation. The notion behind unscented transform lies on approximation of a distribution with more than one deterministic sample points, instead of approximating the non linear transformation function as in EKF. These $2N+1$ sample points of a random variable with dimension N are called sigma points. For instance, if the state of the vehicle at time k is given by a mean $\hat{X}_k$ and covariance P_k, then the sigma points $\mathcal{X}_s$ are generated as follow:

$$\begin{aligned} \mathcal{X}_{s0} &= \hat{X}_k && (4.26)\\ \mathcal{X}_{si} &= \hat{X}_k + \sqrt{\lambda + N} Ps_i, \; i = 1, \ldots, N && (4.27)\\ \mathcal{X}_{si} &= \hat{X}_k - \sqrt{\lambda + N} Ps_{i-N}, \; i = N+1, \ldots, 2N && (4.28) \end{aligned}$$

where corresponding weights are computed as,

$$\begin{aligned} W_0^m &= {}^{\lambda}/_{(\lambda+N)} && (4.29)\\ W_0^c &= {}^{\lambda}/_{(\lambda+N)} + \left(1 - \alpha^2 + \beta\right) && (4.30)\\ W_i^m = W_i^c &= {}^{1}/_{2\cdot(\lambda+N)}, \; i=1,\ldots,2N && (4.31) \end{aligned}$$

here Ps_i is the i^{th} row of the covariance square root ($P_k = Ps^{\intercal} Ps$), $\lambda = \alpha^2 (\kappa + N) - N$ is a scaling parameter to match the third and higher order terms with α determining the spread of sigma points around the mean while κ is an additional scale factor which in most cases can be set to zero. β is factor that depends on the distribution type and it is optimal value is 2 for Gaussian distribution, while setting $(\lambda + n) = 3$ for such distribution can result in matching the fourth order terms [113].

4.3.2 UKF-SLAM Recursion

UKF-SLAM algorithm uses unscented transform for state prediction and measurement update. Since unscented transform uses square root covariance, an initial covariance has to be positive definite, this can be achieve by assigning a small non zero value to the diagonal elements of the initial covariance P_o, the initial robot pose is assumed to be the map origin as discussed in Section 4.2. As in EKF-SLAM subsequent robot states are determined using previous robot state and new measurements both for control and map feature. A

UKF-SLAM recursion include a prediction and update stages just like EKF-SLAM.

a) UKF-SLAM prediction

One way to propagate the robot state is by appending the control command into the robot state, *i.e.* the current control u_k is appended to $\hat{X}_{k-1}$ to give $\bar{X}_{k-1} = \left[\hat{X}_{k-1}, u_k\right]^{\mathsf{T}}$, where as the corresponding covariance Q is appended to state covariance to give new matrix $\bar{P}_{k-1} = diag\left(\hat{P}_{k-1}, Q\right)$. Based on this new state and variance using Equations 4.26-4.31.

$$\mathcal{X}_s, W^m, W^c \leftarrow sigma\_points\left(\bar{X}_{k-1}, \bar{P}_{k-1}, \alpha, \beta, \kappa\right). \tag{4.32}$$

Then these sigma points are transformed by using the state transition equation shown in Equation 4.20 or 4.21, the only difference here is that the control command is retrieved from the state and truncated from the output. Finally the predicted state and covariance are determined as follow,

$$\begin{aligned} \mathcal{Y}_s &= f\left(\mathcal{X}_s\right), & (4.33)\\ \hat{X}_{k|k-1} &= \sum_{i=0}^{2N} W_i^m \mathcal{Y}_{si}, & (4.34)\\ P_{k|k-1} &= \sum_{i=0}^{2N} W_i^c \left(\mathcal{Y}_{si} - \hat{X}_{k|k-1}\right)\left(\mathcal{Y}_{si} - \hat{X}_{k|k-1}\right)^{\mathsf{T}}. & (4.35) \end{aligned}$$

c) UKF-SLAM update

As in EKF-SLAM the data association is done using nearest neighbour classification and individual compatibility test. In this process, a new sonar measurement is either associated with a landmark in the sonar map or represented as a new map feature hence a UKF-SLAM update can either be an update or augmentation. First consider an update of the i^{th} landmark using a new sonar measurement z_k with covariance R. This starts by generating the sigma points and the corresponding weights from $\hat{X}_{k|k-1}$ and $P_{k|k-1}$. Then using the observation model for i^{th} landmark $h_i(\cdot)$ in Equation 4.22, a UKF-SLAM update is done as follows:

$$\begin{aligned}
\mathcal{Z}_s &= h_i(\mathcal{X}_s), && (4.36)\\
\bar{\mathcal{Z}} &= \sum_{i=0}^{2N} W_i^m \mathcal{Z}_{si}, && (4.37)\\
P_{zz} &= \sum_{i=0}^{2N} W_i^c \left(\mathcal{Z}_{si} - \bar{\mathcal{Z}}\right)\left(\mathcal{Z}_{si} - \bar{\mathcal{Z}}\right)^{\intercal} + R, && (4.38)\\
P_{xz} &= \sum_{i=0}^{2N} W_i^c \left(\mathcal{X}_{si} - \hat{X}_{k|k-1}\right)\left(\mathcal{X}_{si} - \hat{X}_{k|k-1}\right)^{\intercal}, && (4.39)
\end{aligned}$$

then using this intermediate values, the measurement innovation v_k and filter gain $\mathcal{K}_k$ are computed, followed by the updated mean and covariance of the vehicle state.

$$\begin{aligned}
v_k &= z_k - \bar{\mathcal{Z}}, && (4.40)\\
\mathcal{K}_k &= P_{xz} P_{zz}^T, && (4.41)\\
\hat{X}_k &= \hat{X}_{k|k-1} + \mathcal{K}_k v_k, && (4.42)\\
P_k &= P_{k|k-1} - \mathcal{K}_k P_{xz} \mathcal{K}_k^{\intercal}. && (4.43)
\end{aligned}$$

On the other hand, if this is a new measurement, it needs to be transformed to the map frame and get augmented into the robot state. The procedure starts by appending the new measurement into predicted robot pose as $\bar{X}_{k|k-1} = \left[\hat{X}_{k|k-1}, z_k\right]^{\intercal}$ and $\bar{P}_{k|k-1} = diag\left(\hat{P}_{k|k-1}, R\right)$, then sigma points are generated which are then propagated using a transformation function $g(\cdot)$ shown in Equation 4.18. The robot state after the augmentation is computed as,

$$\begin{aligned}
\mathcal{Y}_s &= g(\mathcal{X}_s), && (4.44)\\
\hat{X}_k &= \sum_{i=0}^{2N} W_i^m \mathcal{Y}_{si}, && (4.45)\\
P_k &= \sum_{i=0}^{2N} W_i^c \left(\mathcal{Y}_{si} - \hat{X}_k\right)\left(\mathcal{Y}_{si} - \hat{X}_k\right)^{\intercal}. && (4.46)
\end{aligned}$$

One of the key things to consider in the above computation is the fact that angle difference or averaging follows a special rule. For instance the difference between $180°$ and $-180°$ is zero rather than 360°, hence care must be given to avoid wrong computation.

4.4 SLAM Experiment and Results

In this section the performance of EKF and UKF SLAM algorithms for sonar line feature based navigation is tested; first using a simulated environment then a real marina dataset.

4.4.1 Simulated SLAM Test

The simulated test environment used is shown in Figure 4.3, which covers an area of $224\,\mathrm{m}^2$. The simulated vehicle has a sonar sensor similar to the one discussed in Section 3.1.3 and a compass for heading measurement and DVL for forward and lateral velocity measurement. The compass measurement provides an angular rotation with an added white Gaussian noise of $1.15\,^\circ$/sec STD while the DVL error on the forward and lateral velocities is set to $5\,$cm/sec. The simulated multibeam sonar has a maximum range of $10\,$m with $90\,^\circ$ field of view like the Blueview MB2250 short range head. The lines are extracted using Split and Merge line fitting technique discussed in Section 3.2.5. For this test, the vehicle starts from the origin with a straight section until it starts turning to the left. Then it goes through a serious of straight and circular sections to get back to the starting position. The forward velocity is kept at $0.5\,$m/s while the rotational speed varies from $0.5\,^\circ$/sec in the straight sections to $8\,^\circ$/sec for the major turns. These significantly high speed rotations are intended to test the capability of filters in dealing with strongly non linear state transitions.

As shown in Figure 4.3 the dead reckoning path has diverged from the ground truth just after the first turn and the final pose is quite far from the starting position. On the other hand the EKF-SLAM does well until the major turn in the right side of the figure, where it miss the actual trajectory of the vehicle. The error in the EKF-SLAM is highly associated with the non linearity of the motion model in the circular section of the path. This error ultimately lead to a disorientation of the map. In comparison the UKF-SLAM performed better in all circumstances which is also accompanied by a better map representation. Figure 4.4 show the comparison of accumulated X-Y position error for DR, EKF and UKF SLAM. Errors are computed using Euclidean distance between estimated vehicle pose and ground truth at each time step. The DR error start diverging quite early while the EKF SLAM deviates only when the vehicle start turning. From the last section it is clear that an EKF

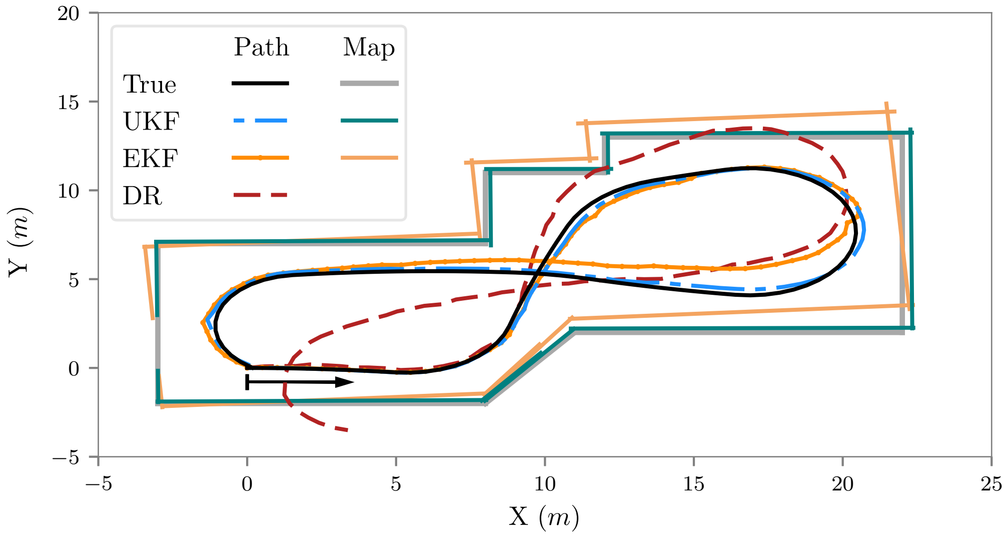


Figure 4.3: Map of simulated environment and estimated vehicle trajectory using DR (dead reckoning) EKF and UKF SLAM against the ground truth. The figure also show the estimated map boundaries.

SLAM converges back to the correct value when the vehicle re-observes old landmarks. In comparison, UKF -SLAM provides a better estimate throughout the trajectory. This can be further demonstrated using 2σ error plot as shown in Figure 4.5 for individual vehicle state dimensions. There are two important details to note from these plots:

- Uncertainty estimation of EKF is too optimistic, means at times filter estimation can get well outside the 2σ error bound, this will result in overconfidence that can cause wrong data association or new landmark initialization.
- AN EKF-SLAM provides a lower angular orientation estimation for the last section of the trajectory which in this case brought the vehicle back to the starting position but in the long run will lead to deviation from the actual trajectory.

Overall both EKF and UKF filter perform well in the straight section of the vehicle trajectory; however, the performance of EKF degrades as soon as the vehicle starts turning sharply.

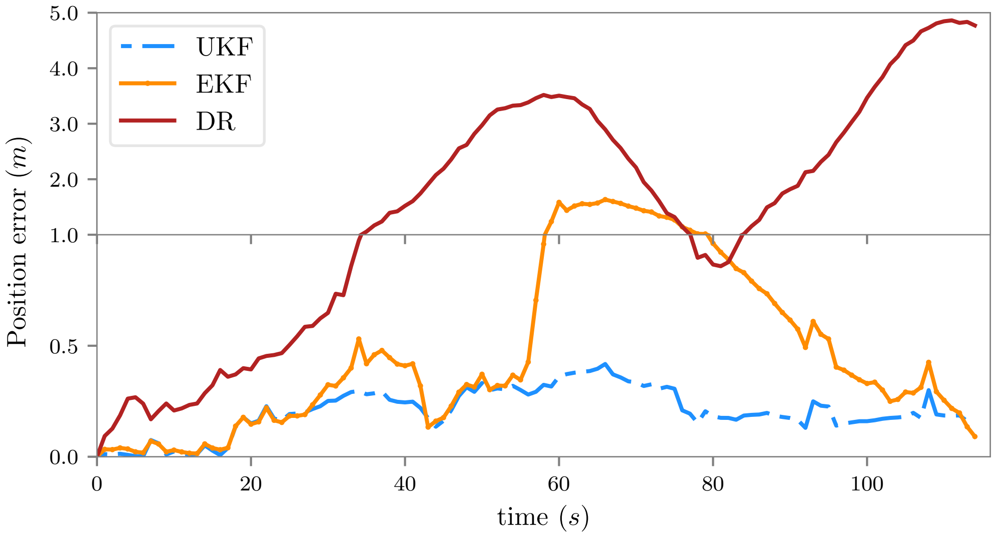


Figure 4.4: Simulated SLAM test accumulated position error comparison between DR, EKF-SLAM and UKF SLAM.

4.4.2 Marina Dataset

To test the performance of the proposed SLAM algorithms on a real dataset we choose the St. Pere Pescador, Spain abandoned marina dataset[1]. The dataset is collected using a Mechanically Scanned Imaging Sonar (MSIS) with a maximum range of 50 m, and beam resolution 0.1 m by 1.8 °. The vehicle covers a distance of 600 m moving at a maximum speed of 0.2 m/s for 50 min. The vehicle trajectory can be seen in Figure 4.7 where after an initial loop in the first section of the trajectory the vehicle take a long straight route to reach the final destination. The vehicle was operating close to the surface in order to take a ground truth GPS measurements at a rate of 14.5 hz, sonar line features are generated at a much slower pace at every 6 sec. To facilitate the comparison the line extraction is done before hand using the sonar measurement.

In addition to comparing EKF and UKF based SLAM, this test is intended to compare the performance of sub-map based EKF-SLAM proposed in [14] with the proposed single map SLAM framework with the right observation model. Our aim in this comparison is to show how considering the four possible observation scenarios can prevent divergence which might necessitate the use of small sub-maps. However it is important to mention that the authors in their recent works proposed a better performing SLAM techniques (see [27, 114]). Figure 4.7 show the performance of these techniques using vehicle trajectories. The

[1] http://cres.usc.edu/radishrepository/view-one.php?name=abandoned_marina

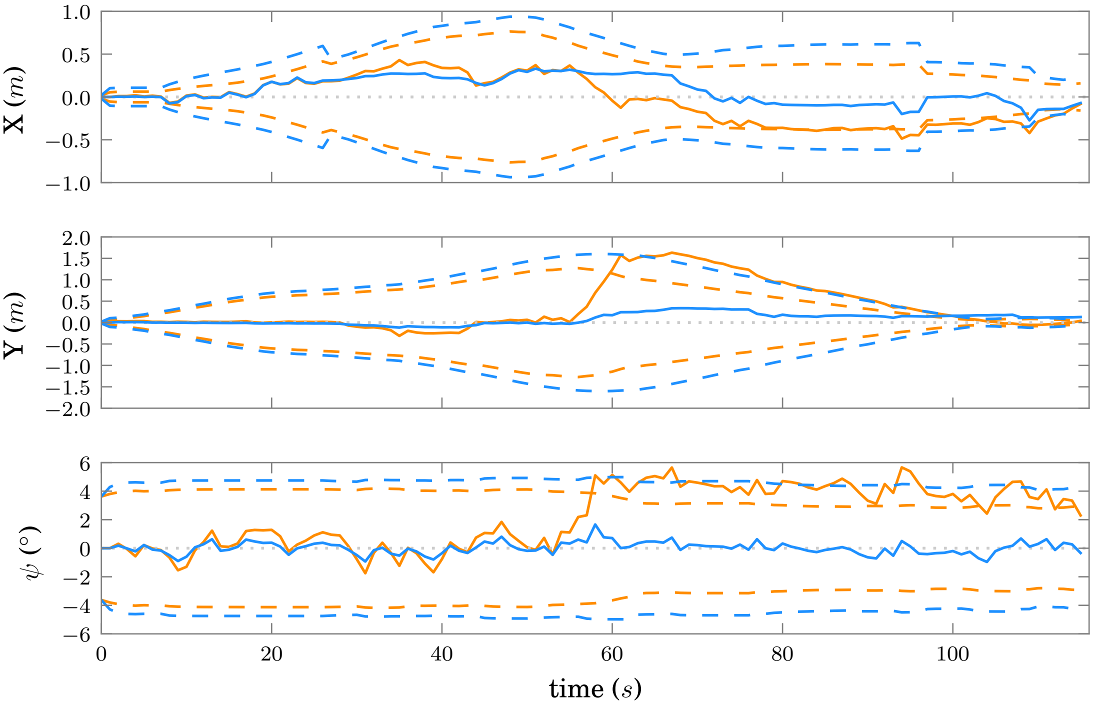


Figure 4.5: A comparison of AUV pose error plots (2σ bounds) for the resulting estimated trajectory. EKF-SLAM (orange) and UKF-SLAM (Blue) for the three axis (X, Y, and ψ).

trajectory from Ribas *et al.* EKF-SLAM misses the possible loop closure when the vehicle comes back to the starting position in the initial section. In this configuration the system uses two maps for going up and coming back. Even though the vehicle leaves the range of the first sub-map just briefly, there is no mechanism to pick up from where it left in the old map when it comes back again. Hence the overlap between the two sub-maps is ignored which caused the error in Ribas *et al.* trajectory to be higher than the proposed EKF and UKF SLAM filters for the section where the vehicle makes a return journey to the starting position. On the other hand, UKF-SLAM performs better in the final section of the trajectory where the measurement becomes unavoidably non-linear.

Figure 4.6 shows X-Y position error for St. Pere Pescador SLAM experiment. Errors are computed at each filter update stage based on an interpolated GPS measurement. Obviously, the use of a filter improved the result quite a lot. In comparison, the proposed filters with a single map performed better than that of a sub-map based SLAM. As stated before UKF-SLAM performs well in the later stage of the trajectory where there is a strong measurement non-linearity.

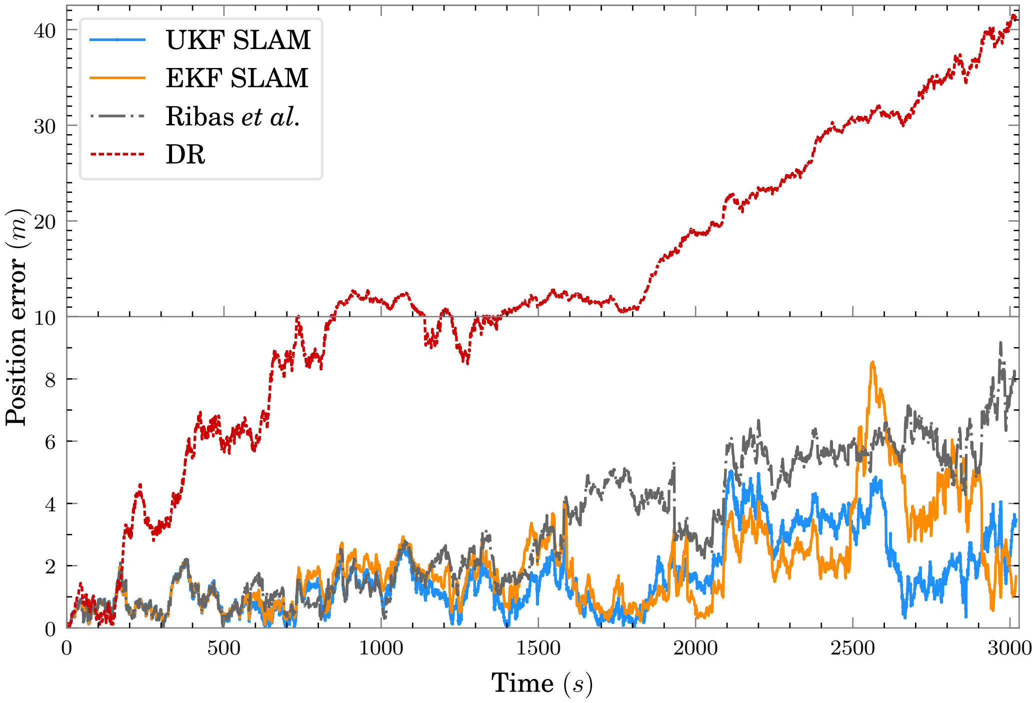


Figure 4.6: Abandoned marina experiment SLAM position error for proposed EKF and UK SLAM as well as Ribas *et al.* sub-map based SLAM.

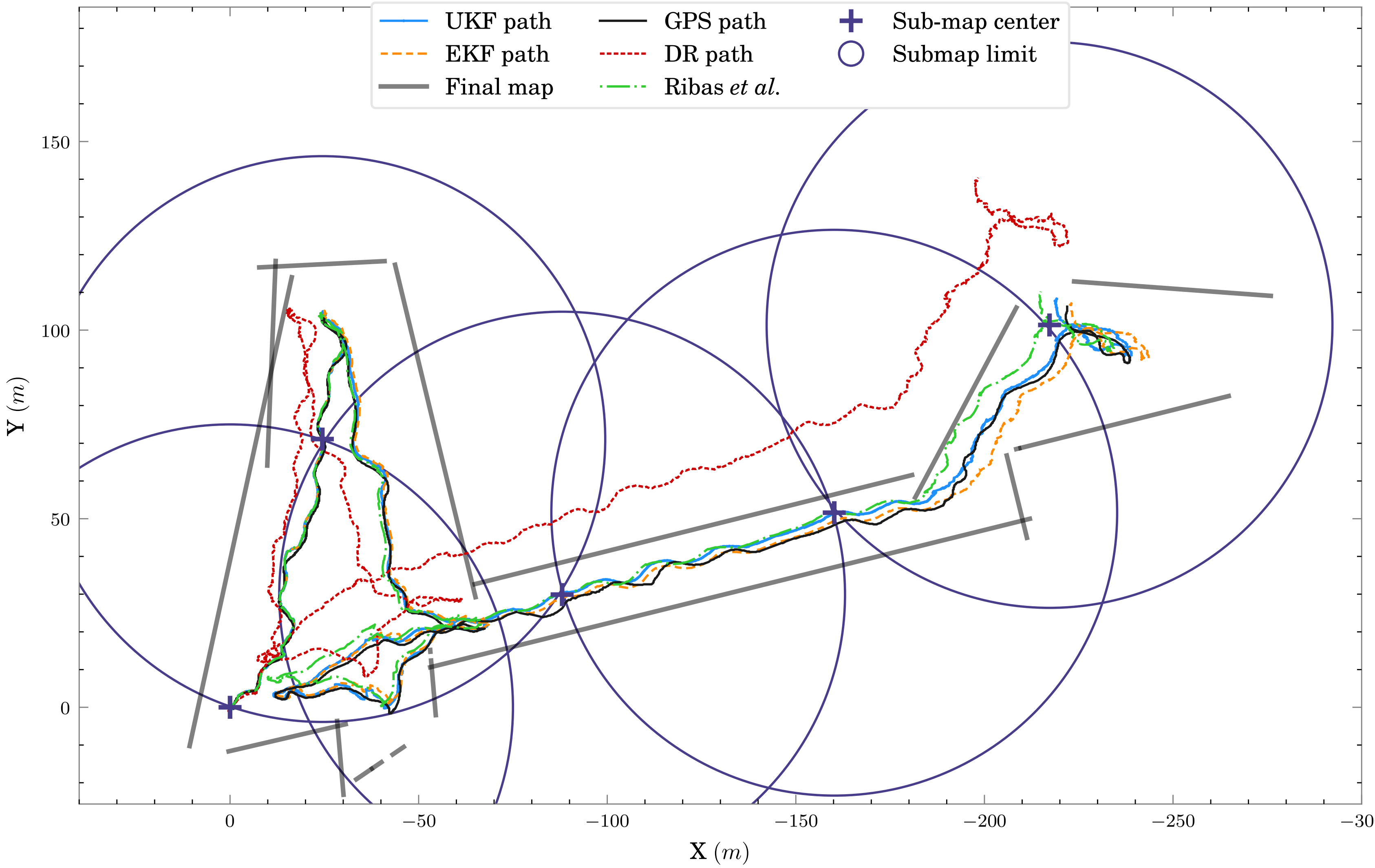


Figure 4.7: St. Pere Pescador, Spain abandoned marina AUV navigation experiment.(dataset source [14])

4.5 Probability Hypothesis Density Filter

One of the goals for this work (as mentioned in Chapter 1) is to explore the possible use of Probability Hypothesis Density (PHD) filter in SLAM framework. Recently there has been a lot of research outputs that propose PHD filter for various tracking applications. Most of the available SLAM techniques use filters that stem from single target Bayesian filter. These filters are extended using multi-target Bayes filter in applications where there exist static or dynamic multiple targets [115]. However, these filters become computationally intractable when the number of targets is very high or the measurement is extremely noisy. These problems led to the birth of FInite Set STatistics (FISST) based multi-sensor multi-target filter which represents observation and state using Random Finite Sets (RFSs) [116]. A closed form approximated filter recursion is provided by propagating Probability Hypothesis Density (PHD) of the posterior multi-object state, hence the name PHD filter. PHD filter propagates only the *first order moment* or *intensity* of the multi-target probability density in a similar fashion that Kalman filter uses the mean of the posterior single-object state.

Two types of PHD filter implementation are widely available one based on Sequential Monte Carlo (SCM) and the other using Gaussian mixture (GM) which outperformed most traditional multi-target tracking filters [117, 118]. Recently, Mullane *et al.* [119] and Lee *et al.* [115] have introduced PHD filter for solving the SLAM problem in cluttered environment. In a typical PHD filter based SLAM framework the vehicle can be modelled as a random variable while the RFSs assumption is restricted to the landmarks which will reduce the complexity. These works claim that the PHD filter based SLAM technique has the following advantages:

- An RFSs based representation provides a way to handle uncertainties on the number and state of environmental features.
- A PHD filter is robust particularly in the presence of large data association uncertainty and measurement clutter.
- A PHD filter SLAM filter does not require a special data association procedure.

In this framework landmarks are assumed to be born (as they appear the first time) then get mature (when observed multiple times) then die (as they disappear). This will provide an efficient filter dimension management when dealing with moving objects. However in our application the targeted environments are less likely to suffer from such problem. It is rather important to keep old landmarks in memory so that a successful loop closure can be achieved at later stage when the same landmark reappears. Besides, the use of line features in stead of point features already provides a good amount of feature compression. The robustness regarding measurement clutter is not that important here because the noise level in current multibeam sonars is manageable. Hence, the only plausible reason to use FISST for our SLAM based navigation system will be to avoid complications arising from data association. Of course, it is also important to see if there is anything that gets lost by using PHD-SLAM, which could be either in accuracy of landmark representation or speed of computation. In the remainder of this Section, we will briefly demonstrate the work done to validate the performance of GM-PHD filter for sonar based mapping (for details see Abdella *et al.* [120]).

