CHANGE DETECTION OF SEA FLOOR ENVIRONMENT USING SIDE SCAN SONAR DATA FOR ONLINE SIMULTANEOUS LOCALIZATION AND MAPPING ON AUTONOMOUS UNDERWATER VEHICLES

by

Timothy Pohajdak

Submitted in partial fulfilment of the requirements
for the degree of Master of Applied Science

at

Dalhousie University
Halifax, Nova Scotia
April 2016

I dedicate this thesis to my parents, Bill and Gloria Pohajdak, who have always been a source of help, inspiration, and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Autonomous underwater vehicles (AUVs) are frequently used to survey sea-floor environments using side-scan sonar technology. A simultaneous localization and mapping (SLAM) algorithm can be used with side-