4.5.1 Gaussian Mixture-PHD Filter

One way of implementing PHD filter is using Gaussian mixtures (GM) which can be taken as a direct counter part for EKF in single target tracking. Vo and Ma [118] first proposed a GM-PHD filter for tracking multiple targets with linear Gaussian dynamics and Gaussian birth models. These assumptions are strongly sensible in our application since features are coming from stationary structures and noise in multibeam sonar sensors can be assumed to be Gaussian as demonstrated in Chapter 3.

When using FISST for solving the SLAM problem it is customary to represent the robot state using random vector since there is no ambiguity about the number of vehicles in the mission. This will also simplify the overall complexity of the system. Since the vehicle is represented in a similar approach as in EKF-SLAM discussed in Section 4.2, if there is a difference it has to be mostly through the map representation. Hence instead of comparing a complete SLAM technique it is more apparent to compare a PHD filter based mapping technique with a mapping technique that uses an EKF. Hence the vehicle state X_v (shown in Equation 4.1) assumed to be accurate and the filters are compared based on their feature representation accuracy.

The state X_k at time k is composed of the map features which are composed of a RFS that represent the location of line features in the space $\mathcal{X} \in \mathbb{R}^{n_x}$. Similarly, sonar measurement Z_k is modelled using a RFS which are sampled from a measurement space $\mathcal{Z} \subseteq \mathbb{R}^{n_z}$. The state and measurement at time k is given as,

$$X_k = \{m_{k,1}, m_{k,2}, \ldots, m_{k,N_k}\} \in \mathcal{F}(\mathcal{X}), \tag{4.47}$$

$$Z_k = \left\{z_{k,1}, z_{k,2}, \ldots, z_{k,n(k)}\right\} \in \mathcal{F}(\mathcal{Z}), \tag{4.48}$$

where $\mathcal{F}(\mathcal{X})$ and $\mathcal{F}(\mathcal{Z})$ are sets of respective collections of all finite subsets of $\mathcal{X}$ and $\mathcal{Z}$ while $m_{k,i} = [\rho_i, \theta_i]^\intercal$, N_k is number of map features in the state. Measurements in Equation 4.48 are given with respect to the map origin as (ρ, θ) pairs.

The filter recursion starts using previous time posterior GM intensities which can also be used to represent zero initial intensity,

$$\upsilon_{k-1}(x) = \sum_{i=1}^{J_{k-1}} w_{k-1}^i \mathcal{N}(x; m_{k-1}^i, P_{k-1}^i), \tag{4.49}$$

where w_{k-1}^i, m_{k-1}^i,P_{k-1}^i are the weight, mean and covariance of the i^{th} Gaussian component. Part of this intensity will survive in the next stage based on a survival probability $p_S(x_k)$ which is assumed to be independent of the sate in the derivation of the filter. The rest of the intensity introduced through an adaptive Gaussian mixture birth intensity $\gamma_k(x)$, which is driven by the measurement.

$$\gamma = \sum_{i=1}^{J_{\gamma,k}} w_{\gamma,k}^i \mathcal{N}(x; m_{\gamma,k}^i, P_{\gamma,k}^i), \tag{4.50}$$

where $m_{\gamma,k}^i$ and $P_{\gamma,k}^i$ are the location and uncertainties of new born intensities, respectively. Hence the PHD prediction is given by,

$$\upsilon_{k|k-1}(x) = p_S(x_{k-1})\upsilon_{k-1}(x) + \gamma_k(x). \tag{4.51}$$

However, the sum of two GM is another GM which can be rewritten as,

$$\upsilon_{k|k-1}(x) = \sum_{i=1}^{J_{k|k-1}} w_{k|k-1}^i \mathcal{N}(x; m_{k|k-1}^i, P_{k|k-1}^i). \tag{4.52}$$

If N_{k-1} is the number of feature from previous step the new number of feature after prediction $N_{k|k-1}$ becomes,

$$N_{k|k-1} = N_{k-1} \cdot p_S(x_k) + \sum_{i=1}^{J_{\gamma,k}} w^i_{\gamma,k}. \tag{4.53}$$

The next stage requires updating these intensities using the new measurement. Unlike single target EKF update, PHD filter update takes into account the possibility of missing a map feature in the current field of view from a new measurement which is determined by probability of detection $p_D(x_k)$. Assuming $p_D(x_k)$ to be constant thorough, the PHD update is given by,

$$\upsilon_k(x) = [1 - p_D(x)]\, \upsilon_{k|k-1}(x) + \sum_{z \in Z_k} \upsilon_{D,k}(x; z), \tag{4.54}$$

where

$$\begin{aligned}
\upsilon_{D,k}(x; z) &= \sum_{j=1}^{J_{k|k-1}} w^j_k(z)\, \mathcal{N}\left(x; m^{j(z)}_{k|k}, P^j_{k|k}\right) && (4.55)\\
w^j_k(z) &= \frac{p_D(x)\, w^j_{k|k-1} q^j_k(z)}{\kappa_k(z) + p_D(x) \sum_{l=1}^{J_{k|k-1}} w^{(l)}_{k|k-1} q^{(l)}_k(z)}, && (4.56)\\
q^j_k(z) &= \mathcal{N}\left(z; H_k m^j_{k|k-1}, R_k + H_k P^j_{k|k-1} H_k^T\right), && (4.57)\\
m^j_{k|k}(z) &= m^j_{k|k-1} + \mathcal{K}^j_k \left(z - H_k m^j_{k|k-1}\right), && (4.58)\\
P^j_{k|k} &= \left[I - K^j_k H_k\right] P^j_{k|k-1}, && (4.59)\\
\mathcal{K}^j_k &= P^j_{k|k-1} H_k^T \left(H_k P^j_{k|k-1} H_k^T + R_k\right)^{-1}. && (4.60)
\end{aligned}$$

As in Equation 4.52 this can be rewritten as a single GM as,

$$\upsilon_k(x) = \sum_{i=1}^{J_k} w^i_k \mathcal{N}(x; m^i_k, P^i_k), \tag{4.61}$$

where the total number of posterior targets is given by,

$$N_k = N_{k|k-1}\left[1 - p_D(x_k)\right] + \sum_{z \in Z_k} \sum_{j=1}^{J_{k|k-1}} w^j_k(z). \tag{4.62}$$

Form Equation 4.54 it is important to note the exponential growth of the number of intensities based on the measurements. Each predicted intensity is propagated by every measurement where only those matching intensities result in a significant intensity weight.

However, this process will add a computational burden on the filter as well as introducing too many insignificant intensities. The intensity expansion can be limited by using target management schemes where after an update closely spaced intensities are merged and only N_k number of intensities with the highest weights are propagated for the next stage. The mean of these N_k number of intensities gives the new map posteriori X_k.

4.5.2 PHD Filter vs KF for Sonar Based Mapping

In order to check for any improvement that can be achieved through the use of FISST based representation of sonar line feature measurement, here a comparison is made between KF and PHD filter based sonar mapping. The Kalman Filter (KF) implementation is similar to the EKF filter discussed in Section 4.2, except here the filter has just an update step where the state is explicitly the map state, and measurements are given with respect to the map origin. The later choice turn the observation Jacobian in to identity.

For this comparison a simulated arena shown in Figure 4.8 is used, where the test environment is composed of five walls. Among the walls, landmark 5 is a circular arc centred at $(3, 0)$, while the rest are a straight walls. The vehicle path is shown in the centre where for each run the vehicle starts at $(0, 0)$ with random orientation. Then after ten steps forward with random steering the vehicle goes back to the origin and repeat the process multiple times. At each step sonar measurements are taken then transformed to map origin using true vehicle pose and transformation function $g(\cdot)$(see Equation 4.18). Once transformed the measurements are altered using white additive Gaussian noise which is then fed to either of the two filters for mapping.

The constant probability of survival and detection are set at $p_S(x_k) = 0.97$ and $p_D(x_k) = 0.99$, assuming a significant survival chance and a very strong detection probability once the landmark is in the line of sight. For an initial test the sonar measurement noise is set to $[0.15m,\ 2°]$ STD, which is later increased to demonstrate the power of PHD to tackle cluttered measurement. The result of the first experiment is shown in Figure 4.9 where the final KF map state and the state of PHD filter throughout the run are shown over the measurements and the true position of the test arena walls. The circular section of the arena is represented using multiple tangent lines, hence the correctness of any line around landmark 5 is determined using the closest possible tangent. The map features from the KF are very close to the ground truth where the right side arc is represented using two lines. From

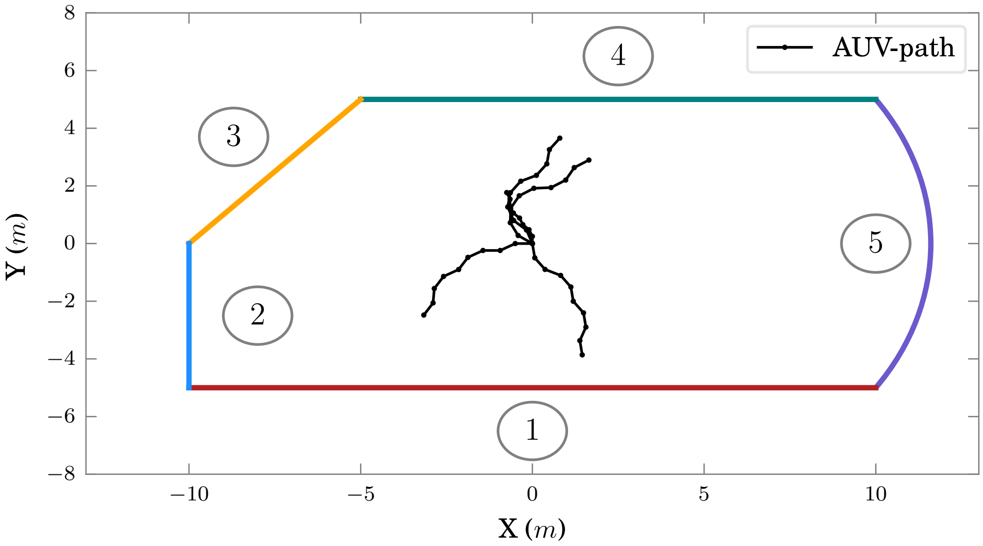


Figure 4.8: Simulated GMM-PHD filter test environment and AUV path taken for the experiment. Landmark 5 is an arc centre at $(3, 0)$ and radius $\sqrt{74}$.

the result, map features in PHD filter based mapping run back and forth around the actual ground truth in stead of converging on a single value as in KF. In case of the PHD filter all

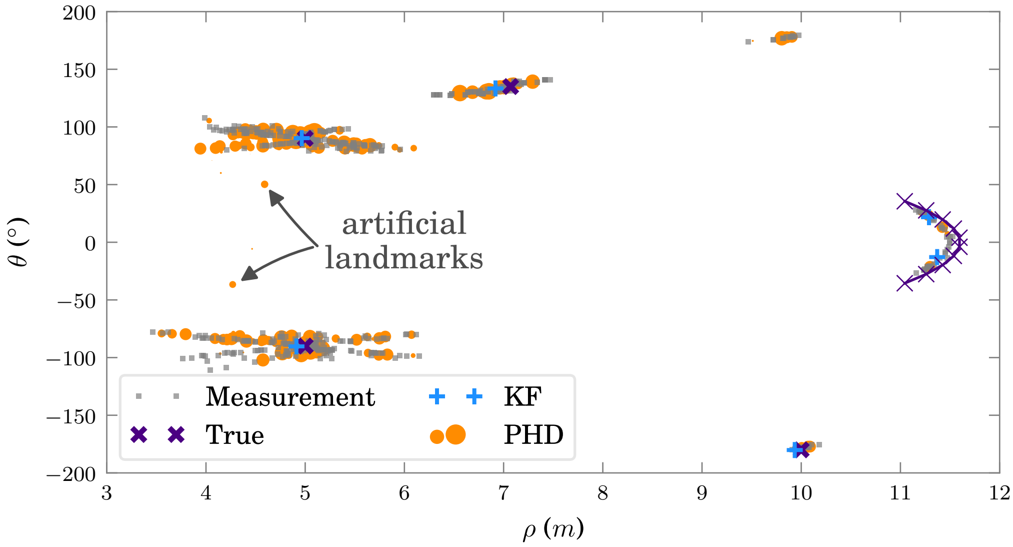


Figure 4.9: A comparison of PHD filter and KF based sonar mapping, map features are shown in parameter space against the measurement and ground truth.

the map features at different time steps are shown in the figure where the size of the marker indicates the magnitude of the corresponding intensity weight. This is mainly due to the fact that intestines in PHD filter can diminish unless there exists an extra map management system (which is not considered here). Representation of map feature in PHD filter can be demonstrated using visibility map in Figure 4.10. As show in the figure, features in KF mapping remain with in the filter once they are initialized even though the vehicle does not visit them in subsequent steps. However, in case of PHD mapping when intensities disappear from the view for a while their weight diminishes which eventually cause the intensity to die. For instance, landmark 1 is first introduced after the 10^{th} iteration then it disappears a little before the 30^{th} iteration. Then the same landmark is re-introduced after the 40^{th} iteration as a new landmark. This might not be a problem in case of target tracking or in the presence of plenty of landmark feature, in stead it will provide filter dimension reduction capability. However, in a line feature based representation the feature reduction has already been achieved and further reduction on the feature dimension in SLAM application will cause loop closure to fail.

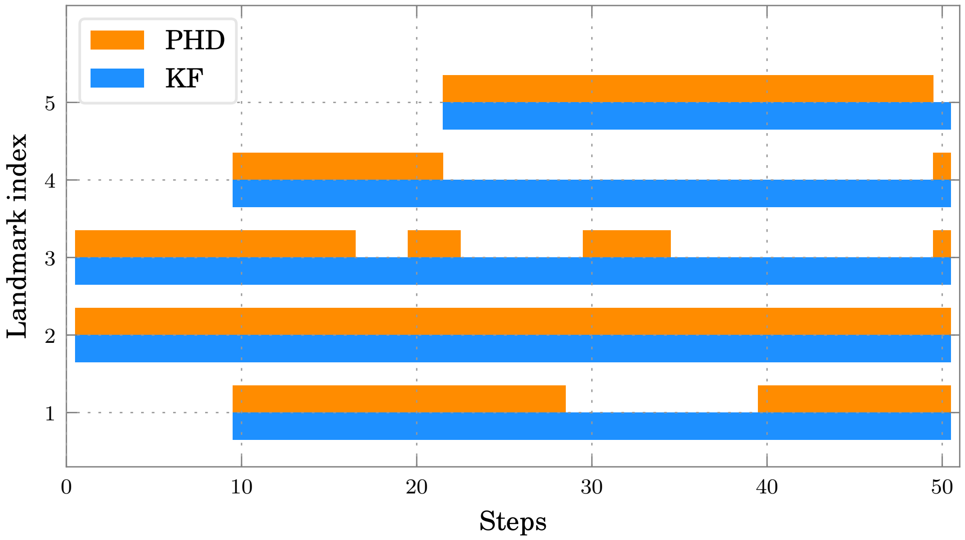


Figure 4.10: Comparison of map landmark visibility between KF and PHD filter based mapping. In PHD filter based sonar mapping landmarks can die and re-born. KF map features remains in the map once introduced.

To compare the performance of the two filters the landmark accuracy is measured at each time step. For comparing map features with the ground truth a distance measure d_h (see Equation 3.8) used where $h = 10°$. Figure 4.11 show the RMSE landmark error at each time step for the same experiment where landmarks are associated to the closest ground truth lines. KF mapping shows an improved landmark accuracy as they are observed more and more except for the slight increase in the total error as new landmark features are added into the filter. However, the same can not be said about PHD filter based mapping which generally has a higher error at all time in comparison. Specifically there are a couple of instances where the landmark error gets extremely high (around 24^{th} and 48^{th} iteration). This is due to artificial landmarks that are generated from a combination of two independent Gaussian kernels. Two of these artificial landmarks are shown in Figure 4.9, were these landmarks are generated by a combination of landmark 1 and 4 (top and bottom border walls). The Gaussian kernels of these two landmarks are overlapping and causing a third ghost landmark to appear between the two landmarks, which is associated with either of the two landmarks when computing the error and this will give rise to the occasional high peak in the total RMSE.

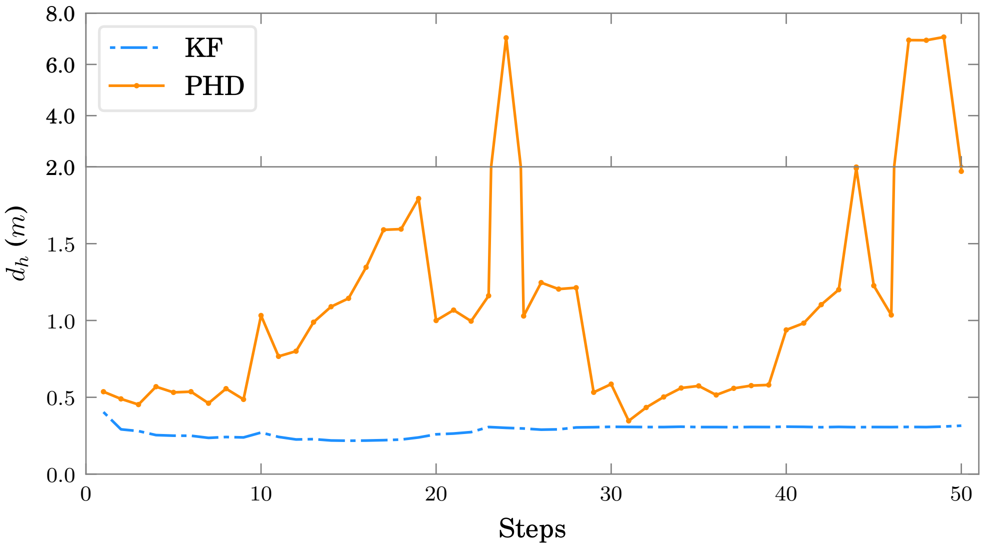


Figure 4.11: PHD filter vs KF based sonar mapping comparison of landmark RMSE error at each time step. Weighted euclidean distance is used to estimate the closeness between map feature and the closest ground truth line.

The real success of PHD filter can only be appreciated in the presence of stronger measurement noise in a more cluttered environment. In a later test the measurement noise is increased to $[0.5m, 5°]$ STD and after each feature extraction stage random lines are added into the measurement to emulate excess measurement clutter. The result is show in Figure 4.12 where except for landmark 4 all the other landmarks have stood out from the extreme measurement clutter. However, in a typical application, features from actual multibeam sonar are much less noisy. Hence, the use of such powerful filter in addition to being over engineering can result in a lower performance as demonstrated.

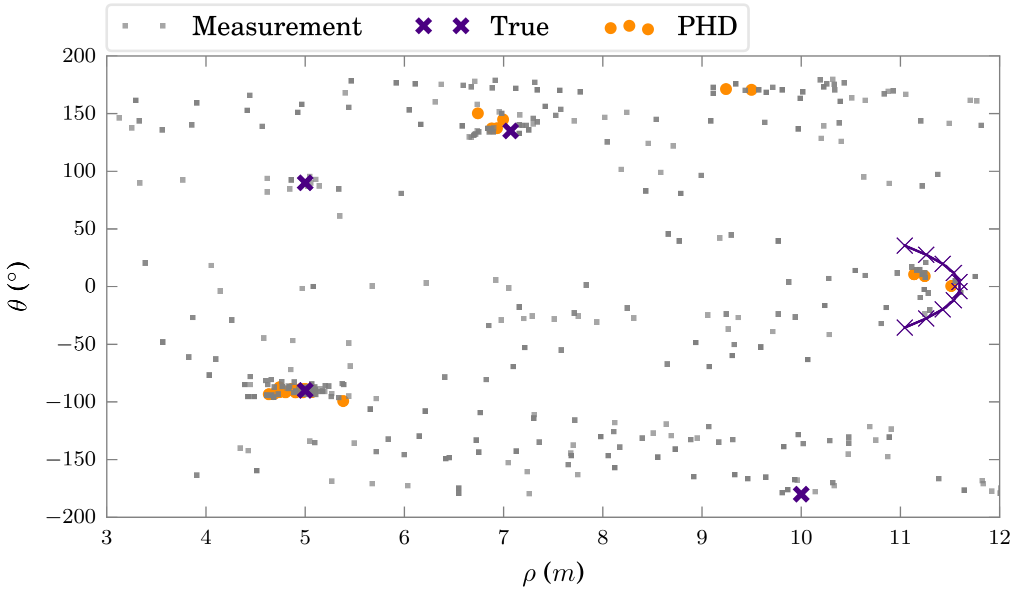


Figure 4.12: PHD filter based sonar mapping with a strong measurement error and clutter.

4.6 Summary

The chapter has presented underwater SLAM techniques using sonar based line features. It claims that the two main challenges in line feature based EKF-SLAM techniques are the use of correct observation model and dealing with strong non linearity either on the motion model or the observation. For the earlier, a piecewise observation function is provided where the choice depends on the position of the vehicle with respect to an observation, its associated feature in the map and the map origin. In the case of strong non-linearity UKF-SLAM is shown to out perform EKF-SLAM. On the other hand, the test on St. Pere Pescador marina dataset proved that the use of sub-map is not an ideal solution for a test environment under 300 m by 200 m. The final section of the chapter presented a PHD filter based sonar mapping as an alternative framework for highly cluttered measurement with huge uncertainty. However, having a multibeam sonars (BlueView P900-2250 and MB2250 see Chapter 3) with significantly smaller noise and gaining a good feature compression from a line segment based representation, the need for alternative framework is limited. Besides, for a typical low noise sonar measurement a standard Kalman Filter based mapping performs better than GM-PHD filter based sonar mapping. In the next chapter, the SLAM systems discussed here are integrated into a novel attentive navigation system.

Chapter 5

Focus of Attention for Underwater Navigation

In the preceding chapter, a sonar line feature based SLAM algorithm has been presented with a notion of passive sensing; hence, all the experiments and the evaluations involving multibeam sonar were performed by mounting an imaging sonar at a fixed angle relative to the vehicle (typically looking forward unless otherwise mentioned). In this chapter, these results are further extended using an active sensor set-up based on sonar focus of attention (SFoA). In addition to proposing a novel acoustic range measurement based saliency map generation technique, this chapter presents a comparison between passive and active/attentive underwater navigation.

5.1 Sonar Based Salience Map

As discussed in Chapter 2, computational attention researches so far base their perception of environment on visual images as in most primates. Hence, there has been no significant contribution of attention based sensing in the underwater environment where the predominant observation sensors are acoustic. Hereby, a first step towards filling this gap, we introduce a novel top-down acoustic salience map. A top-down salience map uses previous experience, deliberate plans or current goal in hand to determine the next best view orientation. The neurophysiological principle behind a top-down attention mechanism is less understood. Yet it is clear that top-down salience directs and focuses resources and neural activity to a particular location, feature or object of interest. Obviously, such type of de-

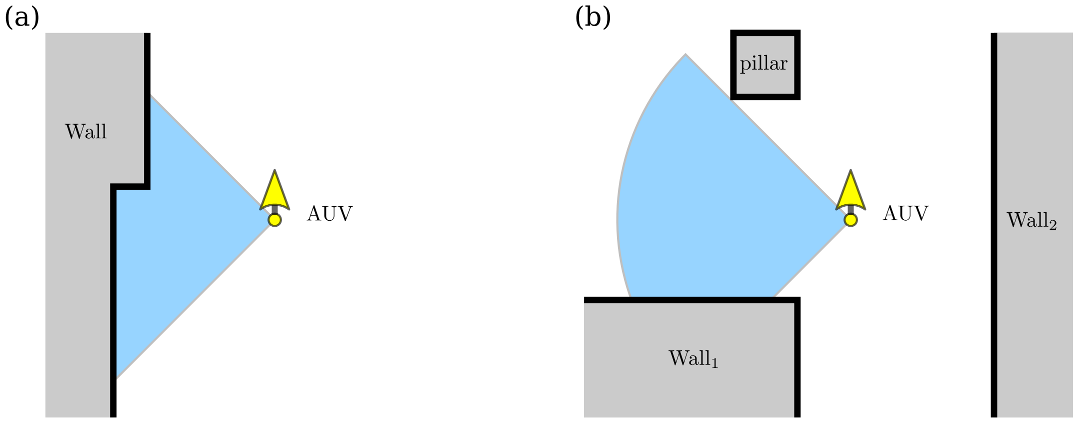


Figure 5.1: Where to direct a sonar for a better feature? In (a) the choice is between getting a feature or looking towards an empty space; however, in (b) the choice among multiple features around the robot which require salience map.

cision is easier when the choice is whether to look towards a wall or an empty environment (as shown in Figure 5.1). However it requires further analysis and the use of evaluation metrics when the decision involve choosing among the different walls or pillars. In such circumstances the decision metric is prepared as a salience map either as space based or object based salience (see discussion in Chapter 2). The first measures the salience of region around an observer while the later one uses the salience of nearby objects for determining the best object to look into. For our navigation application, a space based salience is more suitable which gives higher priority to good sonar features rather than a segmented or classified underwater objects. Besides, the most salient set of features can arise from multiple objects, further supporting the choice of space based salience.

For range based measurement, as in sonar sensing, a good alternative for synthesizing a salience map is the use of measurement constraints rather than features present within measurements. Hence, the next stage requires using sonar characteristics for formulating salience metrics which then leads to identifying the best features. From the discussion in Chapter 3, one of the key factors for obtaining a good beam reflection is to have a zero angle of incidence. In case of multibeam, the reference beam for angle of incidence is set at the middle because if the central beam is normal to a surface then the other beams will have at most an incidence angle equivalent to half the FoV of the sonar. Another factor to consider is the overlap between a feature and the sonar view range, a good feature is only good if it can be reached by rotating the sonar pan and tilt unit. The proposed salience metrics is based on such type of fundamental measurement constraints taken partly from the work of Chen and Li [84]. However a complete evaluation of these metrics require two other key ingredients: knowledge of features surrounding the vehicle, and as this is a space based salience map it also requires the representation of certain points around the vehicle for computing the salience value for. The knowledge of environmental features is obtained from a local map which is a transformation of the current SLAM map into predicted robot pose. In areas where there exist unknown sections of the environment around an AUV, a deliberate full scan is accomplished to enrich this map. At this stage, in order to reduce the computational complexity only those features which are close enough to the vehicle are considered to generate local map. This is achieved by first computing the distance from the current vehicle position to each line segments and their end points. Then a segment is removed if all the three distances are above the maximum sonar range. When it comes to salience computation points, 2D virtual grid points are generated all around the robot using empirically defined *grid spacing* g_s. These grid points are confined with in a circle of radius equal to the maximum sonar range. The grid spacing g_s is determined by the precision of the scanning device and the level of detail required for the final salience map. A very small g_s makes it difficult for the pan and tilt unit to differentiate between adjacent grid points while large g_s results in a reduced focusing capability (we will have more discussion on grid spacing in Section 5.4). Finally these grid points are assigned salience weight based on their relationship with line segments in the local map. The final salience weight is mainly a combination of three weights computed at each grid point per line segment: angle of incidence weight, overlap weight and distance weight.

5.1.1 Grid Point Generation

In an initial set-up, grid points are spread across a square area around the vehicle with dimension twice the maximum sonar range. The first grid point is put at top-left corner of this square then subsequent grid points are determined based on the grid spacing to result in grid point distribution as shown in Figure 5.2 (a). Alternately, the grid points can be generated in spiral order using g_s as the average distance between layers of the spiral as well as the distance between two constitutive grid points as shown in Figure 5.2 (b). Spirally generated grid points have a much broader range and orientation spectrum compared to a square grid points. In this configuration no two grid points lie at exactly the same distance

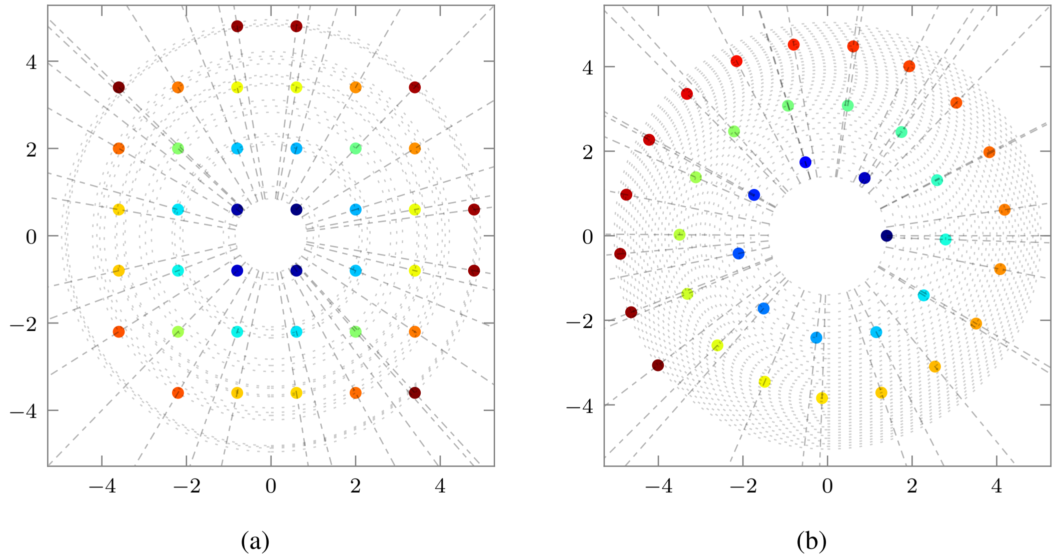


Figure 5.2: Attention grid types ($g_s = 1.4$ & maximum sonar range is $5m$). a) square grid, grid points share common orientation and range b) alternative spiral grid

from the vehicle though there might arise an occasional overlap in their orientation.

5.1.2 Angle of Incidence Weight

Angle of incidence weight G_a measures the quality of the sonar reflection for particular ensonified line feature when looking through a grid point. Sonar return is maximum when a sonar beam hits a flat structure at normal angle. In Figure 5.3, for line L$_i$ the angle weight distance d_a^i at a grid point P_2 is given by the angle between the line normal and the orientation angle of the grid point P_2 (*i.e.* $d_a^i = \theta_g - \theta_i$, where θ_g is the orientation of the grid point while θ_i is orientation of the i^{th} line segment). The corresponding angle of incidence Gaussian weight for N_L local lines is given as

$$G_a = \frac{1}{N_L} \sum_{i=0}^{N_L - 1} \eta_a \mathcal{N}\left(d_a^i; \mu_a, \sigma_a^2\right), \tag{5.1}$$

where η_a is a normalization factor, $\mu_a = 0$ and σ_a is set to $\pi/6$ so that the weight becomes very insignificant as the incidence angles approach to $\pi/2$. One things to notice from the figure, there are certain grid points thorough which we can possibly see multiple line segments including grid point P_2. However, grid points P_1 and P_3 will only allow L_i to be seen and between them P_3 will make it possible for most of the sonar beams to hit the line close to a normal angle.

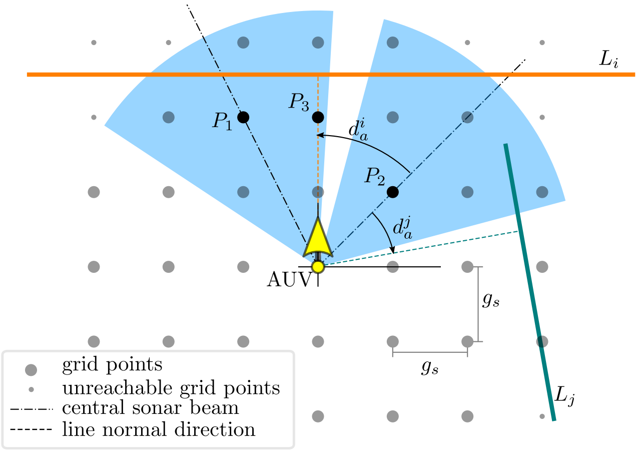


Figure 5.3: Angle of incidence attention weight, maximum for low incidence angle.

5.1.3 Overlap Weight

Overlap weight G_o signifies the section of a line segment that can be seen when pointing the sonar in the direction of a grid point. For instance in Figure 5.4 (a), the sonar points towards grid point P_1 with a sonar field of view S_{FoV} and the reachable span of line segment L$_i$ represented as a triangular span l^i_{span} for $i = \{1, 2, 3\}$. The overlap distance is computed as range of angle where the sonar S_{FoV} and l^i_{span} overlap, and the maximum overlap distance is S_{FoV}. Hence, for a grid point $P = (\rho_g, \theta_g)$ and a line segment $L_i = (\rho_i, \theta_i)$, the overlap distance d_o^i is given by

$$d_o^i = -|\theta_g - \theta_i| + \frac{\left(S_{FoV} + \mathrm{l}^i_{span}\right)}{2},$$

where angle difference $|\theta_g - \theta_i|$ is wrapped between $-\pi$ and π. After computing the overlap distance the overlap Gaussian weight at each grid point for N_L number of local map lines is given by

$$G_o = \frac{1}{N_L} \sum_{i=0}^{N_L-1} \eta_o \mathcal{N}\left(d_o^i; \mu_o, \sigma_o^2\right), \tag{5.2}$$

where η_o is an overlap weight normalization factor, and the mean and standard deviation are set as $\mu_o = S_{FoV}$ and $\sigma_o = {}^{S_{FoV}}\!/_2$ so that the weight diminishes slowly for an overlap distance below half the sonar view angle. Figure 5.4 (b) shows two grid points P_1 and P_2, through which either L_1 or L_2 can be seen. Even though both line segments have equal length, L_2 has a better overlap with the sonar FoV than L_1. As a result grid point P_2 has a

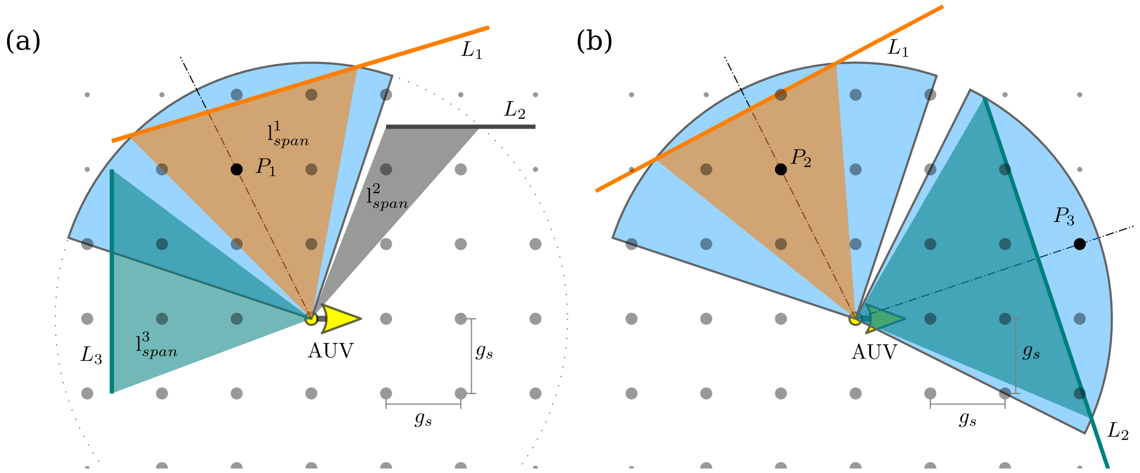


Figure 5.4: Overlap attention weight. (a) lines contributing for an attention weight at a grid point P_1, (b) two equal length line segments providing different weight for two attention points, maximum G_o from a line is obtained when it is with in a reachable distance.

greater salience than P_1 based on overlap weight.

5.1.4 Distance Weight

Distance weight G_d is an other weight which is incorporated to add temporal stability to the salience map. In Figure 5.5 (a) both grid points P_1 and P_2 have the same G_o and G_a for L$_2$ or L$_3$. Additionally, two grid points are more likely to have a higher overall salience since they are in the direction of an actual corner. Imagine the outcome if the robot moves slightly without updating the salience map as shown in Figure 5.5 (b). Looking through the grid point which is far from the line segments (in this case P_1) results a huge deviation from the actual corner for a slight robot position change. This will make the salience map temporally unusable. Hence G_d is introduced to encourage the selection of P_2 from P_1. The distance metric d_d^i is computed as the distance between a grid point P_g and a point P_i which is an intersection point between the line segment L$_i$ and the line connecting the sonar origin to the grid point, or one of the line end point in case the intersection is outside the line segment, $d_d^i = \| P_g - P_i \|$. Hence the corresponding distance weight for a sonar with maximum sonar range S_{r_max} is

$$G_d = \frac{1}{N_L} \sum_{i=0}^{N_L - 1} \eta_d \mathcal{N} \left(d_d^i; \mu_d, \sigma_d^2 \right), \tag{5.3}$$

where η_d is a normalization factor, $\mu_d = 0$ and $\sigma_d = {}^{S_{r\_max}}/_{6}$.

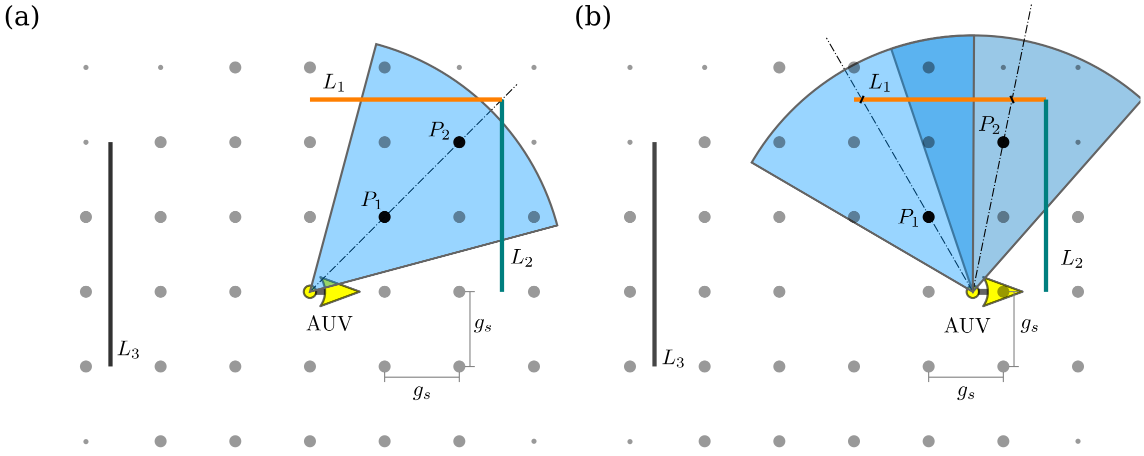


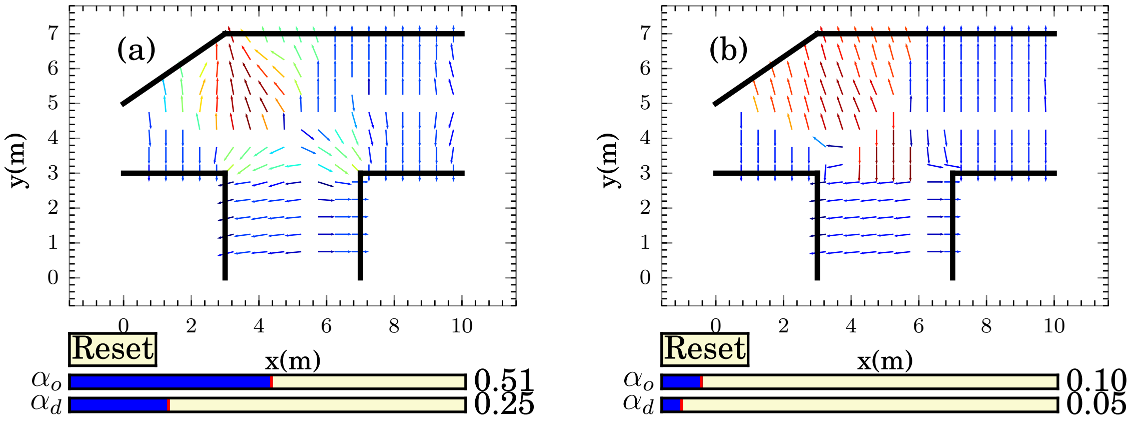
Figure 5.5: Attention distance weight. a) P_1 and P_2 have the same G_o and G_a, but P_2 has a better G_d, (b) using an old grid points after a small AUV movement.

5.1.5 Combined Sonar Salience

A scaled sum of the three salience weights gives an initial combined salience map S_{comb},

$$S_{comb} = \alpha_o G_o + \alpha_a G_a + \alpha_d G_d, \tag{5.4}$$

where α_a, α_o, and α_d are empirically defined parameters between $[0, 1]$ and the sum of the three weight remains to be one. The choice of good scaling parameters depends at least on the type of sonar sensor. For instance, angle of incidence weight is higher for P900-2250 compared to MB2250 due to a higher variation of intensity based on beam incidence angle. The best way to learn these weight factors is by plotting an attention flow diagram which is a map that shows the possible focus of attention (FoA) direction at different places of a given environment. The attention flow diagram shown in Figure 5.6 a) is obtained by setting the weight factors as $\alpha_a = 0.24$, $\alpha_o = 0.51$, and $\alpha_d = 0.25$, where the maximum sonar range is set at 8 m. As shown in the figure the salience map has managed to pick out corner points as a more informative features for SLAM navigation which is a desirable outcome since they can provide correction in 2-DoF. However, changing these parameters to 0.85, 0.1 and 0.05 respectively causes the FoA system to lose some of these corners as shown in Figure 5.6 b).



(a)

(b)

Figure 5.6: Tuning FoA parameters. a) acceptable combination of parameters (b) wrong parameters

5.2 Sonar Focus of Attention (SFoA)

Typically attention systems include Inhibition of Return (IoR) which is a disengagement mechanism which shifts attention from the current view to the next conspicuous location. However, the benefit of IoR is limited when the sensor directing unit is very slow and the environment through which the vehicle navigates does change often. In these type of situations it is rather preferable to avoid a pan and tilt unit oscillation or large changes of FoA orientation. Too much rotation will cause perception instability where the sensor, instead of keeping good features in sonar view, jumps from one to another before getting a firm grip of old landmark. Additionally, since ROS PT-10-FB RS-485 pan and tilt unit is mechanically limited between $-\pi$ and π, if the previous pan orientation is $-\pi + \epsilon$ (for a small positive angle ϵ) and a new FoA request is π, then the device has to make a full 2π counter clockwise rotation. This will take a while and by the time the sensor complete this large rotation, the vehicle might shift backward requiring a reverse rotation. Hence, it is rather preferable to keep the unit where it was, which is not far from the requested orientation in this particular case. In order to satisfy this requirement an additional map is proposed as Inhibition of Large Rotation (IoLR).

5.2.1 Inhibition of Large Rotation (IoLR)

Inhibition of Large Rotation (IoLR) is an inhibition map which is used to avoid large rotation of the sonar mounting device. This map is added as an additional sensor constraint map where at each grid point a weight is computed based on the previous time FoA orientation angle θ_{FoA}. For instance, the change of angle required to move the pan and tilt unit from the previous attention angle to the i^{th} grid point orientation θ_i is given by $d_{IoLR} = \theta_i - \theta_{FoA}$. Then the IoLR weight of this grid point is given by

$$S^i_{IoLR} = \eta_{IoLR} \mathcal{N}\left(\theta_i - \theta_{FoA}; \mu_{IoLR}, \sigma^2_{IoLR}\right), \tag{5.5}$$

where $\mu_{IoLR} = 0$, $\sigma_{IoLR} = {}^{\pi}\!/_{2}$ while η_{IOLR} is a normalization factor.

5.2.2 Attention Salience Map

In order to determine the final sonar sensor focus orientation, first the initial salience map is fused with IoLR map to generate the final sonar salience map S_{map}. The level of contribution from either of this two maps is determined by a constant $k \in [0 \ldots 1]$:

$$S_{map} = k * S_{comb} + (1 - k) * S_{IoLR}. \tag{5.6}$$

Finally, the FoA angle is determined by winner-take-all (WTA) technique where a grid point with the maximum salience value indicate the most salient location in space. The full procedure of our sonar salience map building and FoA generation technique is summarized in Figure 5.7. The vehicle is centred at the origin of a local map, from this position all the map wall line segments, except L_1 and L_4, are partially reachable. The dotted circle on the local map shows the possible sonar range for different sonar orientations while the previous sonar orientation is indicated as a shaded sector. The angle of incidence weight is pointing down word where L_6 and L_7 combine to provide a strong weight. However, both the overlap and distance weight picked the corner between L_2 and L_3 which is also the brightest spot on the initial salience map. The main difference between the initial salience map and the final salience map is the bottom left corner which appear more bright initially but fades due to an inhibition from IoLR map. Using WTA strategy the final focus orientation will be around $45°$ counter-clockwise from the vehicle heading.

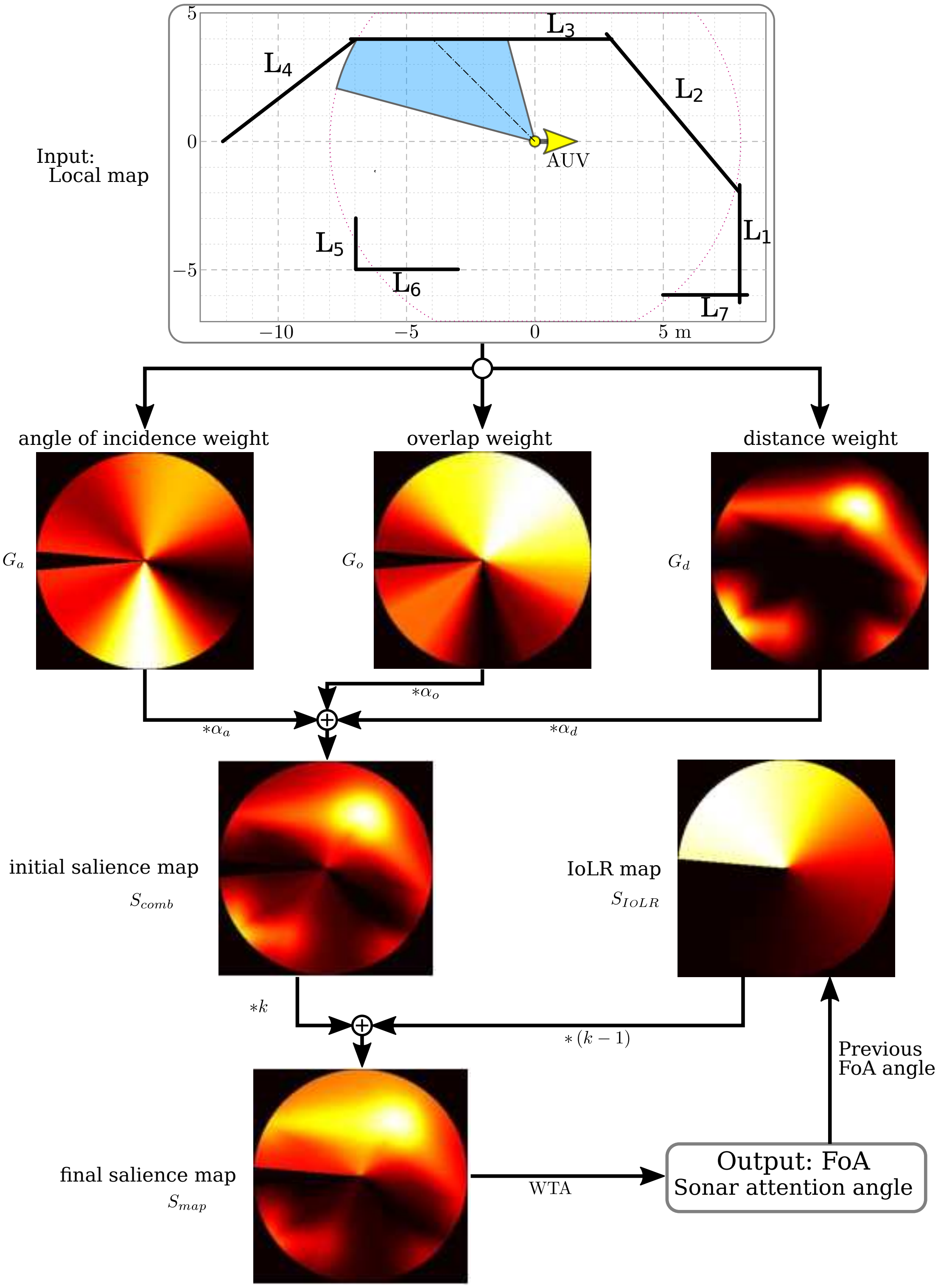


Figure 5.7: Top-down sonar salience computation using line segment based local map and sensor parameters.

5.3 Attention Based Active SLAM

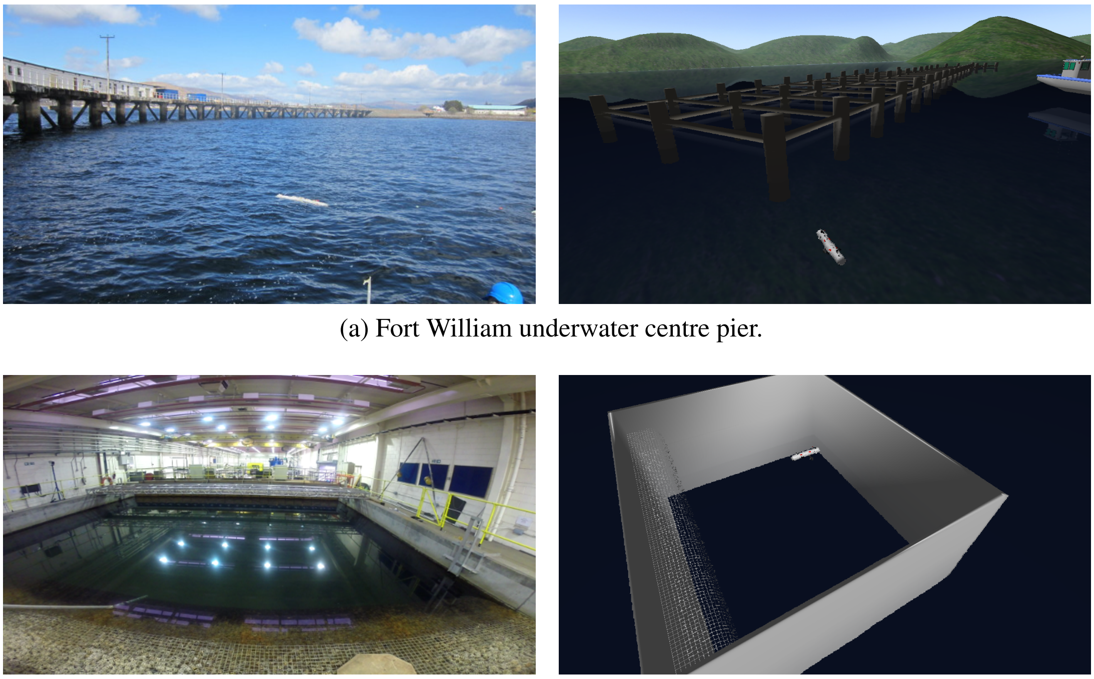
The main aim of this thesis is to devise a system that uses sonar salience map for improving underwater navigation (shown in the thesis outline Figure 1.4). In order to achieve Hence the last stage of this work requires substituting the passive SLAM discussed in Chapter 4 by an active FoA based SLAM algorithm. This is done by adding the FoA mechanism just before each observation to acquire features that can maximize measurement overlap between consecutive vehicle position or identify the best feature possible for a robust landmark mapping. Even though we have already established that UKF-SLAM can be far more superior than EKF-SLAM in various circumstances, here both filters are used to validate the proposed attentive system. The proposed FoA based system is filter independent and can be used on top of most SLAM algorithms as long as feature quality and quantity affects the performance of these systems. In the next section the proposed attentive navigation technique is first validated using simulation tests then in a real underwater environment.

5.4 Experimental Results

Testing the validity of the proposed system is done in two stages. First the algorithm is tested rigorously in a simulated environment. These tests provide the overall performance of the complete navigation system in comparison to a passive SLAM navigation. Additionally, the simulated tests are intended to determine the significance and importance of each parameters involved in the proposed system. The second stage of this test involve a real vehicle and/or a real multibeam sonar sensor test in the OSL tank and Wave tank (detail of these test facilities is provided in Section 1.2).

5.4.1 Simulated Tests

Before embarking on a real underwater test, it is important to determine the importance of each parameters introduced in the formulation of sonar salience map. A simulated test can also provide a controlled ground truth to compare the proposed attentive SLAM with an equivalent passive sensor based SLAM technique. Test environments are built for UWsim [4] through which a vehicle is driven for collecting sonar data as well as vehicle trajectory. Typical test environments include those shown in Figure 5.8. The figure shows



(a) Fort William underwater centre pier.

(b) Wave tank indoor test facility.

Figure 5.8: UWsim simulation of real world environment for initial testing. Actual images (left), and UWsim simulated images (right).

simulation of pillars and cross beams of Fort William underwater centre as well as walls and artificial beach of wave tank indoor test environment. However, in the some of the tests presented here, the wave tank structure is modified to match Figure 4.3 for extra complexity.

The sonar sensors for the test is devised based on the procedure discussed in Chapter 3. In all these tests a simulated Blueview MB2250 (see Figure 3.1) sonar sensor is used, although a similar result can be achieved by using P900-2250 sonar sensor with a different image pre-processing set-up. The simulated vehicle returns an integrated 2D velocity and position information for the test, where the positions are used for generating sonar measurements as well as a ground truth while the velocity sequences are used to generate noisy control commands. Next, the detail for the test arena set-up for all the simulated test is presented.

Simulation test arenas

Three test environments are used for validating the proposed algorithm and the different parameters. Before discussing the result from these tests, here are the details of each test set-ups with the corresponding arenas and motion trajectories.

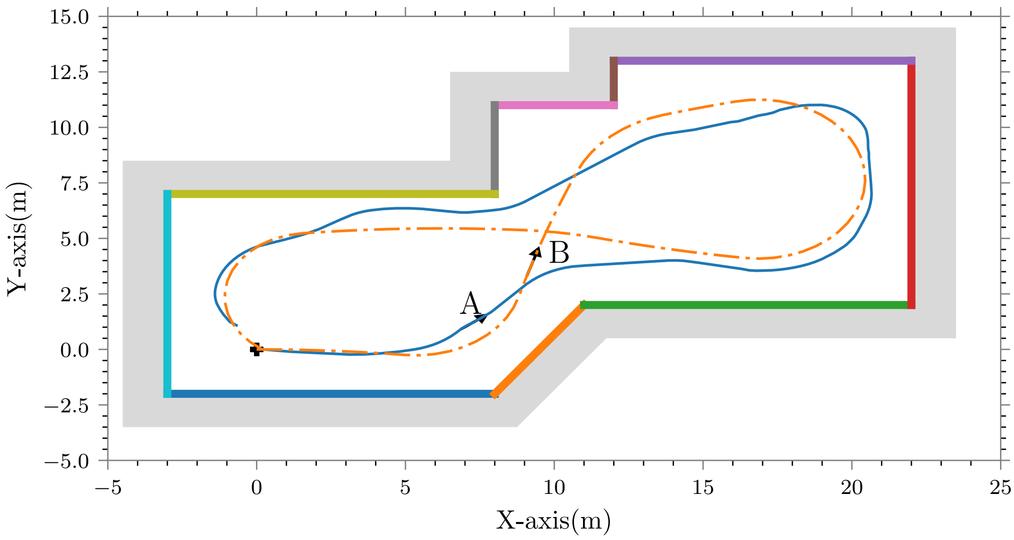


Figure 5.9: Test type-A and type-B map of simulated environment and trajectory. Thick lines indicate tank boundaries, while trajectory A and B are vehicle paths for test type-A and type-B respectively. In both trajectories the vehicle start from the map origin and return to a nearby point.

Test type-A : The environment for tests ;under this category is shown in Figure 5.9 where the ground truth trajectory of the vehicle is designated as A. In this run the vehicle starts from the bottom left corner of the test arena and moves to the right, then half way through the map it switches to the top section of the map. The vehicle has a constant surge velocity with a set of angular rotation to choose from to generated the desired trajectory.

Test type-B : This type of tests use the same arena as test type-A yet the vehicle in this case follows trajectory B, as shown in Figure 5.9. For this run the trajectory is generated manually using joystick by driving the vehicle next to the wall throughout the full mission. The use of joystick makes the trajectory more rough with variable velocities in forward later direction as well as variable angular rotation.

Test type-C : The arena and the corresponding trajectory for this test type are shown in Figure 5.10. This type of tests are added to provide a slightly different scenario for validating system parameters. The addition of pillar like structures and grooves is intended to simulate real life pier environments.

Test type-D : This is by far the largest arena for testing the performance of attentive navigation in long missions. The test arena, shown in Figure 5.8, is built based on the Fort William underwater centre pier. The structure is composed of a set of pillars

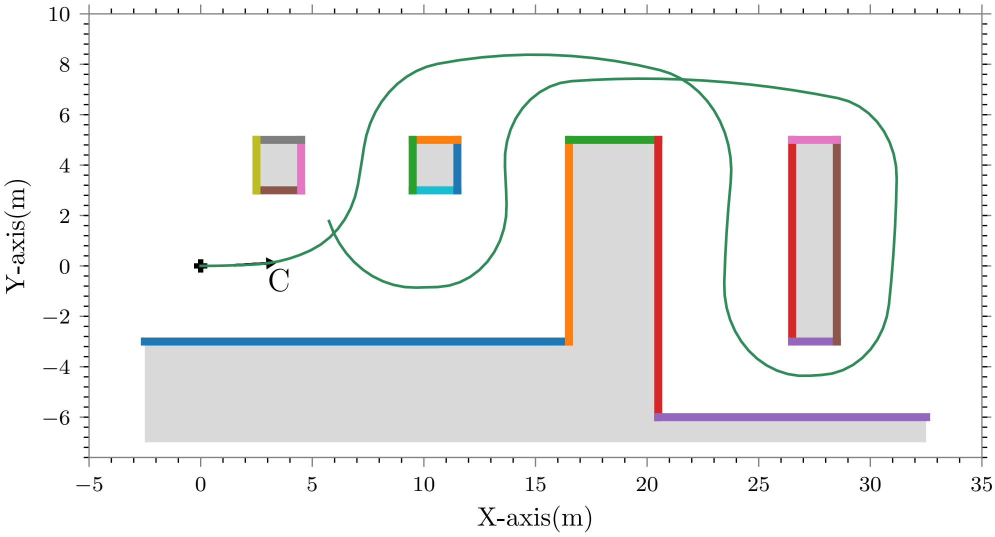


Figure 5.10: Test type-C map of simulated pier environment and trajectory. Thick lines indicate pillar and harbour boundaries. The vehicle path is shown as trajectory C, which starts from the map origin on the left then goes all the way to the right and returns to a point 6 m off the origin.

and cross beams connecting these pillars. In this test type the vehicle makes $300\,\mathrm{m}$ trajectory, starting from the left side of the structure and finishing at a nearby point after visiting significant portion of the pier.

In all of these simulated tests, the indicated trajectories are used as a ground truth. For each test type a noisy version of their corresponding trajectory is generated by adding zero mean velocity noises with standard deviation given by $10\,\%$ of the maximum velocity in each direction. The rate of pose update is $2\,\mathrm{hz}$ except for test type-A which runs at $1\,\mathrm{hz}$, which is in synch with sonar measurement while the other get sonar measurement at every other control update. The sonar sensor is simulated using the procedure discussed in Section 3.1.3 where the line segment deviations in ρ and θ are set as $\sigma_{\rho\rho}$=2 cm and $\sigma_{\theta\theta} = 1.0°$. Next the result from simulated experiments is presented where some of these tests are intended to tune parameters of attentive SLAM.

Tuning grid spacing g_s

In order to determine the effect of changing grid spacing an attentive SLAM test is set-up by varying the grid spacing for each run. During these runs, all the other parameters, including the odometry noise per batch, are kept constant. Individual runs are compared using the mean X-Y RMSE position error for EKF-SLAM and UKF-SLAM. This process is repeated

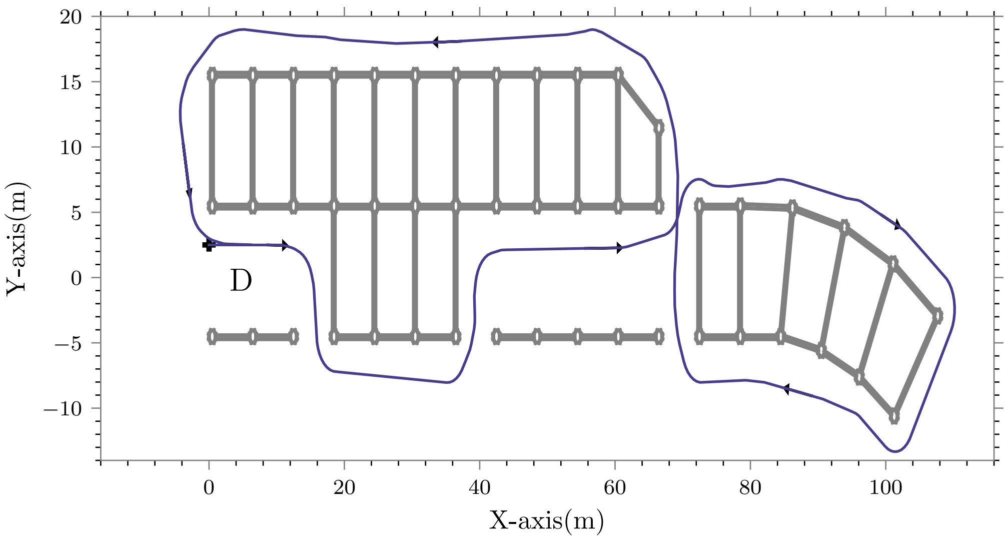
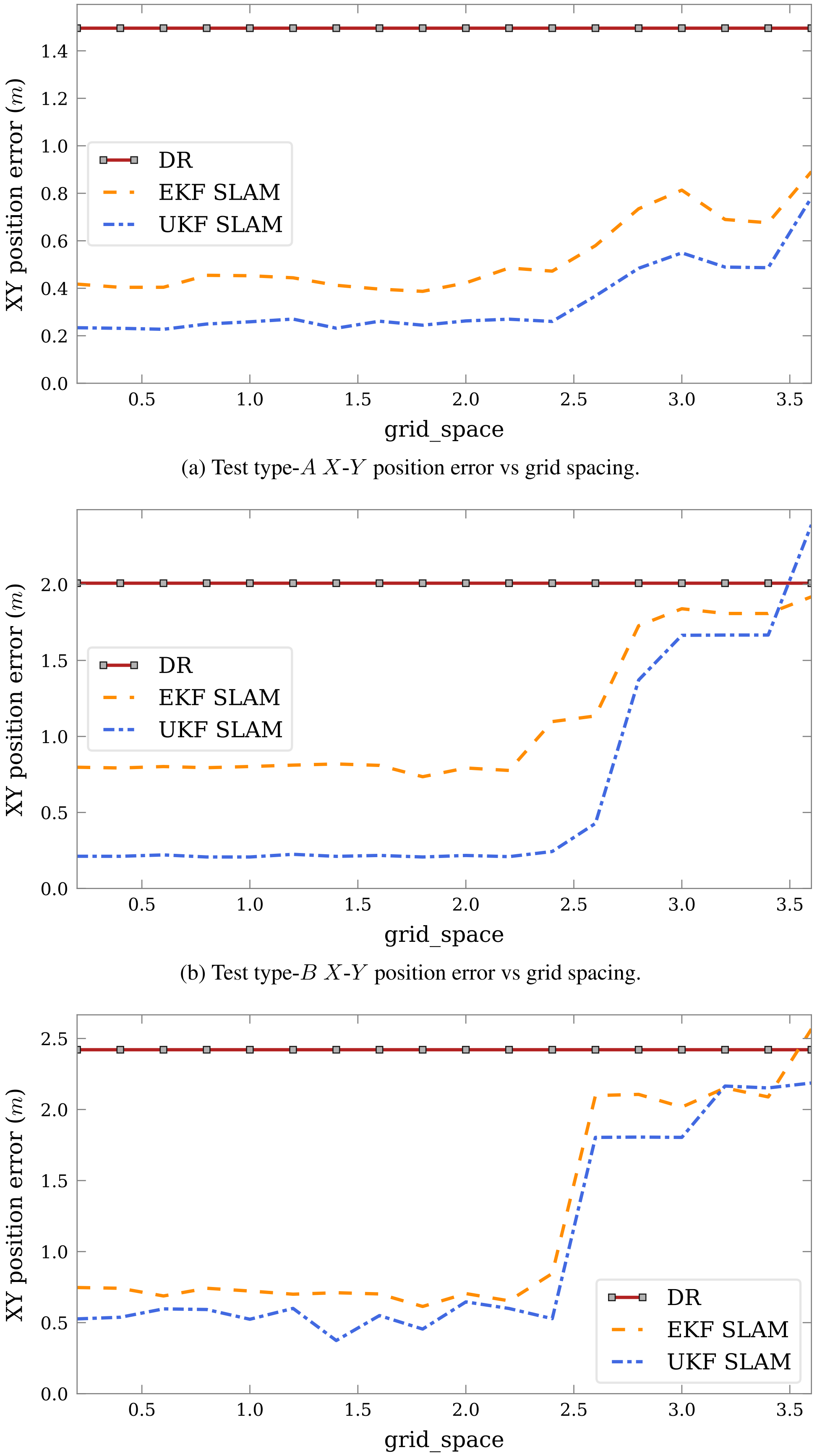


Figure 5.11: Test type-D map of simulated Fort William pier and AUV trajectory. Thick grey lines indicate pillars and beams while the vehicle path is shown by trajectory D. The map origin is assumed to be the starting position, then the vehicle moves to the right making an intertwined loops covering a total of $300\,\text{m}$ before coming back to the starting area.

ten times to get the mean position error for each grid spacing as show in Figure 5.12. Figure 5.12 shows the result for tests runs using test type-A, type-B and type-C. A first look at the results in the figure indicates that a smaller grid spacing lowers the mean RMSE position error. However, there is a minimum grid spacing below which the change in grid spacing is less significant. In this particular case the limit is $2.4\,\text{m}$ for a sonar sensor with maximum range of $4\,\text{m}$, below which the error is roughly constant with occasional peaks. The lower limit of this result is set by hardware constraint, the pan and tilt unit has a minimum resolution of $2\,^{\circ}$, finer grid spacing will not introduce any new possible sensor orientation. Occasional dips in the error, especially between $g_s = 1.0\,\text{m}$ and $g_s = 2.4\,\text{m}$ for test type-C (see Figure 5.12 (c)), this may indicate the presence of important sensor orientations throughout the mission that can be introduced through the use of certain grid spacing. Note that these fluctuations are less apparent for test type-A and type-B which share the same test arena. The best grid spacing can only be identified based on actual test environment. In real sonar experiment grid spacings which result in a minimum mean position error are identified by manually tuning the grid spacing in a simulated environment.

Generally, when grid spacing is too high it results in a significant increase on the mean position error, which in extreme case accompanied by filter divergence as shown in Figure 5.12 (b) UKF-SLAM or Figure 5.12 (c) EKF-SLAM both at $g_s = 3.6\,m$. For such



(a) Test type-A X-Y position error vs grid spacing.

(b) Test type-B X-Y position error vs grid spacing.

(c) Test type-C X-Y position error vs grid spacing.

Figure 5.12: Attentive SLAM X-Y position error for variable grid spacing using three different simulation set-ups.

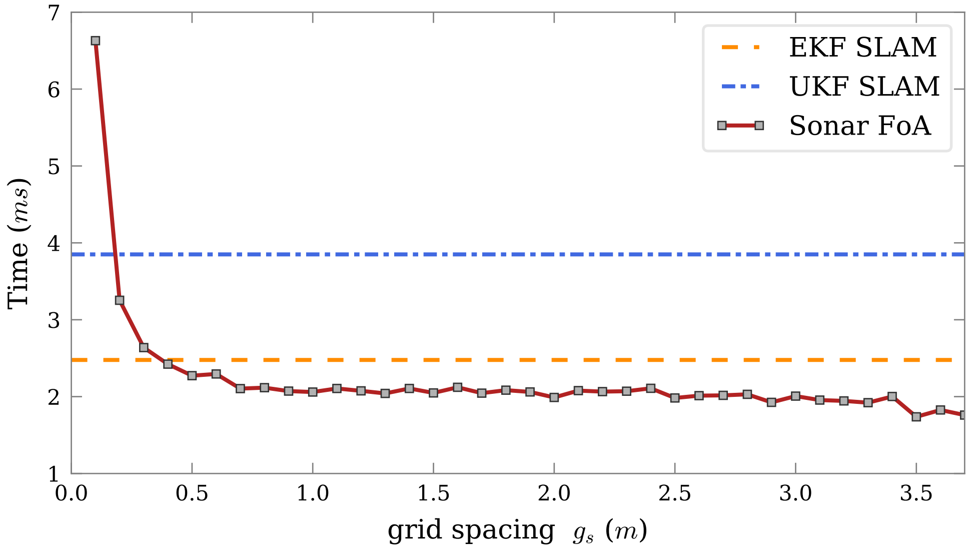


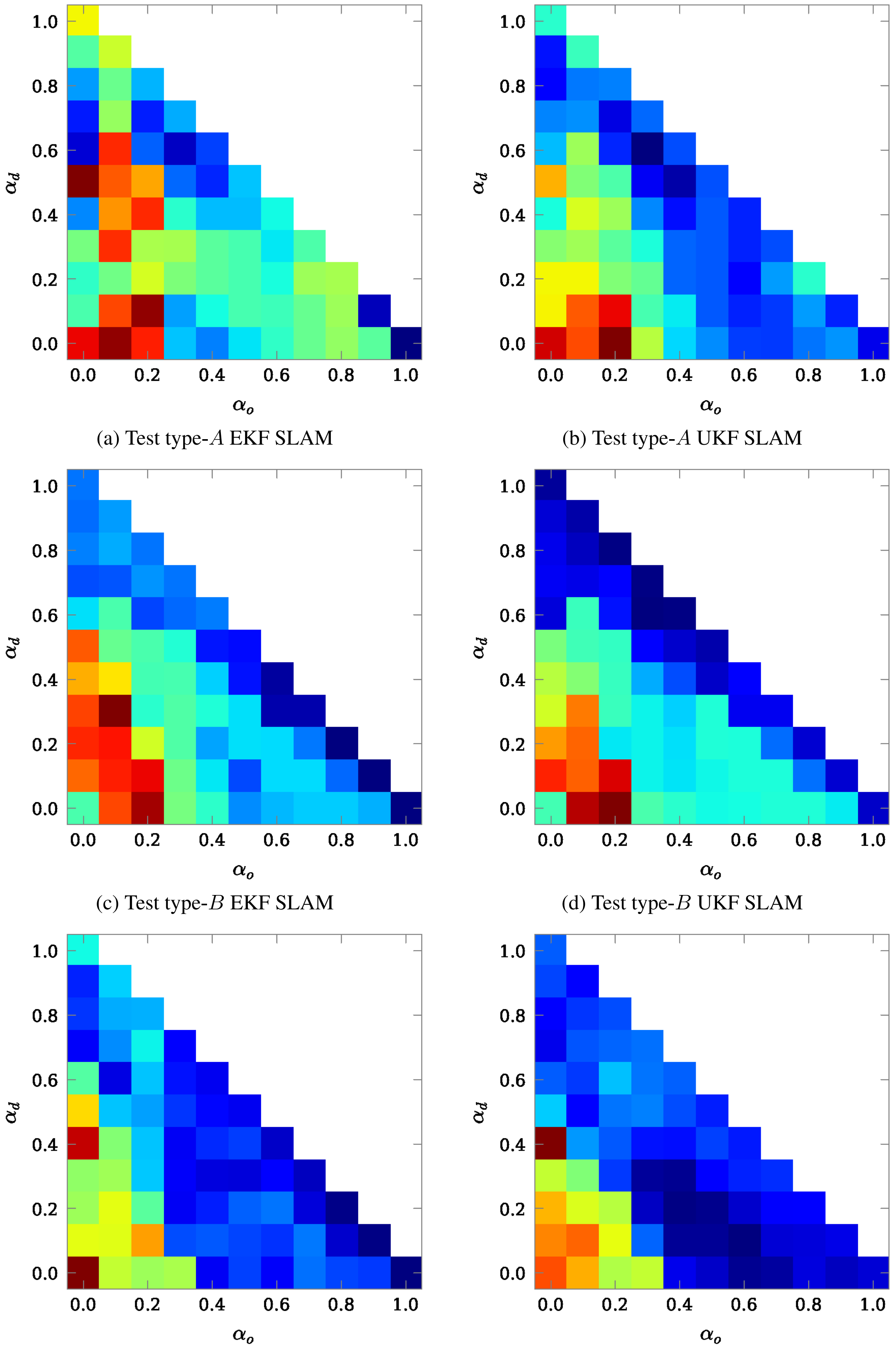
Figure 5.13: Effect of changing grid spacing g_s on the computational cost of sonar focus of attention. The average single step SLAM computational time is also included for EKF and UKF based implementations.

high grid spacing the number of salience points around the vehicle is limited which intern narrows the choice of orientation for sonar image acquisition. This problem is exacerbated when some of these limited grid points lie outside the vicinity of the surrounding landmark features.

Another important factor to consider when choosing grid spacing is the computational cost. The relationship between the number of grid points and grid spacing is governed by inverse square law, *i.e.* $no.\ grids \propto {}^{1}/_{g_s^2}$ (excluding those unreachable grid points within a square area). Hence a lower grid space means an exponential increase on the computational cost of attention weight estimation. Figure 5.13 show this relationship between the computational cost of computing sonar attention weights and the size of grid spacing. For a reference the average SLAM prediction and update time is given for both EKF and UKF filter. Accordingly, a single pair of SLAM prediction and update takes about 2.5 ms for EKF and 3.8 ms for UKF, excluding feature extraction. The corresponding computational cost for computing sonar salience varies between 2.0 ms and 6.6 ms for a grid spacing g_s of 1.0 m and 0.1 m respectively. In conclusion, the grid spacing has to be small enough to provide enough detail yet large enough to avoid unnecessary computational cost without compromising the upper limit to obtain enough salience points.

Tuning scale parameters (α_a, α_o , α_d)

The other import parameters in the proposed attentive SLAM technique are the scale factors used for combining the three types of attention weights (see Equation 5.4). In order to determine the behaviour of the system with these parameters a brute force approach is employed where each type of scale is varied between 0 and 1 with a step of 0.1. Then for each valid combination of scales (*i.e.*the sum of the three scale has to be one), a SLAM test is executed using test type-A, B, and C. Figures through 5.14 to 5.17 show results from these experiments. In Figure 5.14 EKF and UKF-SLAM position errors are displayed against α_o and α_d which is intended to identify regions in the parameter space that can lead to a reduced position error. One thing that sticks out among all the test types is the fact that a pure angle of incidence weight based salience results in poor SLAM performance. This can be explained looking back on how the angle of incidence constraint weight defined in Section 5.1.2, which does not account the actual overlap between the line segment and the sonar measurement. Hence, lines with an optimal orientation will result in a higher angle weight even if the segment might be out of the sonar field of view. On the contrary, the other two weights take proximity of the grid point to the segment in to account as a result the position error for $\alpha_o = 1$ or $\alpha_d = 1$ is relatively small. Finding the best combination of scale parameters is a difficult task which depends on the type of route and environment involved, yet safe zones can be suggested based on the result. Generally, a smaller angle of incidence scale ($\alpha_a \approx 0.1$) and similar overlap and distance scales results in acceptable level of mean RMSE SLAM position error. Individual relationship between scale parameter and position error are presented in Figure 5.15-5.17. These results also include special SFoA runs where the final attention orientation is determined by the mean of n-past attention orientations. This provides a smooth transition from one orientation to the other reducing the load on the pan and tilt unit as well as minimizing unaccounted vehicle dynamics disturbance. Including a smoothing factor in simulation mostly results in a slightly higher position error, unless this procedure happen to avoid any oscillation in attention angle for a particular test run. However, this rise is less significant compared to how much an attentive navigation improve DR navigation system. Additionally, the individual parameter test confirms the recommendation for smaller angle of incidence weight scale.



(a) Test type-A EKF SLAM

(b) Test type-A UKF SLAM

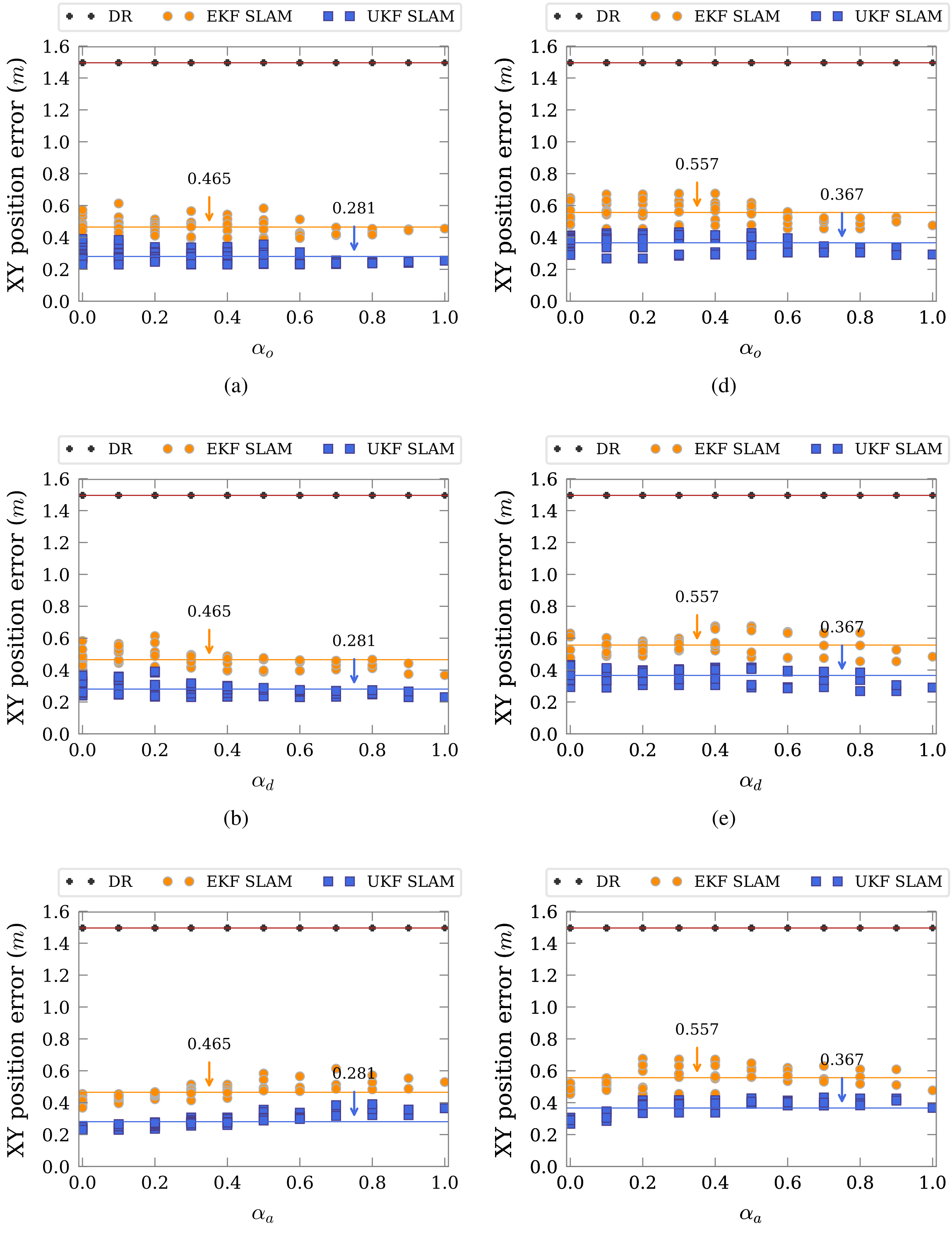
(c) Test type-B EKF SLAM

(d) Test type-B UKF SLAM

(e) Test type-C EKF SLAM

(f) Test type-C UKF SLAM

Figure 5.14: Tuning salience scaling parameters using EKF and UKF SLAM position errors. Axis in each image correspond to the scale α_o and α_d, while α_a runs diagonally where it is maximum at the origin; Colour represent position error (dark blue:min - brown:max).



(a)

(d)

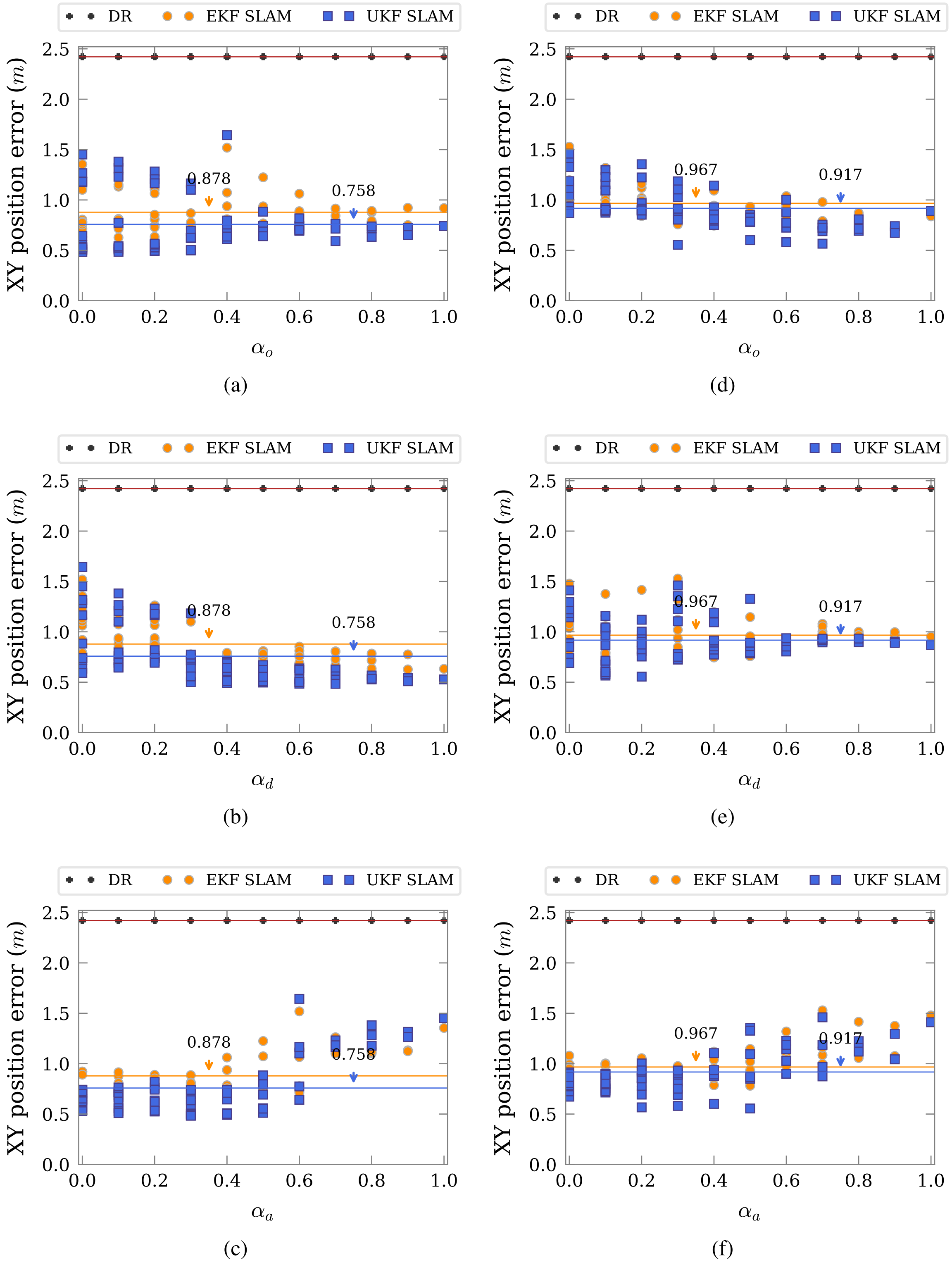
(b)

(e)

(c)

(f)

Figure 5.15: Test type-A relationship between individual attention scale parameters and SLAM X-Y position error. (a-c) show test results for α_o , α_d and α_a respectively with just IoLR. (d-f) show similar experiment with temporally smoothed attention orientations which is given by the mean of the last five attended angles.



(a)

(d)

(b)

(e)

(c)

(f)

Figure 5.16: Test type-B relationship between attention scale parameters (α_o , α_d and α_a) and SLAM position error. (a-c) result without temporal smoothing of attention angles, while (d-f) show the effect of using temporally smoothed attention orientation with a window size of five steps.

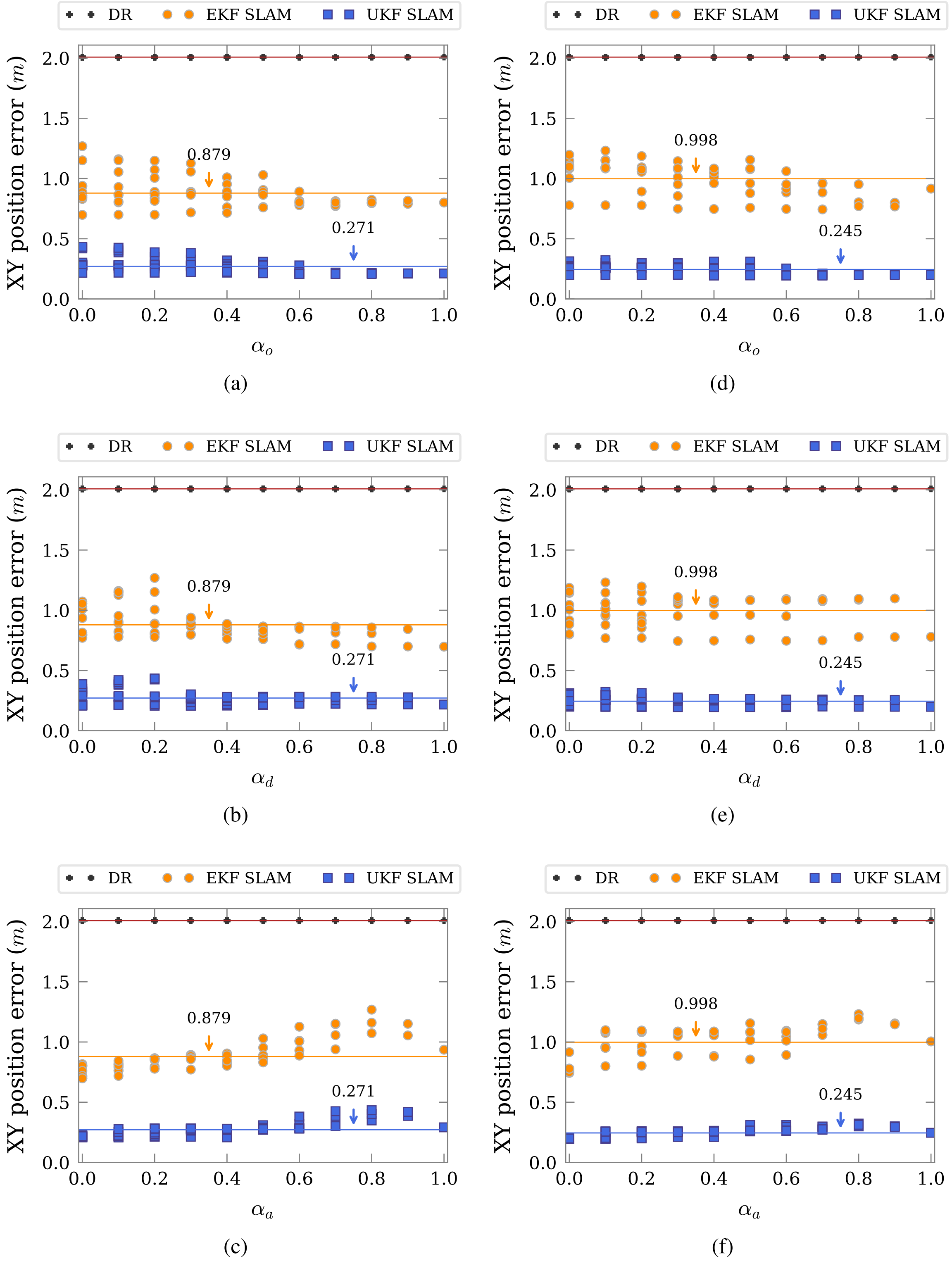


Figure 5.17: Test type-C relationship between attention scale parameters (α_o , α_d, and α_a) and SLAM position error. (a-c) result without temporal smoothing of attention angles, while (d-f) show the effect of using temporally smoothed attention orientation with a window size of five steps.

Fort William pier simulation

Navigation challenges faced in a small confined environments do not fully account problems faced in real life long mission navigations. To successfully address long mission challenges a final simulation test is done using a mock-up of Fort William pier as shown in Figure 5.11. The resulting position errors using a passive SLAM navigation are presented in Figure 5.18. In this test the sonar sensor is mounted looking forward in the direction of motion. The result demonstrates that even through a passive SLAM can reduce the overall navigation drift, especially with a more non-linear UKF-SLAM, for much of the first section of the trajectory the SLAM errors are comparable to that of DR position error. On the other hand, Figure 5.20 shows a position error for a test using an attentive SLAM technique where the sonar heading at each measurement pose is determined based on the output of an SFoA system. This will cause the sonar to point upward in the first section of the trajectory where there is a cross beam to provide a sonar feature. As a result the position error for both types of SLAM is lower than their passive equivalent.

From Figure 5.18 and Figure 5.20 it can be seen around 362 sec the dead reckoning error is much better than the SLAM error. At this time of the mission the vehicle is around $x = 68.8$ m and $y = -1.94$ m. This position is magnified in the corresponding trajectories shown in Figure 5.19 and Figure 5.21. The dead reckoning path momentarily get very close to the ground truth X-Y position even thought there exist a significant drift in the corresponding heading which ultimately cause the trajectory to deviate. This is merely due to the shape of the test trajectory and sequence of trajectory errors for the particular run.

Table 5.1 summarise the comparison between passive and active SLAM navigation using Fort William pier simulation. Based on the result, the improvement introduced by using a SFoA is independent of the type of SLAM filter yet an attentive UKF-SLAM provide the lowest mean X-Y position error (1.06 m down from 9.82 m for DR) and the smallest maximum position error (2.18 m down from 26.14 m for DR).

5.4.2 OSL tank experiment

This test was carried out in OSL test facility (see Section 1.2). A BlueView MB2250 sonar is mounted on the Cartesian robot using a pan and tilt unit. Though the Cartesian robot has 3-DoF for this test only the X and Y axis are used yet the vehicle model is kept as it was

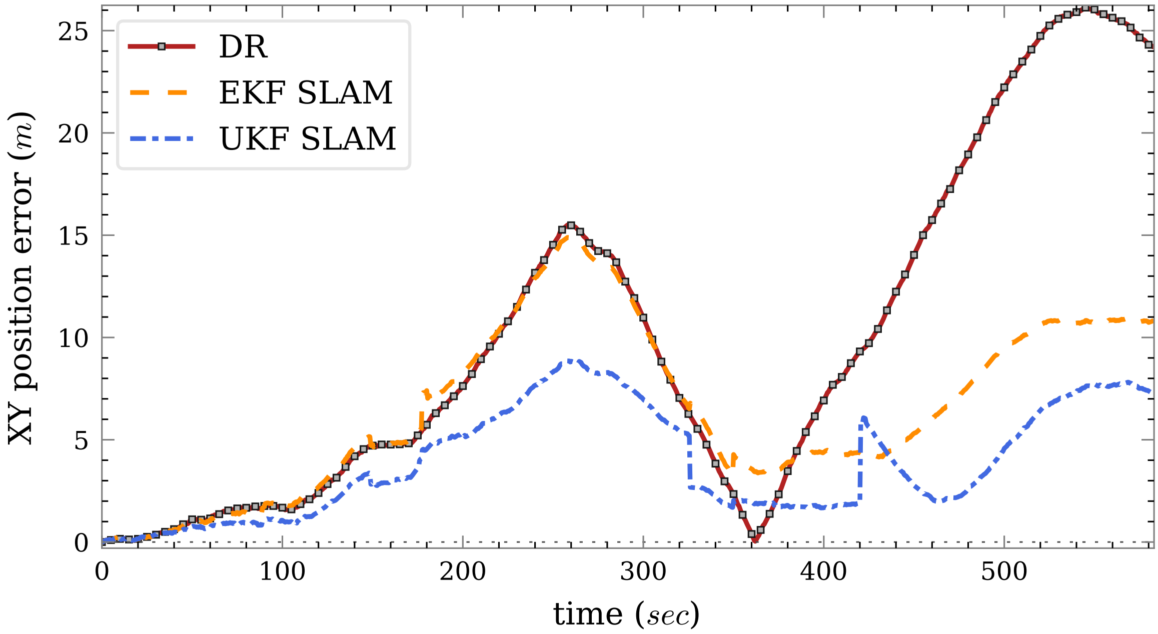


Figure 5.18: Fort William pier simulated SLAM test X-Y position error for DR as well as EKF and UKF SLAM with the sonar mounted facing the direction of motion.

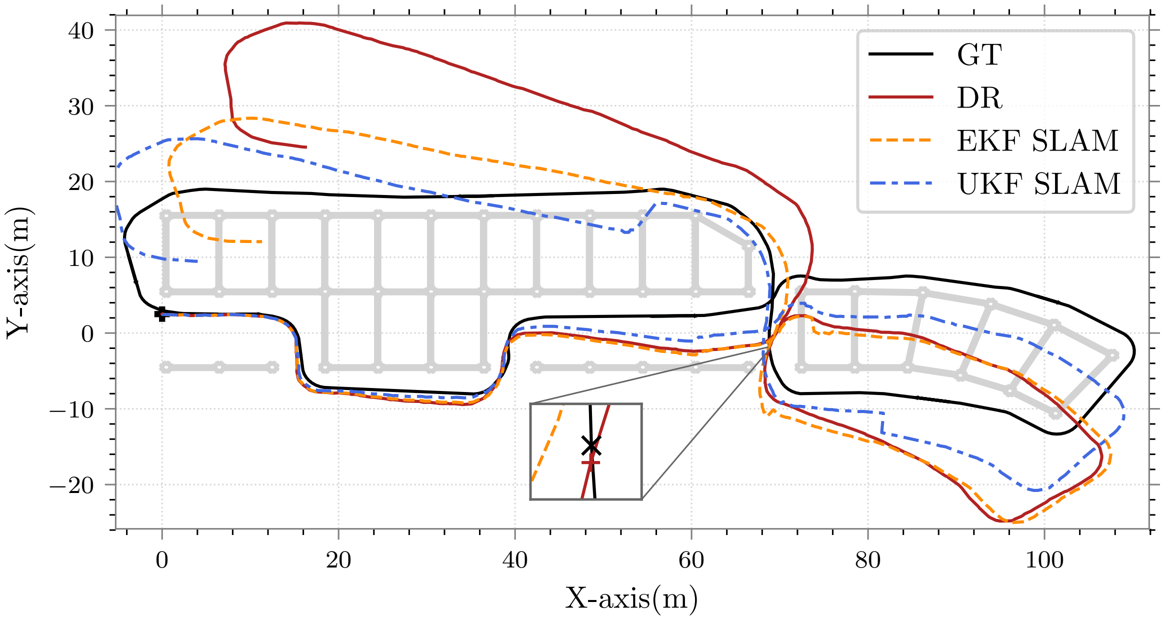


Figure 5.19: Fort William pier simulated passive SLAM test vehicle trajectory and the corresponding ground truth and dead reckoning path. The markers with in the magnified section indicate the vehicle position at $t = 362\,\text{sec}$.

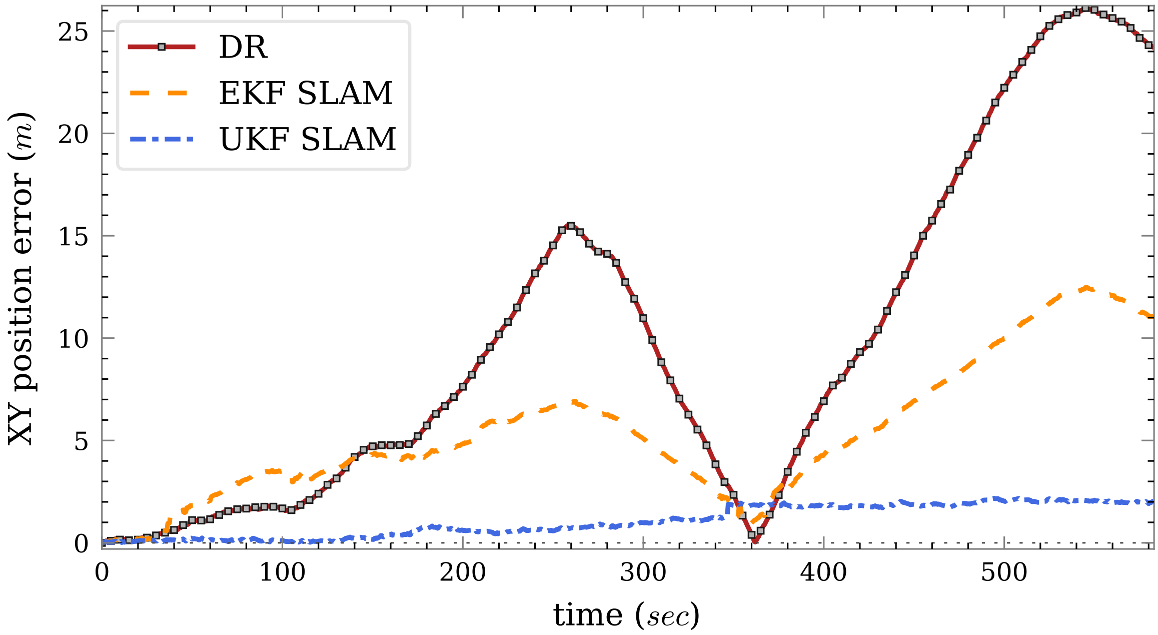


Figure 5.20: Fort William pier simulated SLAM test where a SFoA is used to determine sonar heading before every image acquisitions. X-Y position errors are shown for DR and SLAM navigations using EKF and UKF filter.

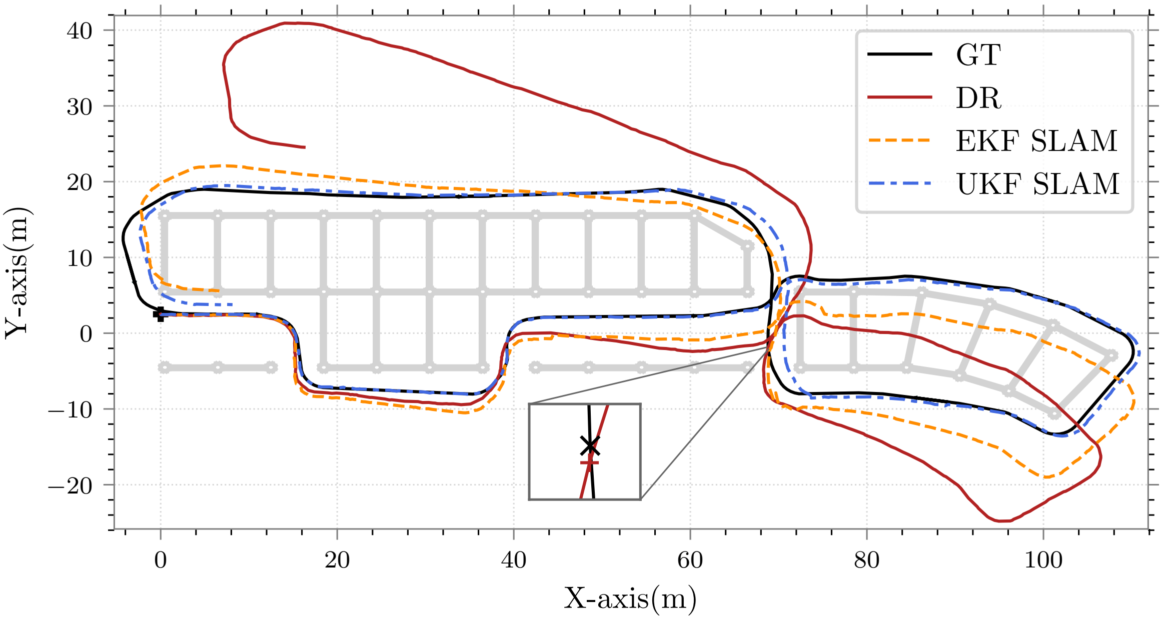


Figure 5.21: Fort William pier simulated FoA based SLAM test trajectories for EKF-SLAM, UKF-SLAM, dead reckoning and ground truth. The markers with in the magnified section indicate the vehicle position at $t = 362\,\text{sec}$.

Sensor	mean			STD			max		
	DR	EKF	UKF	DR	EKF	UKF	DR	EKF	UKF
Passive	9.82	6.49	3.90	8.19	4.11	2.67	26.14	14.89	8.86
Active	9.82	5.37	1.06	8.19	3.30	0.77	26.14	12.49	2.18

Table 5.1: Fort William pier simulation experiment comparison between passive and attentive SLAM using EKF and UKF filters. The mean, STD and maximum X-Y position errors are shown in m.

discussed in Chapter 4. The pan and tilt unit is restricted with in the horizontal axis where it can be steered left and right without changing the vertical alignment. The sonar sensor is set to have 77° FoV and a maximum range of 4 m. The sonar sensor is publishing images every 2 sec while the Cartesian robot provide a ground truth at the rate of 10 hz. For this test three types of motion trajectories are devised which results in tests type-I, type-II and type-III.

Test type-I : Figure 5.22a shows the trajectory for this test type, where the robot moves from a corner to opposite corner then shifts to the next corner and repeat the process. The challenge in this type of trajectory arises when the robot makes those long diagonal moves. In such circumstance features might not be available in the direction of motion hence typical sensor configuration is bound to fail.

Test type-II : Lawn-mower pattern is a very common exploration trajectory in most robotic applications. Figure 5.22b shows such type of mission set-up where the vehicle makes three small loops. The longest sides walls are reachable from most part of the trajectory but a better landmark grip can be obtained by incorporating the two other ends of the tank.

Test type-III : Figure 5.22c shows a vehicle trajectory where the robot is always next to a wall. These type of trajectories are common in a mission involving visual servoing. In these applications the vehicle is initially pre-programmed to face the direction of feature which does not account any other robotic requirement *i.e.* ease of movement or change in the environment. This test is intended to show the capability of attentive navigation to fill these shortcomings.

The starting position in all the three types of motion trajectories is indicated as O which is also the map origin. The section of the trajectory starting from point A is repeated twice for all test types. These trajectories are then loaded to the robot as a sequence of two linear velocities in the X and Y direction. A noisy version of these velocities are used as odometry input with a linear and angular velocity noises having standard deviation of 0.005 m/s and 0.02 DEG/sec respectively. The simplicity of the motion model together with the smallness of the confined test environment limits the non-linearity of the system, hence the test results will be presented using a UKF filter based SLAM as changing the filter does not have any effect.

With regards to sensor configuration five different set-ups are considered, out of which two are based on SFoA. These sensor configurations are:

P1 : passive sensor configuration where the sonar is always looking in the direction of positive X-axis (relative to the map frame).

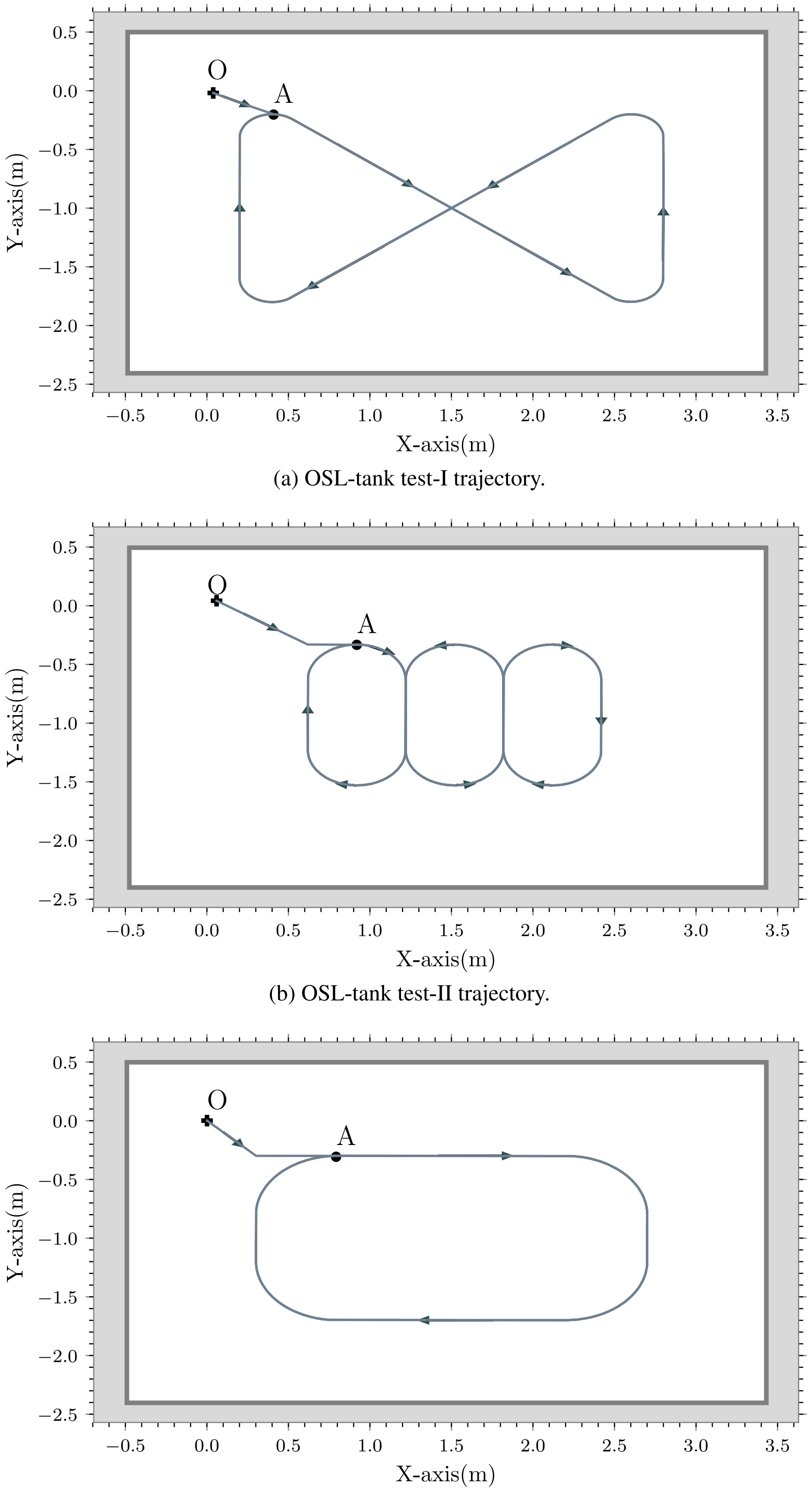
P2 : passive sensor set-up where the sensor is always pointing in the direction of motion. This is the most common sensor configuration when using forward-looking sonar.

P3 : this set-up is an intermediate between passive and active sensor set-up where the sonar sensor continuously moves right and left where the speed of rotation is set as 10 DEG/sec. There is no synchronization between the landmark feature appearance and sensor orientation hence labelled as passive.

A1 : this one is an active attentive sensor configuration where the sonar is directed towards the instantaneous maximum attention grid point.

A2 : this configuration provide an active sensing with inhibition of large rotation (IoLR) (see Section 5.2.1) which helps to avoid abrupt sensor movement as well as unnecessary brief oscillations due to landmarks that disappear before the sensor makes full rotation.

Next for each test type a SLAM run is executed using the five possible sensor configurations. The performance of a UKF-SLAM algorithm in this five sensor configurations is compared in each test set-up. In order to increase the validity of this comparison the odometry noise sequence is kept the same for all possible sensor configurations of a given test run. For instance, the result shown in Figure 5.23-5.25 are obtained using the same sequence of velocity noises where the trajectory is that of test type-I. Figure 5.23 shows the resulting UKF-SLAM trajectory and map for the three passive sensor configurations (P1,



(a) OSL-tank test-I trajectory.

(b) OSL-tank test-II trajectory.

(c) OSL-tank test-III trajectory.

Figure 5.22: OSL-tank test trajectories. Vehicle start position is indicated as O where the section starting from A is repeated twice for the mission.

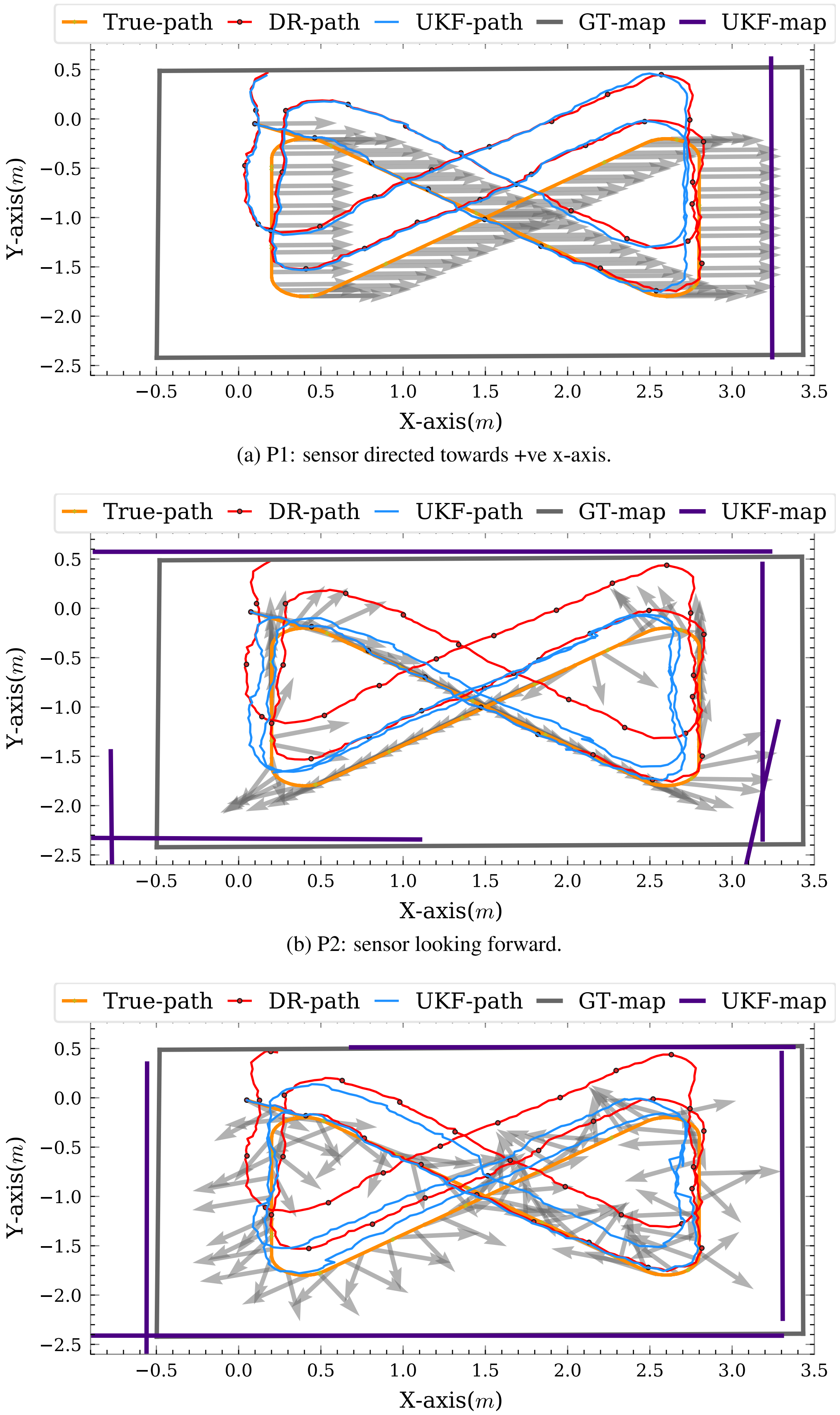
P2, P3) for test type-I while that of the two active sensor configuration (A1, A2) is shown in Figure 5.24. Although the odometry noise sequence for all the five tests is kept similar, the measurements differ as a result of the difference in sensor orientations. Each of these figures also display the expected DR trajectory as well as the ground truth trajectory and map. In order to generate the ground truth map the true position of the robot is used to transform all the sonar measurements into the map frame from this the four lines are determined using k-mean clustering algorithm.

For test type-I, P1 is the most restrictive sensor configuration where the vehicle is only able to pick measurement when it is close to the right hand side wall. As a result the UKF-SLAM trajectory for this run is barely differ from that of a DR trajectory. However the result for P2 and P3 is significantly better where for most part of the trajectory the vehicle manages to detect sides of the tank. The SLAM run for the forward looking sensor configuration, P2, has especially managed to provide a better trajectory in the second loop of the trajectory. This is actually expected as the vehicle for this test type is heading in the direction of corners for most part of the trajectory. When it comes to sensor configuration P3, the possibility of getting a good measurement in a run depends on relationship between sections of the environment and order of rotations where a good synchronization results in a reduced SLAM position error. On the other hand, attentive sensor configurations A1 and A2 result in a better SLAM runs using landmarks from various corner points of the arena at different stages of the trajectory. Although for most part of the trajectory the sensor orientation in A1 and A2 are similar, there is a noticeable difference at the top left corner of the tank. At this corner, as the robot comes up and turn right, the configuration A2 initially restrict the sensor from making a full rotation to direct the sonar towards the corner, even though ultimately it makes the rotation as a result of persistent request. It is important to see the effect of such change through a comparison of SLAM position error estimate. Additionally, although the five trajectory plots can show individual performance and corresponding sensor orientations, it is difficult to make comparisons among the five sensor configurations. Hence Figure 5.25 shows the behaviour of the UKF-SLAM position error through time for the five possible sensor configurations. As expected, the SLAM error for P1 sensor configuration is in general divergent while that of all the other sensor configuration shows a reduction in the later section of the trajectory due to a possible loop closure through revisiting landmarks. The error in attentive SLAM navigation is generally lower than that of passive configuration. However P2 sensor configuration performing as good in the last section of the trajectory. This is mainly due to the synchronization that appear to happen between the sensor rotation and part of the tank for that section. This synchronization can be lost in the long run or with the change on the type of trajectory as it will be seen for test type-III later.

A similar sequence of experimental results for test type-II are shown in Figure 5.26-5.28. As in the previous test, the trajectory for sensor configuration P1 is very close to the DR trajectory as shown in Figure 5.26 a where the sensor is always pointing in the direction of the right-hand side wall. In Figure 5.26 b the result for P2 sensor configuration is shown. Although the sensor is intended to mimic forward looking sonar configuration, the hardware limitations make it difficult to follow the sudden changes in the direction of motion through the trajectory. For most part of the trajectory the sensor is looking inward instead of the wall as result the trajectory is significantly off from the ground truth. The result for configuration P3 is relatively good where the trajectories for the first and the second half are overlapping as a result of loop closure. Additionally, the sensor is getting enough sonar acquisitions of the walls even though the same cannot be said regarding the corners.

Attentive sensor configurations appear to direct the sonar in the direction of corners whenever possible as shown in Figure 5.27. Additionally for these two sensor configurations the amount of sonar measurement is higher, this is reflected on the coverage of the final SLAM map. A direct comparison of the performance in all five sensor configuration is presented in Figure 5.28 in the form of position error throughout the full period of mission. The SLAM result for P1 is very close to the DR navigation error as in case of test type-I. Among the rest, sensor configuration P2 resulted in a distinctively lower SLAM performance. This is due the reduction in the number of landmark sighting as depicted by the final SLAM map in Figure 5.26 b. The choice among the other three type of sensor configuration is slightly vague for this test type.

Lastly, the performance of the five sensor configurations is compared using test type-III as shown in Figure 5.29, Figure 5.30 and Figure 5.31. As in previous test, the sensor in P1 configuration only managed to pick up the right hand side wall which leads to a poor SLAM performance. In case of P2, even though the sensor managed to get a better sighting of the

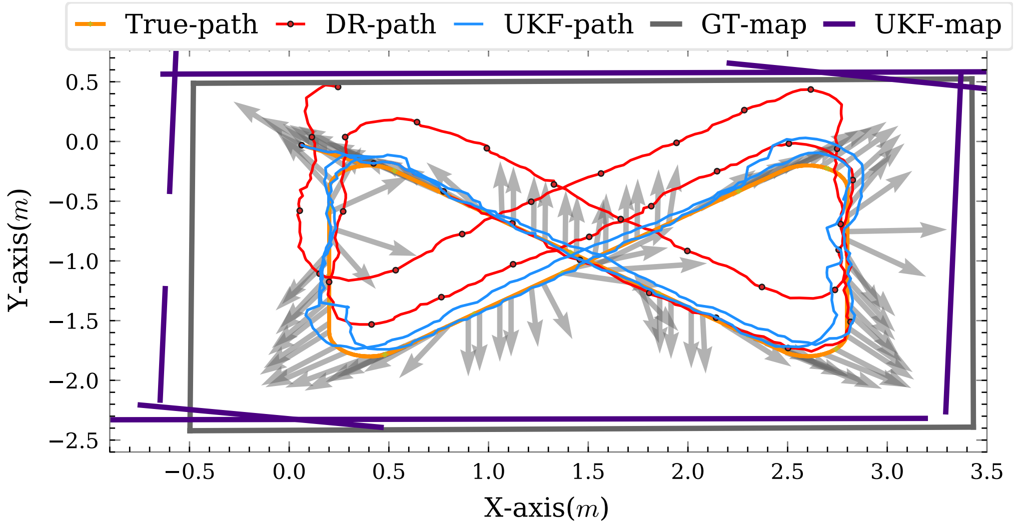


(a) P1: sensor directed towards +ve x-axis.

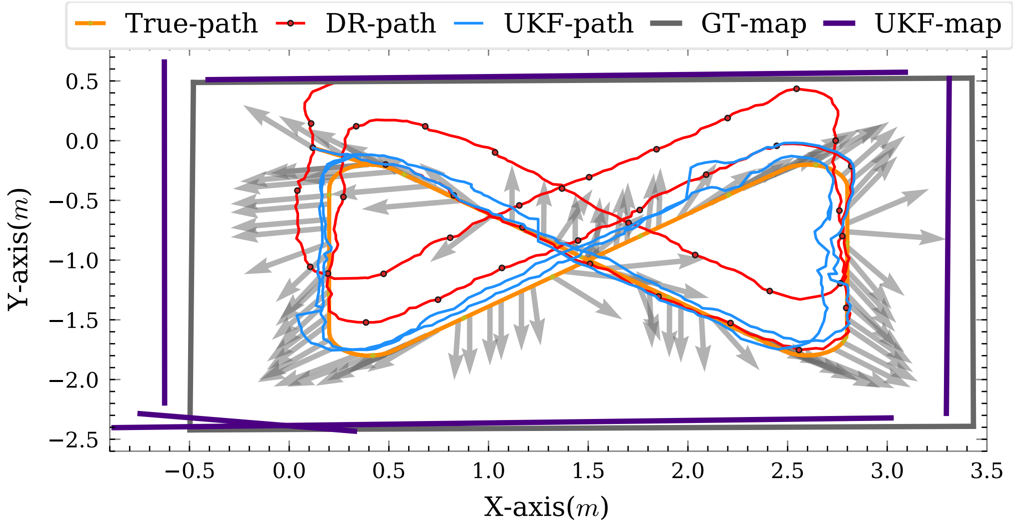
(b) P2: sensor looking forward.

(c) P3: sensor rotating continuously.

Figure 5.23: OSL tank test type-I SLAM trajectories for passive sensor configurations.



(a) A2: attentive sensor configuration using instantaneous value.



(b) A2: attentive sensor configuration with IoLR.

Figure 5.24: OSL tank test type-I SLAM trajectories for attentive sensor configurations.

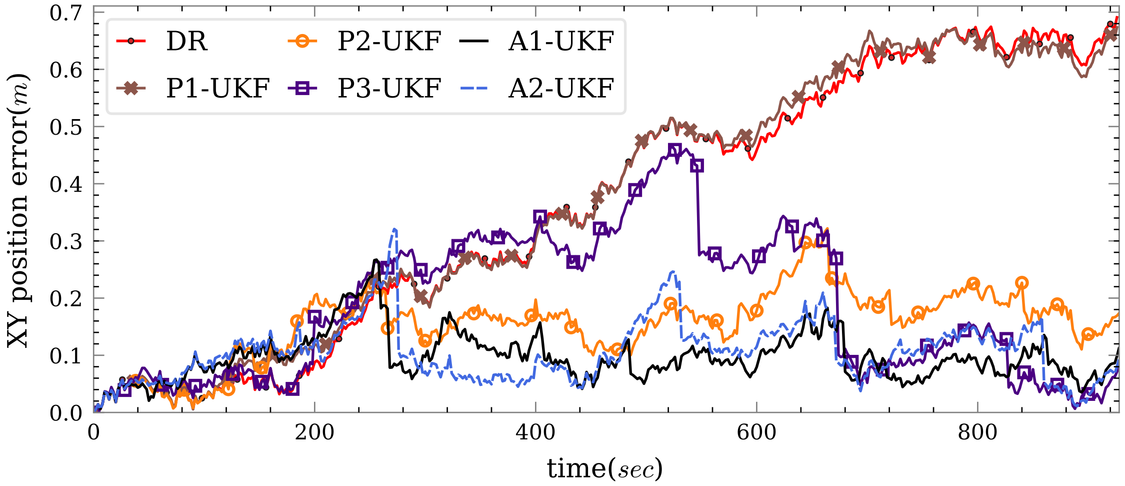
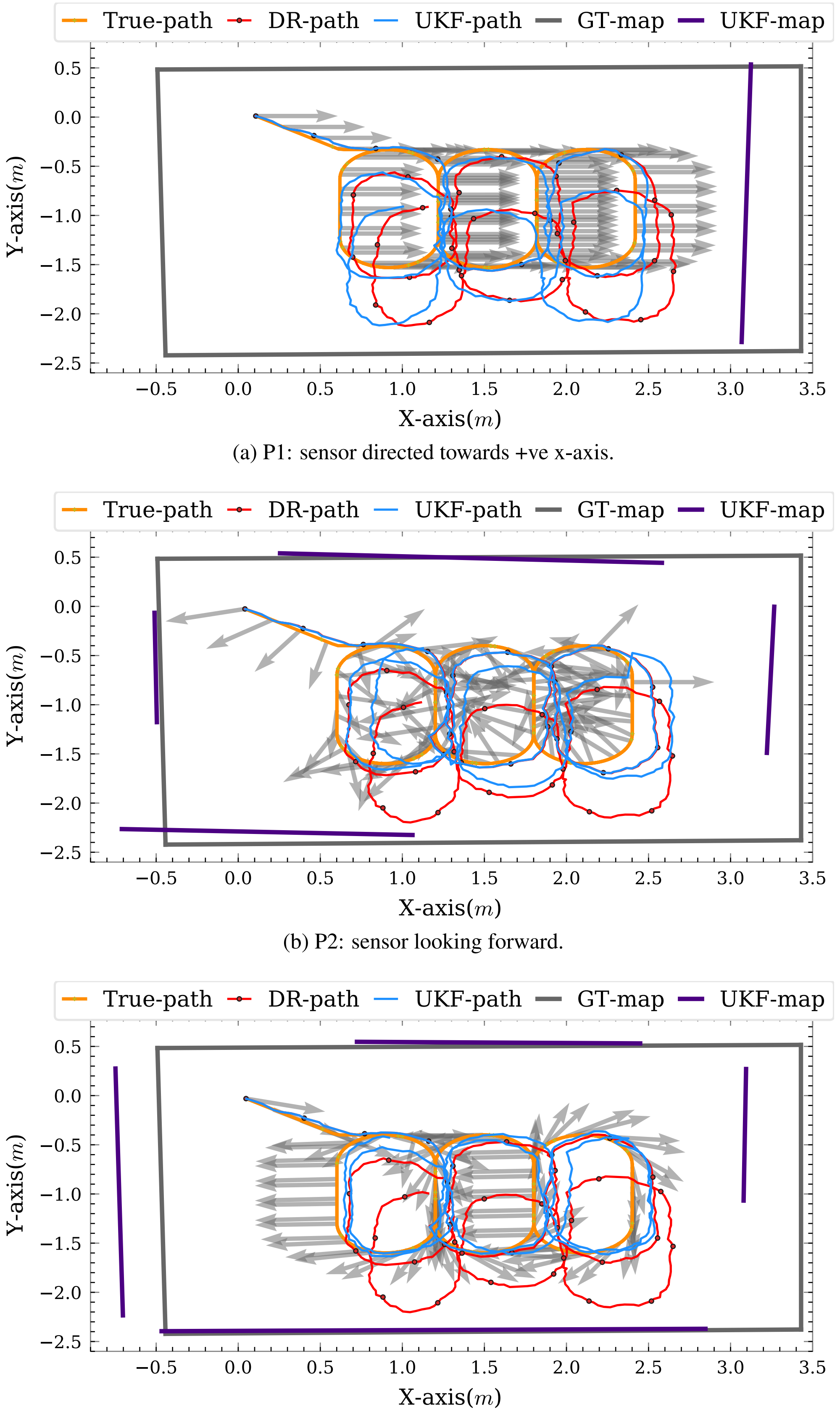


Figure 5.25: OSL tank test type-I UKF-SLAM XY position error for different sensor configurations.

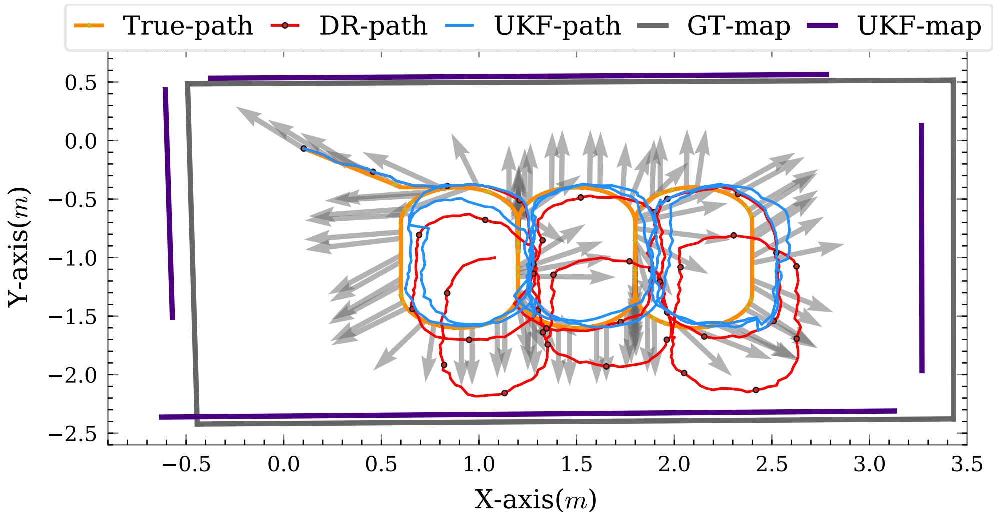


(a) P1: sensor directed towards +ve x-axis.

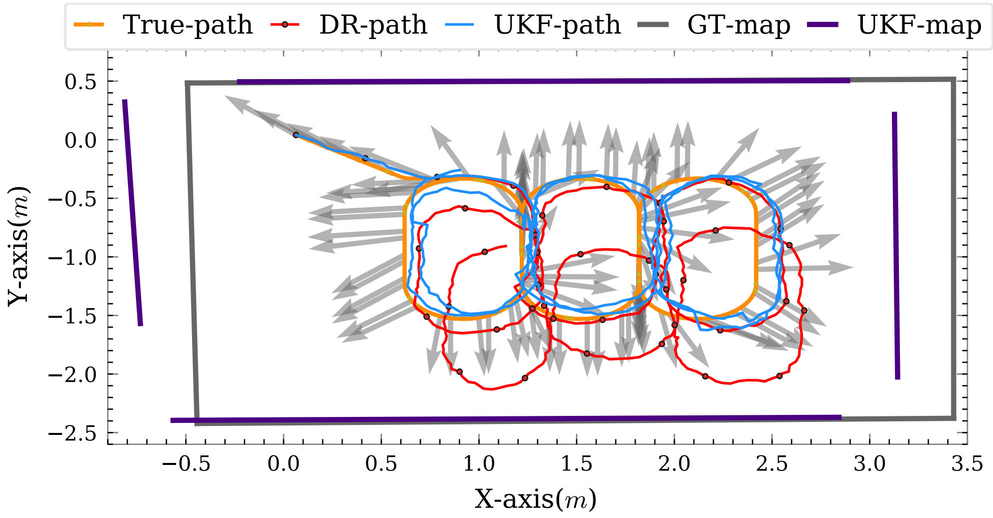
(b) P2: sensor looking forward.

(c) P3: sensor rotating continuously.

Figure 5.26: OSL tank test type-II SLAM trajectories for passive sensor configurations.



(a) A2: attentive sensor configuration using instantaneous value.



(b) A2: attentive sensor configuration with IoLR.

Figure 5.27: OSL tank test type-II SLAM trajectories for attentive sensor configurations.

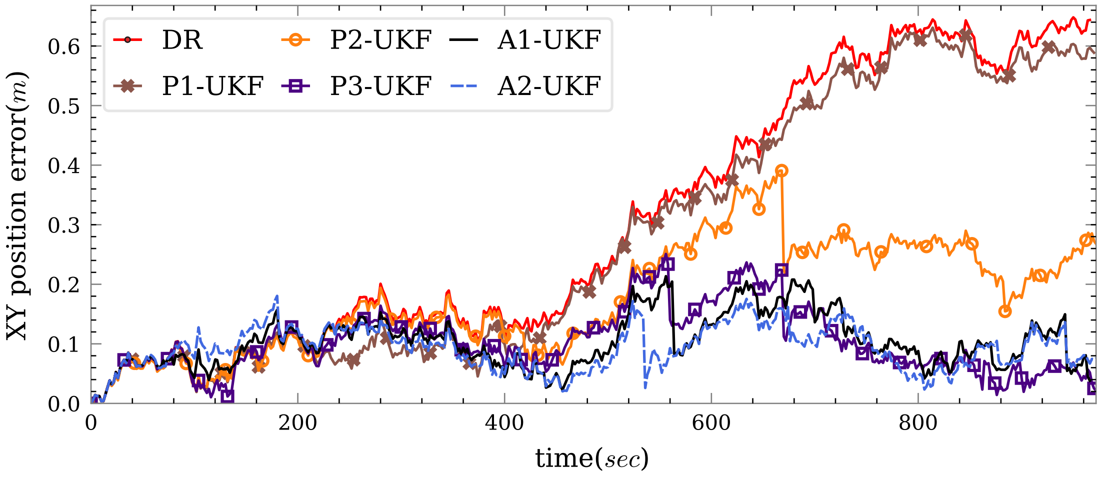
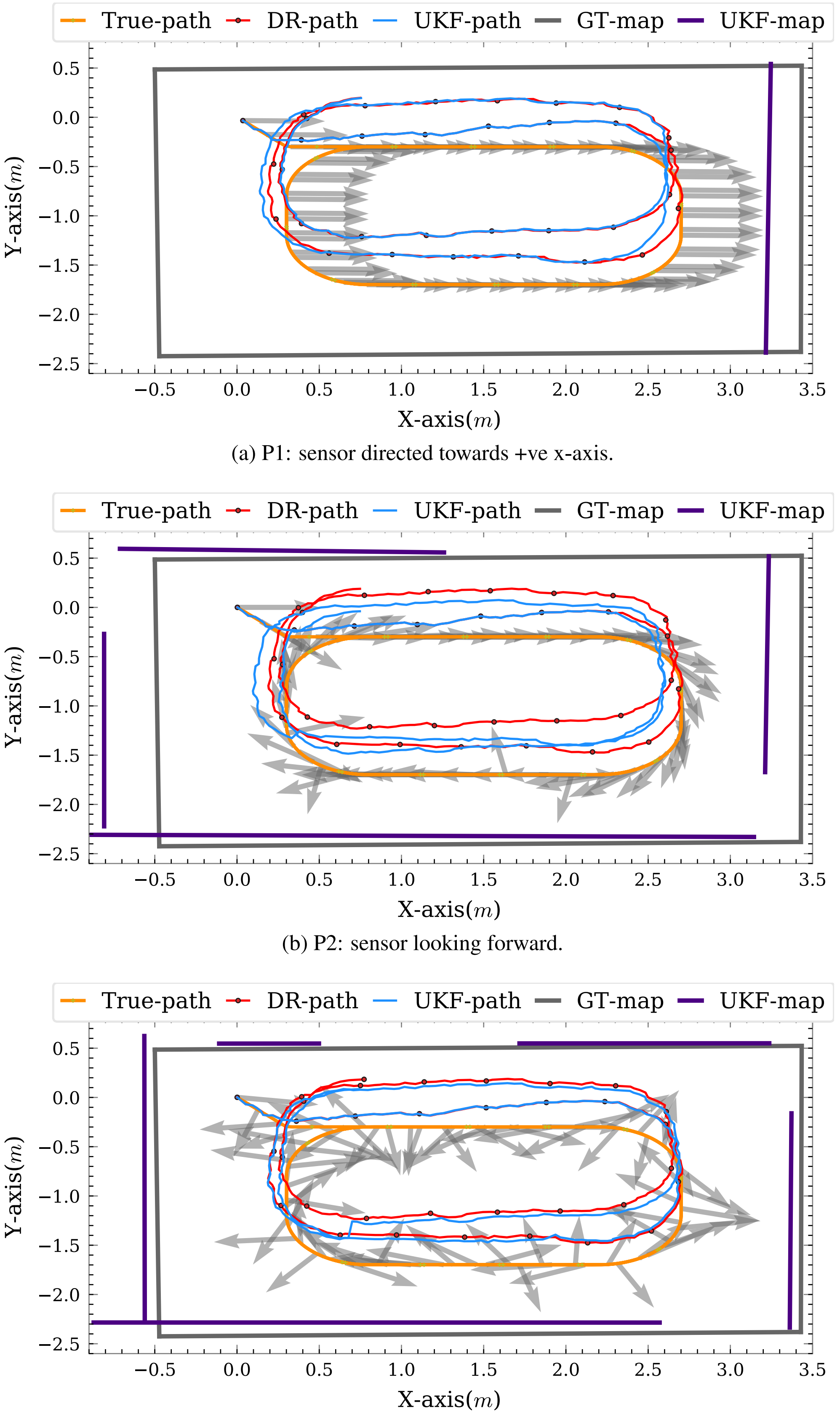


Figure 5.28: OSL tank test type-II UKF-SLAM XY position errors for different sensor configurations.

walls, the SLAM trajectory (see Figure 5.29 b) is significantly off the ground truth. This can be due to the high incidence angle when making sonar acquisitions while moving parallel to walls. Unlike previous test types, this trajectory expose the pitfall of configuration P3. In the previous two test types a contentious rotation is able to provide acceptable level of sonar measurement; however, for test type-III it is rather important to keep the sonar sensor orientation steady in the right direction. This is what both attentive sensor configurations (A1 and A2) managed to achieve using the decision from an FoA system. The sensor orientations in Figure 5.30 are noticeably stable for the long straight sections of the trajectory where the sonar is directed towards wall features. A direct comparison of the SLAM position error resulting from the five sensor configuration is shown in Figure 5.31. As expected the two attentive sensor orientations provide the best UKF-SLAM performance except for the first few seconds. During this period the robot moves from the starting position to the beginning of the loop where the sonar started. Lack of measurement early on causes a bias on the overall vehicle position leading to slightly shifted map.

All the OSL tank test results discussed so far based on a single run per test type and sensor configuration. However, before making any generalization it is important to see the repeatability of these tests. Obliviously, it is difficult to repeat the full test multiple times, hence for the next result the initial data is altered using a new sets of odometry noises to generate 100 distinct SLAM runs per test type per sensor configuration. The mean, STD and maximum X-Y position errors from these 100 runs are presented in Table 5.2-5.4. Table 5.2 and Table 5.3 show the summary of SLAM position error for test type-I and test type-II using the five sensor configurations. For these trajectories the best non attentive sensor configuration is P3 where the sensor is continuously moving. The movement of sensor for this test type seems important even for attentive sensing where sensor configuration A1 provides a much lower mean position error than A2. However for test type-III (see Table 5.4) the best non attentive sensor configuration is P2 where the sonar is looking forward. As it has been presented in Figure 5.29 for this test type even if the forward-looking sonar cannot get a strong sonar measurement looking directly into the walls for most part there is a small fraction of the sonar beam that hits the nearby boundary which leads to a relatively better performance. However, when it comes to attentive sensing still A1 provides the lowest navigation error yet this difference is less significant in case of test

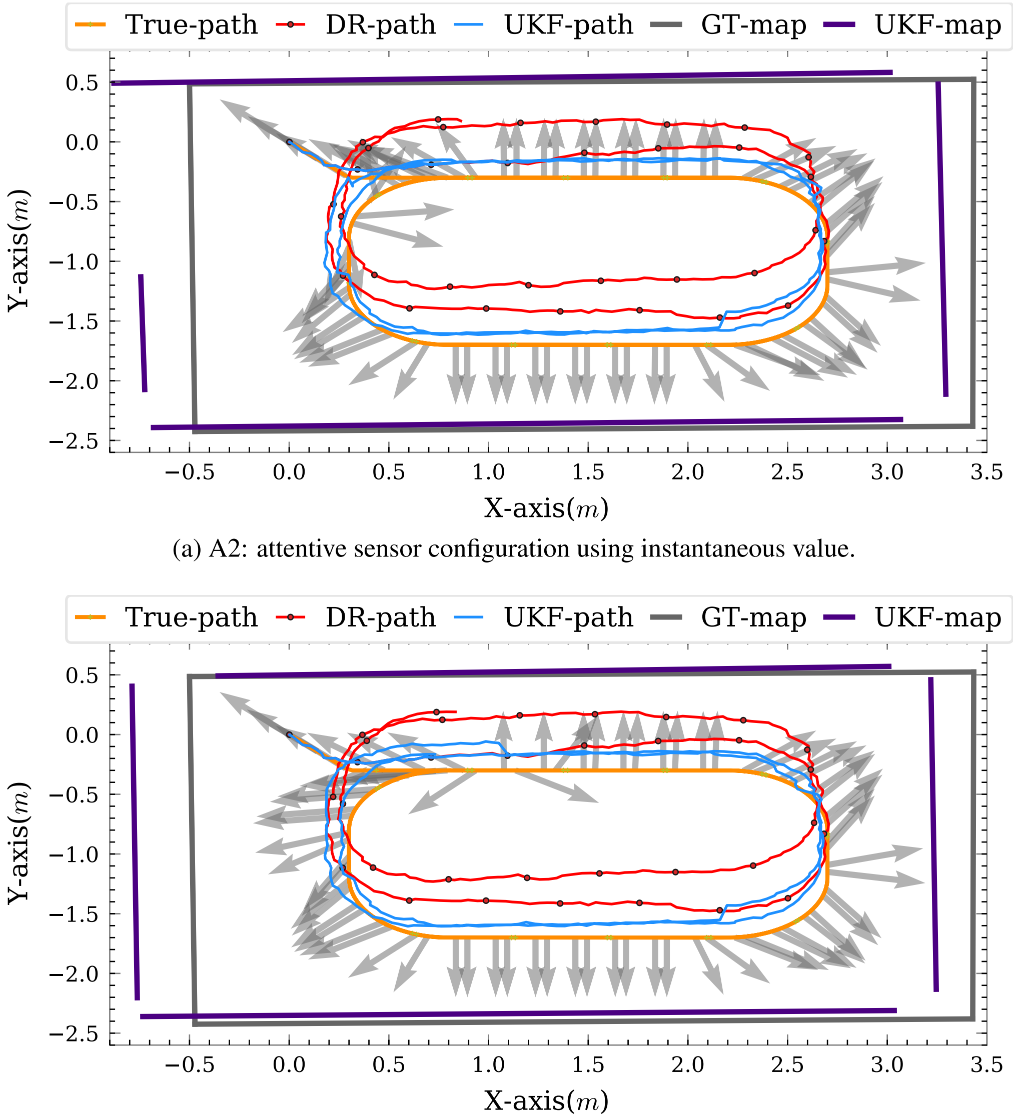


(a) P1: sensor directed towards +ve x-axis.

(b) P2: sensor looking forward.

(c) P3: sensor rotating continuously.

Figure 5.29: OSL tank test type-III SLAM trajectories for passive sensor configurations.



(a) A2: attentive sensor configuration using instantaneous value.

(b) A2: attentive sensor configuration with IoLR.

Figure 5.30: OSL tank test type-III SLAM trajectories for attentive sensor configurations.

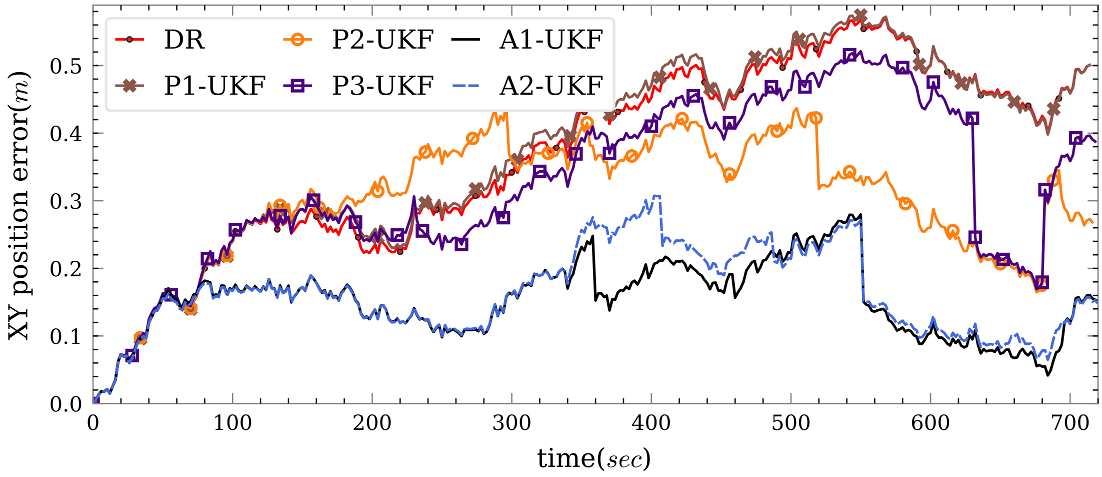


Figure 5.31: OSL tank test type-III UKF-SLAM XY position errors for different sensor configurations.

Sensor	mean		STD		max	
	DR	UKF	DR	UKF	DR	UKF
P1	0.300	0.229	0.049	0.060	0.486	0.387
P2	0.300	0.190	0.049	0.063	0.484	0.342
P3	0.301	0.151	0.049	0.056	0.490	0.297
A1	0.301	0.094	0.049	0.028	0.485	0.160
A2	0.300	0.116	0.049	0.038	0.483	0.206

Table 5.2: OSL tank SLAM test type-I result for different sensor configurations (P-passive, A-attentive).

Sensor	mean		STD		max	
	DR	UKF	DR	UKF	DR	UKF
P1	0.303	0.262	0.051	0.053	0.497	0.410
P2	0.303	0.261	0.051	0.061	0.497	0.424
P3	0.303	0.189	0.051	0.070	0.496	0.404
A1	0.302	0.124	0.051	0.041	0.494	0.257
A2	0.303	0.172	0.051	0.069	0.492	0.384

Table 5.3: OSL tank SLAM test type-II result for different sensor configurations (P-passive, A-attentive).

type-III which require minimum number of changes in attention angles anyway. To appreciate the advantage of attentive navigation see the SLAM performance for the run with the maximum DR navigation error. For instance for test type-I (see Table 5.2), the maximum RMS DR error is close to half a meter. Using the best passive sensor configuration it is only possible to achieve a minimum RMS SLAM error of 0.3 m which is almost double that of the equivalent RMS SLAM error for sensor configuration A1.

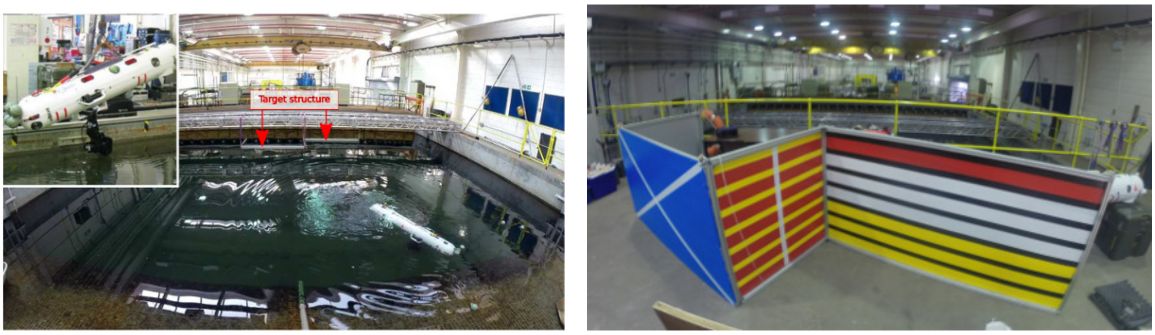
Here we want to underline the fact that SLAM error is not the only criterion of importance. It is important to considering the reduction in sensor movement and power consumption, as a result. These are two factors are essential for achieving persistence in a real AUV experiment. The next experiment present a comparison between attentive SLAM with IoLR and passive forward looking sensor configuration on *Nessie VII*.

5.4.3 Wave-tank test

This final experiment was performed in Heriot-Watt University wave tank facility (see Section 1.2) using MB2250 sonar mounted under *Nessie VII* as shown in Figure 5.32. Additional environmental features are added to the tank by hanging two L-shaped aluminium metallic frames as shown in Figure 5.32 (b). We found out that a thin aluminium plate

Sensor	mean		STD		max	
	DR	UKF	DR	UKF	DR	UKF
P1	0.295	0.227	0.044	0.059	0.456	0.375
P2	0.295	0.207	0.044	0.053	0.454	0.332
P3	0.296	0.220	0.044	0.043	0.457	0.344
A1	0.296	0.160	0.044	0.044	0.461	0.277
A2	0.296	0.171	0.044	0.047	0.457	0.304

Table 5.4: OSL tank SLAM test type-III result for different sensor configurations (P-passive, A-attentive).



(a) *Nessie VII* and HWU wave tank.

(b) Additional sonar structure.

Figure 5.32: Heriot-Watt university Wave tank test set-up. a) HWU indoor test environment (8m × 10m) and *Nessie VII* AUV with ROS PT-10-FB RS-485 pan and tilt unit underneath. b) Two metallic L shaped targets that are hanged in the middle of the tank to create additional features.

is invisible to sonar, hence the surface is decorated using a duck tape to provide surface roughness for sonar reflection.

A similar sequence of velocities are sent out for two separate runs yet there is no ground truth for these trajectories to asses the performance numerically. Throughout the experiment the forward velocity is kept under 0.22 m/sec while the maximum angular rotation is 15 DEG/sec. These velocities are assumed to be altered by a white Gaussian noise with STD 0.075 m/sec and 5 DEG/sec respectively. Clearly the output from a DVL sensor is much more accurate than that of a compass in a tank where there are significant amount metallic structures that affect the measurement. The starting position, which is also the map origin, is manually determined by aligning the vehicle perpendicular to one face of the metallic structure. Some example images for this experiment are shown in Figure 5.33, accordingly frames of the metallic structure result in bright sonar reflection and extra noise depending on the viewing angle. However sonar images from the wall and the artificial beach are less noisy and provide a smooth sonar line feature.

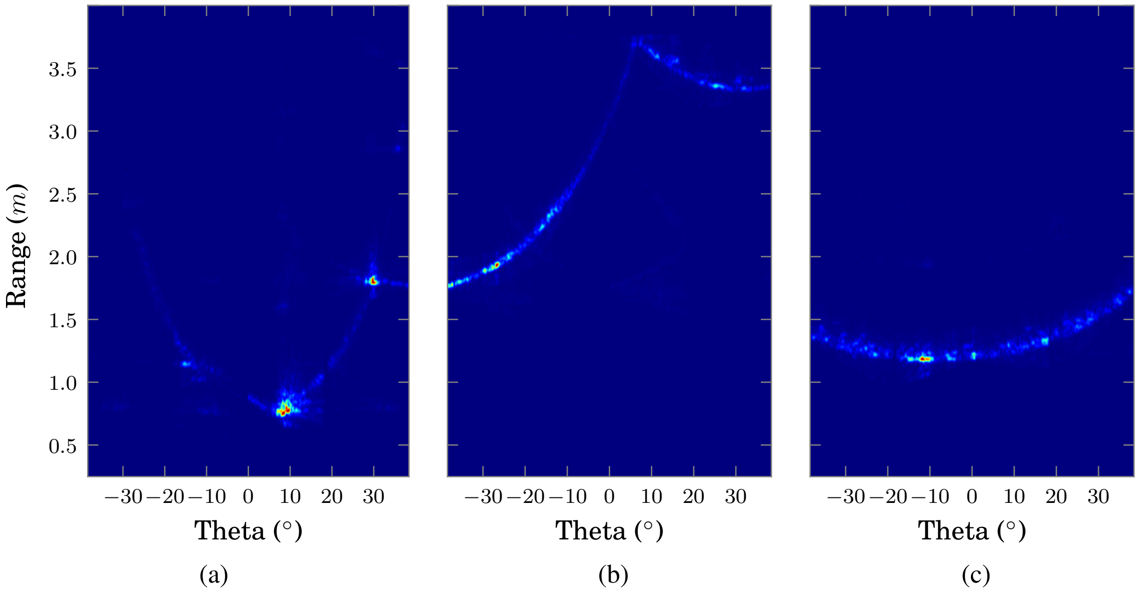


Figure 5.33: Example sonar images from HWU wave tank experiment. a) sonar reflection from the metallic structure, bright noisy spots arise from the frames. b) typical image view of concrete tank wall. c) sonar image of the artificial beach.

Figure 5.34 shows the result of a passive SLAM run where the sonar is mounted looking forward. In this configuration the robot does not manage to collect enough features to properly correct the navigation route. This mission lasted for 89 sec where the vehicle covers 15 m, yet 36% of the time the sonar sensor was looking away from any possible landmark. However, an attentive sonar salience based SLAM provide features 94% of the time as shown in Figure 5.35. In this case an SFoA-SLAM is used for determining sensor orientation according to Equation 5.6 where the scale parameter $k = 0.9$. Though there is no ground truth, the smooth robot path and structured map lines indicate the improvement in the robot localization and mapping by using attentive sensing. This is further strengthened by analysing the number and frequency of landmark features that are employed by the SLAM algorithm. As can be seen in Figure 5.36 an attentive SLAM provides many more features where individual landmark features appear across significant number of sonar acquisitions. Note that some of these lines features with appearance frequency less than 15 are not show in Figure 5.34 and 5.35 to avoid unnecessary clutter from the result. Most of these clutters are caused by the sonar noise arising from the metallic frame around corners of the metallic structure.

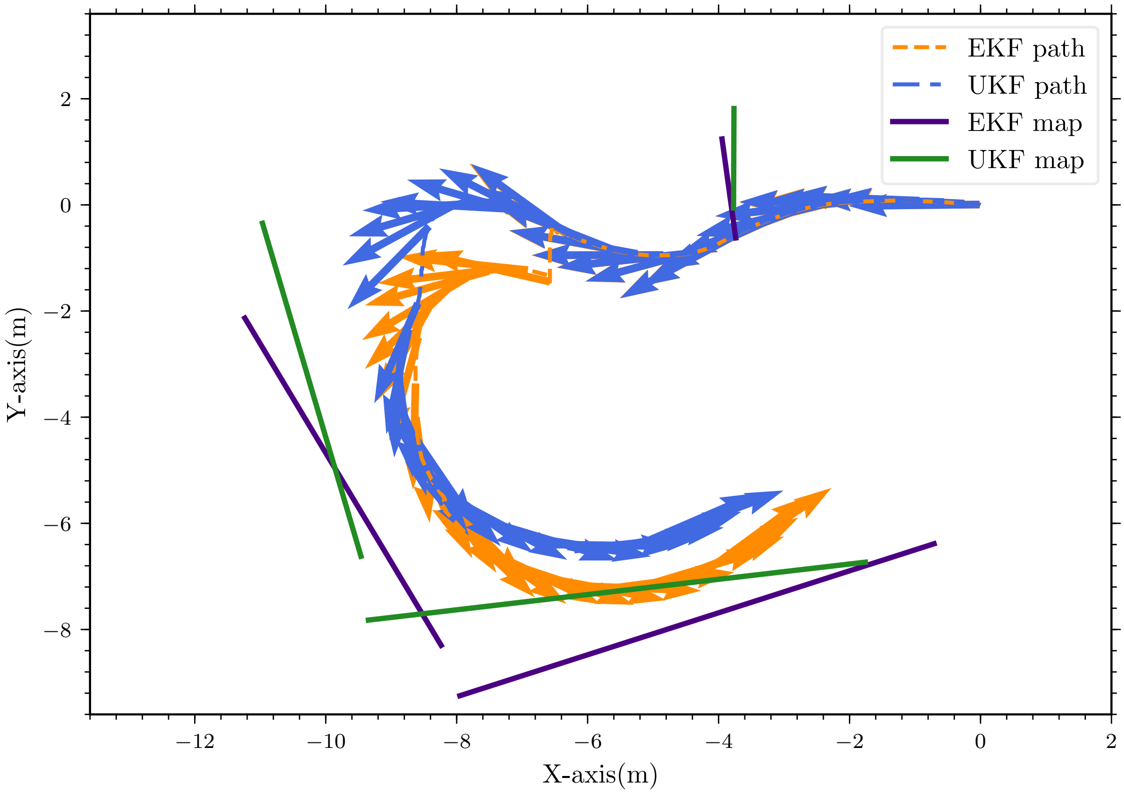


Figure 5.34: HWU Wave-tank passive SLAM test robot path, SLAM map and sonar orientation using EKF and UKF.

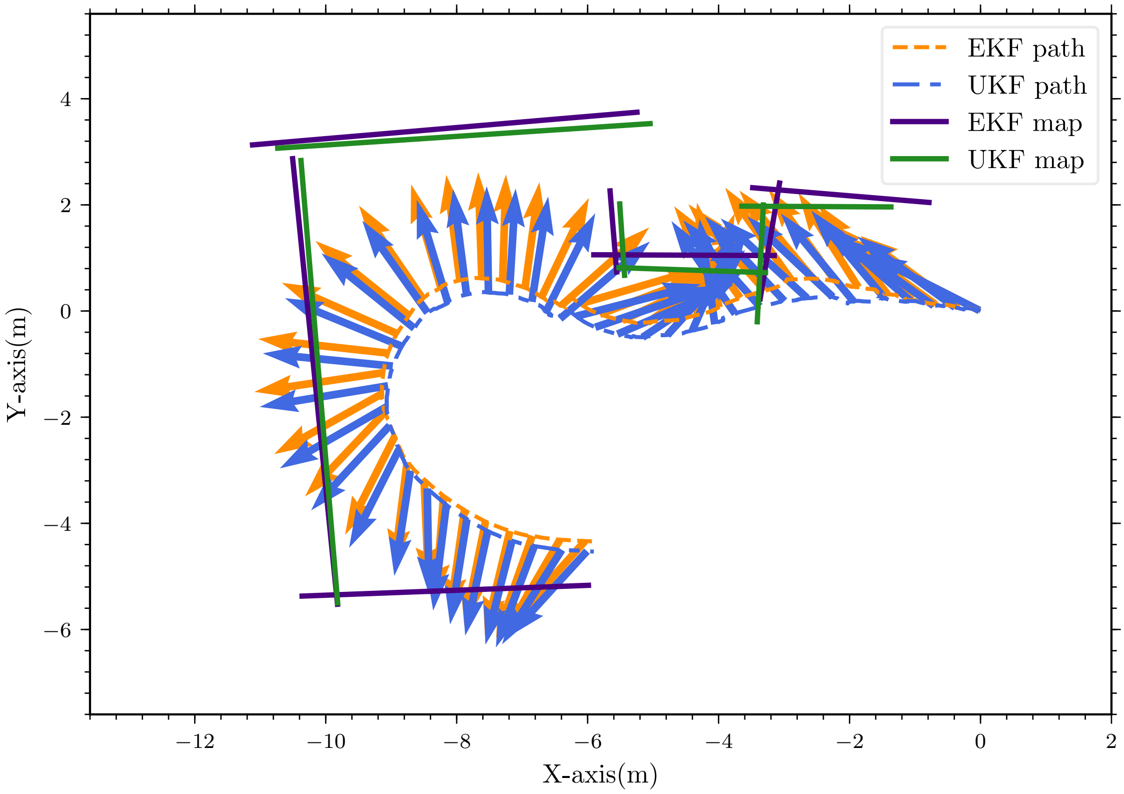
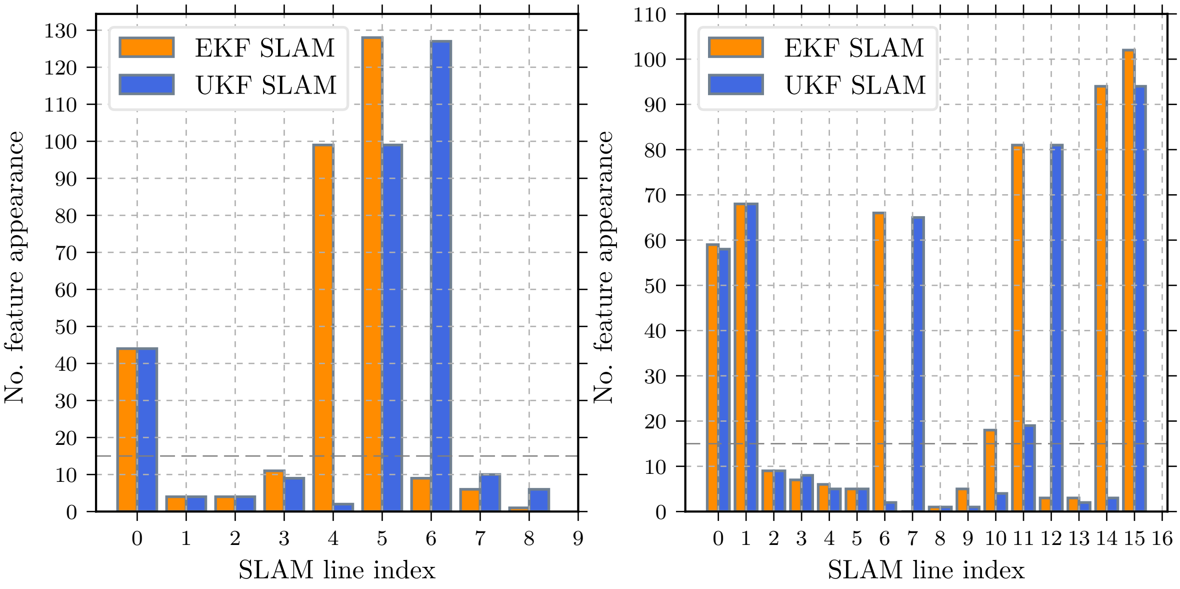


Figure 5.35: HWU wave-tank attentive EKF and UKF SLAM test. Arrows indicate sonar heading determined using SFoA while solid lines indicate final SLAM maps.



(a) Passive SLAM

(b) Attentive SLAM

Figure 5.36: SLAM line features and frequency of their visibility throughout a mission.

5.5 Summary

This chapter has introduced an attentive sonar salience based focus of attention for SLAM navigation. Earlier in the chapter a novel technique for sonar salience computation is proposed using three sonar constraint based weights: angle of incidence, feature distance and overlap on the sonar field of view. The remainder of the chapter demonstrated the performance of an attentive SLAM algorithm that uses SFoA for sensor orientation. Based on simulated experiment, a guideline is suggested for choosing acceptable parameter values yet the best parameter combinations depend on the actual vehicle trajectory and environment. Various tests done in OSL tank using precise Cartesian robot and real sonar sensor provide the level of improvement that an attentive SLAM can bring into a passive equivalent. Based on a final experiment performed in HWU wave-tank using *Nessie VII* AUV, SFoA is found out to provide more features where most of the features are frequently viewed throughout a mission to provide landmark based navigation correction. This in turn improves the quality of the SLAM algorithm as demonstrated using the final estimated map.

Chapter 6

Conclusion

Navigation being one of the critical problems in underwater AUV applications, this thesis propose a novel attentive SLAM based navigation system. Among the various navigation techniques available, there is a growing interest towards SLAM techniques, for their ability to make the most out of environmental landmarks. However, in an environment where the availability of landmark features is scarce the proper use of these landmarks is very important for achieving high quality navigation. The work in this thesis propose a sonar focus of attention system for online sensor planning to improve a robust underwater SLAM based navigation. In this chapter a summary of this work is presented along a list of main contributions. Then the chapter is concluded by discussing future research directions.

6.1 Summary and Contributions

After a brief introduction, the thesis provides an overall background on topics related to underwater navigation in Chapter 2. The chapter begins by discussing state-of-the-art underwater navigation techniques and sensors used for navigation purpose. This discussion indicates that for a robust navigation an AUV need to understand the surrounding environment rather than relying fully on on-board sensors which can easily be affected by measurement drifts. Hence it was shown that for most real underwater applications SLAM based navigation techniques are the best options. The chapter also presents a review on sonar underwater feature representation and extraction. Later on the chapter the importance of active sensing for navigation is highlighted putting emphasis on focus of attention to improve the quality and quantity of sonar information for SLAM based navigation.

Chapter 3 focuses on sonar feature extraction. In this chapter two BlueView multibeam sonar sensors are characterised and then simulated using two approaches. In the first approach each sonar beam is considered as independent range measurement which are modelled as 1-D Gaussian distribution. In the later case, a sonar image representation of a line is formed using multiple line segments generated based on the line's uncertainty. The chapter compares three widely used line extraction techniques using simulation and real sonar images. The validation tests confirm *split-and-merge* technique is relatively faster and can provide a precise estimation of line segments. Further improvement on line extraction can be achieved by using intensity weights in the final refinement stage where the location as well as intensity of sonar beam peaks are used in a weighted least square fitting.

Chapter 4 presents a SLAM framework using sonar line features. Although EKF filter is one of the most commonly used SLAM filter, it can easily diverge due to either poor implementation or strong non-linearity. One of the common problem in line feature based SLAM is the use of single observation model which in certain scenario leads to divergence. On the other hand, an EKF based SLAM fails to cope with a strong non linearity arising from high angular rotation or an observation of line feature far out from the map origin. In these situations better result can be achieved by using a UKF filter, which can be implemented at similar computation complexity as an EKF-SLAM as shown in [112]. The use of FISST for solving the SLAM problem is rising trend yet as discussed in the chapter the real time application of GM-PHD SLAM still require further technological advancement.

Finally, in Chapter 5 we introduced a novel sensor constraint based sonar salience map computation technique. The result from an extensive simulated and real AUV tests demonstrate that the use of SFoA can improve the accuracy of a SLAM underwater navigation by selecting informative features with in the reach of the sonar sensor using a pan and tilt unit. Comparisons are provided between various sensor configurations both in passive and active mode.

6.2 Review of Contributions

The main contributions presented in this work are listed as follows:

- An improved ray casting based multibeam sonar simulation which uses characterisation of real sonar sensor for beam simulation.

- In addition to the position of sonar peaks in an image, the intensity value is also used for weighted least square line extraction which improves the accuracy of the process. The use of weight is also adapted to Hough-transform based line extraction.

- When it comes to filter divergence in line feature based EKF SLAM non-linearity is not the only challenge. In this work we highlight the importance of having two complementary observation models to avoids error arising due to polar line representation.

- In tackling a strong non-linearity either in motion or observation we showed the capability of UKF based SLAM over the most commonly used EKF-SLAM.

- The most important contribution of this thesis is the introduction of sonar based focus of attention system which uses sensor constraints to compute sonar salience map. This SFoA system is integrated inside an attentive SLAM framework to provide a robust underwater navigation system in structured environments.

6.3 Future Work

There are various areas where the research presented can be extended. Next, we address some of this possible extenuations:

Alternative technique for WTA: In the proposed SFoA based sensor planning, the decision of the final orientation is made using winner-takes-all (WTA) rule which is the most common strategy in most computational attention mechanisms. A decision made solely on WTA strategy can result in a huge transition gap between two consecutive measurement poses which put a pan and tilt unit under constant load. As a remedy for such problems, temporal smoothing is suggested where the instantaneous WTA values for the past n-measurement poses are averaged to determine the final orientation of the sensor. However, these WTA attention points are a result of actual landmarks which actually appear gradually. Hence, we believe the use of second and higher order maxima in the initial decision can provide a prediction capability and a smooth attention transition.

Salience map for loop-closer: As shown in this work one application of sonar salience map is to determine sensor orientation while traversing a pre-defined navigation route. However, some of the work on active navigation presented in Chapter 2 indicates the importance of combining path planning and navigation. In order to maximize landmark based navigation correction, navigation routes need to satisfy sensor requirement on top of other compulsory mission objectives. If the full vehicle trajectory or map is inaccessible before a mission, deliberate loop-closer can be applied to avoid accumulation of measurement drift by going back to past vehicle poses with strong global sonar salience.

Salience map for other feature types: One way of extending the proposed sonar salience map computation technique require adaptation of the method to features other than line segments. Although line features are found to be sufficient for representation of most man made structures, other types of features are imminent when working in an open sea where landmarks are natural sea bed structures. Additionally, for some applications the dimension of the SLAM operation might require an expansion in to 3D. In such scenario the salience map has to be defined using 3D space which then can be achieved through the use of the full pan and tilt unit capability. The challenge in such set-up would increase the computational complexity.

Bibliography

[1] D. M. Lane, F. Maurelli, T. Larkworthy, D. Caldwell, J. Salvi, M. Fox, and K. Kyriakopoulos, "PANDORA: Persistent autonomy through learning, adaptation, observation and re-planning," in *Proceedings of IFAC - Navigation, Guidance and Control of Underwater Vehicles*, 2012.

[2] D. Ribas, P. Ridao, J. D. Tardós, and J. Neira, "Underwater SLAM in man-made structured environments," *Journal of Field Robotics*, vol. 25, no. 11-12, pp. 898–921, 2008.

[3] A. Garulli, A. Giannitrapani, A. Rossi, and A. Vicino, "Mobile robot SLAM for line-based environment representation," in *Proceedings of the 44th IEEE Conference on Decision and Control*, pp. 2041–2046, IEEE, 2005.

[4] M. Prats, J. Perez, J. Fernandez, and P. Sanz, "An open source tool for simulation and supervision of underwater intervention missions," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2577–2582, 2012.

[5] J. C. Kinsey, R. M. Eustice, and L. L. Whitcomb, "A survey of underwater vehicle navigation: Recent advances and new challenges," in *IFAC Conference of Manoeuvering and Control of Marine Craft*, vol. 88, 2006.

[6] L. Paull, S. Saeedi, M. Seto, and H. Li, "AUV navigation and localization: A review," *IEEE Journal of Oceanic Engineering*, vol. 39, no. 1, pp. 131–149, 2014.

[7] S. Augenstein and S. Rock, "Estimating inertial position and current in the midwater," in *Proceedings of the IEEE/MTS OCEANS Conference and Exhibition*, pp. 1–6, Sept 2008.

[8] P. Krishnamurthy and F. Khorrami, "A self-aligning underwater navigation system based on fusion of multiple sensors including DVL and IMU," in *Proceedings of the 9th Asian Control Conference*, pp. 1–6, June 2013.

[9] J. Jouffroy, "A relaxed criterion for contraction theory: Application to an underwater vehicle observer," in *Proceedings of the European Control Conference*, 2003.

[10] L. Stutters, H. Liu, C. Tiltman, and D. Brown, "Navigation technologies for autonomous underwater vehicles," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 38, pp. 581–589, July 2008.

[11] O. Hegrenaes, E. Berglund, and O. Hallingstad, "Model-aided inertial navigation for underwater vehicles," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1069–1076, IEEE, 2008.

[12] B. Jalving, K. Gade, O. Hagen, and K. Vestgard, “A toolbox of aiding techniques for the HUGIN AUV integrated inertial navigation system,” in *Proceedings of the IEEE/MTS OCEANS Conference and Exhibition*, vol. 2, pp. 1146–1153, Sept 2003.

[13] O. Hegrenaes and E. Berglund, “Doppler water-track aided inertial navigation for autonomous underwater vehicle,” in *Proceedings of the IEEE/MTS OCEANS Conference and Exhibition*, pp. 1–10, May 2009.

[14] D. Ribas, P. Ridao, and J. Neira, “Simultaneous localization and mapping,” in *Underwater SLAM for Structured Environments Using an Imaging Sonar*, vol. 65 of *Springer Tracts in Advanced Robotics*, pp. 77–112, Springer Berlin Heidelberg, 2010.

[15] F. A. Faruqi and K. J. Turner, “Extended kalman filter synthesis for integrated global positioning/inertial navigation systems,” *Applied Mathematics and Computation*, vol. 115, no. 2-3, pp. 213–227, 2000.

[16] L. Zhao and W. Gao, “The experimental study on GPS/INS/DVL integration for AUV,” in *Proceedings of the IEEE/ION Position, Location and Navigation Symposium*, pp. 337–340, April 2004.

[17] M. Morgado, P. Oliveira, C. Silvestre, and J. Vasconcelos, “USBL/INS tightly-coupled integration technique for underwater vehicles,” in *Proceedings of the 9th IEEE International Conference on Information Fusion*, pp. 1–8, July 2006.

[18] L. Chen, S. Wang, K. McDonald-Maier, and H. Hu, “Towards autonomous localization and mapping of AUVs: a survey,” *International Journal of Intelligent Unmanned Systems*, vol. 1, no. 2, pp. 97–120, 2013.

[19] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents).* The MIT Press, 2005.

[20] P. Zhang, J. Gu, E. Milios, and P. Huynh, “Navigation with IMU/GPS/digital compass with unscented kalman filter,” in *Proceedings of the IEEE International Conference on Mechatronics and Automation*, vol. 3, pp. 1497–1502, 2005.

[21] B. Allotta, L. Chisci, R. Costanzi, F. Fanelli, C. Fantacci, E. Meli, A. Ridolfi, A. Caiti, F. Di Corato, and D. Fenucci, “A comparison between EKF-based and UKF-based navigation algorithms for AUVs localization,” in *Proceedings of the IEEE/MTS OCEANS Conference and Exhibition*, pp. 1–5, May 2015.

[22] G. Fagogenis, D. Flynn, and D. M. Lane, “Improving underwater vehicle navigation state estimation using locally weighted projection regression,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 6549–6554, IEEE, 2014.

[23] P. Oliveira, “MMAE terrain reference navigation for underwater vehicles using eigen analysis,” in *Proceedings of the 44th IEEE Conference on Decision and Control*, pp. 5468–5473, Dec 2005.

[24] Z. Tao, “Autonomous underwater vehicle navigation using an adaptive kalman filter for sensor fusion,” in *Proceedings of the 29th Chinese Control Conference*, pp. 1588–1591, July 2010.

[25] Y. Sun, J. Sun, L. Wan, C. Li, and Y. Zhang, “An improved self-adaptive kalman filter for underwater integrated navigation system based on DR,” in *Proceedings of the 2nd International Conference on Intelligent Control and Information Processing*, vol. 2, pp. 993–998, July 2011.

[26] E. Hernández, P. Ridao, D. Ribas, and A. Mallios, “Probabilistic sonar scan matching for an AUV,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 255–260, IEEE, 2009.

[27] A. Mallios, P. Ridao, D. Ribas, and E. Hernández, “Scan matching slam in underwater environments,” *Journal of Autonomous Robots*, vol. 36, no. 3, pp. 181–198, 2014.

[28] A. Burguera, Y. González, and G. Oliver, “Underwater scan matching using a mechanical scanned imaging sonar,” *Proceedings of Intelligent Autonomous Vehicles (IAV)*, 2010.

[29] J. J. Leonard and H. F. Durrant-Whyte, “Mobile robot localization by tracking geometric beacons,” *IEEE Transactions on robotics and Automation*, vol. 7, no. 3, pp. 376–382, 1991.

[30] R. C. Smith and P. Cheeseman, “On the representation and estimation of spatial uncertainty,” *The international journal of Robotics Research*, vol. 5, no. 4, pp. 56–68, 1986.

[31] J. L. Leonard, R. N. Carpenter, and H. J. S. Feder, “Stochastic mapping using forward look sonar,” *Robotica*, vol. 19, no. 05, pp. 467–480, 2001.

[32] I. Ruiz, Y. Petillot, D. Lane, and C. Salson, “Feature extraction and data association for AUV concurrent mapping and localisation,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2785–2790, 2001.

[33] I. T. Ruiz, S. De Raucourt, Y. Petillot, and D. M. Lane, “Concurrent mapping and localization using sidescan sonar,” *IEEE Journal of Oceanic Engineering*, vol. 29, no. 2, pp. 442–456, 2004.

[34] J. Aulinas, X. Lladó, J. Salvi, and Y. R. Petillot, “Selective submap joining for underwater large scale 6-DOF SLAM,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2552–2557, Oct 2010.

[35] C. Roman and H. Singh, “Improved vehicle based multibeam bathymetry using submaps and SLAM,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3662–3669, IEEE, 2005.

[36] C. S. Lee, S. Nagappa, N. Palomeras, D. E. Clark, and J. Salvi, “Slam with sc-phd filters: An underwater vehicle application,” *IEEE Robotics and Automation Magazine*, vol. 21, no. 2, pp. 38–45, 2014.

[37] M. Pfingsthorn and A. Birk, “Simultaneous localization and mapping with multimodal probability distributions,” *The International Journal of Robotics Research*, vol. 32, no. 2, pp. 143–171, 2013.

[38] M. Pfingsthorn and A. Birk, “Generalized graph slam: Solving local and global ambiguities through multimodal and hyperedge constraints,” *The International Journal of Robotics Research*, vol. 35, no. 6, pp. 601–630, 2016.

[39] P. Agarwal, G. D. Tipaldi, L. Spinello, C. Stachniss, and W. Burgard, “Robust map optimization using dynamic covariance scaling,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 62–69, IEEE, 2013.

[40] Y. Latif, C. Cadena, and J. Neira, “Robust loop closing over time for pose graph slam,” *The International Journal of Robotics Research*, vol. 32, no. 14, pp. 1611–1626, 2013.

[41] E. Olson and P. Agarwal, “Inference on networks of mixtures for robust robot mapping,” *The International Journal of Robotics Research*, vol. 32, no. 7, pp. 826–840, 2013.

[42] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, “isam2: Incremental smoothing and mapping using the bayes tree,” *The International Journal of Robotics Research*, vol. 31, no. 2, pp. 216–235, 2012.

[43] N. Sünderhauf, *Robust optimization for simultaneous localization and mapping*. PhD thesis, Technischen Universitat Chemnitz, 2012.

[44] N. Sünderhauf and P. Protzel, “Switchable constraints for robust pose graph SLAM,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1879–1884, Oct 2012.

[45] T. Tuytelaars and K. Mikolajczyk, “Local invariant feature detectors: a survey,” *Found. Trends. Comput. Graph. Vis.*, vol. 3, pp. 177–280, July 2008.

[46] H. Wang, Z. Zheng, and E. K. Teoh, “Gray-level corner detection,” *IAPR Workshop on Machine Vision Application*, vol. 1, pp. 207–200, 1998.

[47] C. Harris and M. Stephens, “A combined corner and edge detector.,” in *Proceedings of the Alvey vision conference*, vol. 15, p. 50, Manchester, UK, 1988.

[48] K. Mikolajczyk and C. Schmid, “Scale & affine invariant interest point detectors,” *International Journal of Computer Vision*, vol. 60, no. 1, pp. 63–86, 2004.

[49] H. Bay, T. Tuytelaars, and L. Gool, “SURF : Speeded up robust features,” in *European Conference in Computer Vision* (A. Leonardis, H. Bischof, and A. Pinz, eds.), vol. 3951 of *Lecture Notes in Computer Science*, pp. 404–417, Springer Berlin Heidelberg, 2006.

[50] D. G. Lowe, “Object recognition from local scale-invariant features,” in *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2, pp. 1150–1157, 1999.

[51] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 511–518, 2001.

[52] J. Sawas, Y. Petillot, and Y. Pailhas, "Cascade of boosted classifiers for rapid detection of underwater objects," in *Proceedings of the European Conference on Underwater Acoustics*, 2010.

[53] J. Aulinas and X. L. Y. P. J. S. R. G. A. Fazlollahi, J. Salvi, "Robust automatic landmark detection for underwater SLAM using side-scan sonar imaging," in *Proceedings of the International Conference on Mobile Robots and Competitions*, pp. 21–26, 2011.

[54] M. Aykin and S. Negahdaripour, "On feature extraction and region matching for forward scan sonar imaging," in *Proceedings of the IEEE/MTS OCEANS Conference and Exhibition*, pp. 1–9, oct. 2012.

[55] I. Tena Ruiz, D. Lane, and M. Chantler, "A comparison of inter-frame feature measures for robust object classification in sector scan sonar image sequences," *IEEE Journal of Oceanic Engineering*, vol. 24, pp. 458–469, oct 1999.

[56] E. Trucco, Y. Petillot, I. Ruiz, K. Plakas, and D. Lane, "Feature tracking in video and sonar subsea sequences with applications," *Computer Vision and Image Understanding*, vol. 79, pp. 92–122, jul 2000.

[57] D. Ribas, P. Ridao, J. Neira, and J. D. Tardós, "A method for extracting lines and their uncertainty from acoustic underwater images for SLAM," *Proceedings of IFAC*, vol. 40, no. 15, pp. 61–66, 2007.

[58] D. Ribas, *Underwater SLAM for Structured Environments Using an Imaging Sonar*. PhD thesis, University of Girona, Girona, Spain, September 2008.

[59] K. Pathak, A. Birk, and N. Vaskevicius, "Plane-based registration of sonar data for underwater 3d mapping," in *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, pp. 4880–4885, IEEE, 2010.

[60] K. Pathak, A. Birk, N. Vaskevicius, and J. Poppinga, "Fast registration based on noisy planes with unknown correspondences for 3-d mapping," *IEEE Transactions on Robotics*, vol. 26, no. 3, pp. 424–441, 2010.

[61] H. Bülow and A. Birk, "Spectral registration of noisy sonar data for underwater 3D mapping," *Autonomous Robots*, vol. 30, no. 3, pp. 307–331, 2011.

[62] N. Hurtós, D. Ribas, X. Cufí, Y. Petillot, and J. Salvi, "Fourier-based registration for robust forward-looking sonar mosaicing in low-visibility underwater environments," *Journal of Field Robotics*, vol. 32, no. 1, pp. 123–151, 2015.

[63] V. Nguyen, S. Gächter, A. Martinelli, N. Tomatis, and R. Siegwart, "A comparison of line extraction algorithms using 2d range data for indoor mobile robotics," *Auton. Robots*, vol. 23, pp. 97–111, Aug. 2007.

[64] R. Bajcsy, "Active perception," *Proceedings of the IEEE*, vol. 76, no. 8, pp. 966–1005, 1988.

[65] S. Chen, Y. Li, and N. M. Kwok, "Active vision in robotic systems: A survey of recent developments," *Int. J. Rob. Res.*, vol. 30, pp. 1343–1377, Sept. 2011.

[66] Y. Li and Z. Liu, "Method for determining the probing points for efficient measurement and reconstruction of freeform surfaces," *Measurement Science and Technology*, vol. 14, no. 8, p. 1280, 2003.

[67] T.-H. Tsai and K.-C. Fan, "An image matching algorithm for variable mesh surfaces," *Measurement*, vol. 40, no. 3, pp. 329–337, 2007.

[68] S. Treuillet, B. Albouy, and Y. Lucas, "Finding two optimal positions of a hand-held camera for the best reconstruction," in *IEEE 3DTV Conference*, pp. 1–4, 2007.

[69] K. Jonnalagadda, R. Lumia, G. Starr, and J. Wood, "Viewpoint selection for object reconstruction using only local geometric features," in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2, pp. 2116–2122, 2003.

[70] M. Chang and S. C. Park, "Automated scanning of dental impressions," *Computer-Aided Design*, vol. 41, no. 6, pp. 404–411, 2009.

[71] D. MacKinnon, V. Aitken, and F. Blais, "Adaptive laser range scanning," in *Proceedings of the IEEE American Controls Conference*, pp. 3857–3862, 2008.

[72] Y. Huang and X. Qian, "A dynamic sensing-and-modeling approach to three-dimensional point-and area-sensor integration," *transactions of the ASME*, vol. 129, no. 3, p. 623, 2007.

[73] G. A. Hollinger, B. Englot, F. S. Hover, U. Mitra, and G. S. Sukhatme, "Active planning for underwater inspection and the benefit of adaptivity," *The International Journal of Robotics Research*, vol. 32, no. 1, pp. 3–18, 2013.

[74] S. Se and P. Jasiobedzki, "Stereo-vision based 3d modeling for unmanned ground vehicles," in *Proceedings of SPIE 6561, Unmanned ground vehicle*, pp. 65610–65610X, International Society for Optics and Photonics, 2007.

[75] K. Shubina and J. K. Tsotsos, "Visual search for an object in a 3d environment using a mobile robot," *Computer Vision and Image Understanding*, vol. 114, no. 5, pp. 535–547, 2010.

[76] T. A. Riggs, T. Inanc, and W. Zhang, "An autonomous mobile robotics testbed: construction, validation, and experiments," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 3, pp. 757–766, 2010.

[77] J. Barreto, L. Perdigoto, R. Caseiro, and H. Araujo, "Active stereo tracking of n less than or equal to 3 targets using line scan cameras," *IEEE Transactions on Robotics*, vol. 26, no. 3, pp. 442–457, 2010.

[78] H. Zhou and S. Sakane, "Learning bayesian network structure from environment and sensor planning for mobile robot localization," in *Proceedings of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pp. 76–81, IEEE, 2003.

[79] H. H. González-Banos and J.-C. Latombe, "Navigation strategies for exploring indoor environments," *The International Journal of Robotics Research*, vol. 21, no. 10-11, pp. 829–848, 2002.

[80] R. Sim and N. Roy, "Global a-optimal robot exploration in slam," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 661–666, IEEE, 2005.

[81] C. Leung, S. Huang, and G. Dissanayake, "Active SLAM in structured environments," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1898–1903, 2008.

[82] G. Zhang, S. Ferrari, and M. Qian, "An information roadmap method for robotic sensor path planning," *Journal of Intelligent and Robotic Systems*, vol. 56, no. 1-2, pp. 69–98, 2009.

[83] M. Liu, Y. shen Liu, and K. Ramani, "Computing global visibility maps for regions on the boundaries of polyhedra using minkowski sums," *Computer-Aided Design*, vol. 41, no. 9, pp. 668–680, 2009.

[84] S. Y. Chen and Y. Li, "Automatic sensor placement for model-based robot vision," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 34, no. 1, pp. 393–408, 2004.

[85] F. Farshidi, S. Sirouspour, and T. Kirubarajan, "Robust sequential view planning for object recognition using multiple cameras," *Image and Vision Computing*, vol. 27, no. 8, pp. 1072–1082, 2009.

[86] M. Begum and F. Karray, "Visual attention for robotic cognition: A survey," *IEEE Transactions on Autonomous Mental Development*, vol. 3, no. 1, pp. 92–105, 2011.

[87] Y. Yu, G. K. I. Mann, and R. Gosine, "An object-based visual attention model for robotic applications," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 40, no. 5, pp. 1398–1412, 2010.

[88] Y. Sun, R. Fisher, F. Wang, and H. M. Gomes, "A computer vision model for visual-object-based attention and eye movements," *Computer Vision and Image Understanding*, vol. 112, no. 2, pp. 126–142, 2008.

[89] C. Koch and S. Ullman, "Shifts in selective visual attention: towards the underlying neural circuitry," in *Matters of Intelligence*, pp. 115–141, Springer, 1987.

[90] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[91] S. Frintrop, *VOCUS: A visual attention system for object detection and goal-directed search*, vol. 3899. Springer, 2006.

[92] G. W. Kootstra, *Visual attention and active vision: from natural to artificial systems*. PhD thesis, 2010.

[93] B. A. Draper and A. Lionelle, "Evaluation of selective attention under similarity transformations," *Computer Vision and Image Understanding*, vol. 100, no. 1-2, pp. 152–171, 2005.

[94] B. Rasolzadeh, M. Björkman, K. Hübner, and D. Kragic, “An active vision system for detecting, fixating and manipulating objects in the real world,” *The International Journal of Robotics Research*, vol. 29, no. 2-3, pp. 133–154, 2010.

[95] F. Miau, C. S. Papageorgiou, and L. Itti, “Neuromorphic algorithms for computer vision and attention,” vol. 4479, pp. 12–23, 2001.

[96] D. Walther, U. Rutishauser, C. Koch, and P. Perona, “On the usefulness of attention for object recognition,” in *ECCV Workshop on Attention and Performance in Computational Vision*, pp. 96–103, 2004.

[97] D. Walther, D. Edgington, and C. Koch, “Detection and tracking of objects in underwater video,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 544–549, 2004.

[98] K. Lee, H. Buxton, and J. Feng, “Cue-guided search: a computational model of selective attention,” *IEEE Transactions on Neural Networks*, vol. 16, no. 4, pp. 910–924, 2005.

[99] Y. Hu, X. Xie, W.-Y. Ma, L.-T. Chia, and D. Rajan, “Salient region detection using weighted feature maps based on the human visual attention model,” in *PCM on Advances in Multimedia Information Processing* (K. Aizawa, Y. Nakamura, and S. Satoh, eds.), vol. 3332 of *Lecture Notes in Computer Science*, pp. 993–1000, Springer Berlin Heidelberg, 2005.

[100] G. Kootstra, J. Ypma, and B. de Boer, “Active exploration and keypoint clustering for object recognition,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1005–1010, IEEE, 2008.

[101] Y. Yu, G. K. I. Mann, and R. Gosine, “A goal-directed visual perception system using object-based top down attention,” *IEEE Transactions on Autonomous Mental Development*, vol. 4, no. 1, pp. 87–103, 2012.

[102] Y. Yu, J. Gu, G. Mann, and R. Gosine, “Development and evaluation of object-based visual attention for automatic perception of robots,” *IEEE Transactions on Automation Science and Engineering*, vol. 10, no. 2, pp. 365–379, 2013.

[103] S. Frintrop and P. Jensfelt, “Attentional landmarks and active gaze control for visual SLAM,” *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 1054–1065, 2008.

[104] E. Einhorn, C. Schröter, and H. Gross, “Attention-driven monocular scene reconstruction for obstacle detection, robot navigation and map building,” *Robotics and Autonomous Systems*, vol. 59, no. 5, pp. 296–309, 2011.

[105] A. Kim and R. M. Eustice, “Real-time visual slam for autonomous underwater hull inspection using visual saliency,” *IEEE Transactions on Robotics*, vol. 29, no. 3, pp. 719–733, 2013.

[106] A. Kim, *Active visual SLAM with exploration for autonomous underwater navigation*. PhD thesis, 2012.

[107] BlueView Technologies, Inc., “Home - teledyne-blueview.” http://www.blueview.com. Accessed: 2016-09-14.

[108] V. Nguyen, A. Martinelli, N. Tomatis, and R. Siegwart, "A comparison of line extraction algorithms using 2d laser rangefinder for indoor mobile robotics," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1929–1934, IEEE, 2005.

[109] G. C. Karras, C. P. Bechlioulis, H. K. Abdella, T. Larkworthy, K. Kyriakopoulos, and D. Lane, "A robust sonar servo control scheme for wall-following using an autonomous underwater vehicle," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3893–3898, Nov 2013.

[110] R. Lakaemper, "Simultaneous multi-line-segment merging for robot mapping using mean shift clustering," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1654–1660, 2009.

[111] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part i," *Robotics Automation Magazine, IEEE*, vol. 13, pp. 99–110, June 2006.

[112] S. A. Holmes, G. Klein, and D. W. Murray, "An o(n^2) square root unscented kalman filter for visual simultaneous localization and mapping," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, pp. 1251–1263, July 2009.

[113] H. Hu and X. Huang, "SINS/CNS/GPS integrated navigation algorithm based on UKF," *Journal of Systems Engineering and Electronics*, vol. 21, no. 1, p. 102, 2010.

[114] E. Hernández Bes, P. Ridao Rodríguez, D. Ribas Romagos, and J. Batlle i Grabulosa, "Msispic: A probabilistic scan matching algorithm using a mechanical scanned imaging sonar," *Journal of physical agents*, vol. 3, no. 1, pp. 3–11, 2009.

[115] C. S. Lee, D. Clark, and J. Salvi, "SLAM with single cluster PHD filters," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2096–2101, 2012.

[116] R. Mahler, "Multitarget bayes filtering via first-order multitarget moments," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 4, pp. 1152–1178, 2003.

[117] B.-N. Vo, S. Singh, and A. Doucet, "Sequential monte carlo methods for multitarget filtering with random finite sets," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 4, pp. 1224–1245, 2005.

[118] B. Vo and W. Ma, "The gaussian mixture probability hypothesis density filter," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4091–4104, 2006.

[119] J. Mullane, B.-N. Vo, and M. Adams, "Rao-blackwellised PHD SLAM," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 5410–5416, 2010.

[120] H. Abdella, D. Lane, and F. Maurelli, "Sonar based mapping using PHD filter," in *Proceedings of the IEEE/MTS OCEANS Conference and Exhibition*, pp. 1–7, 2014.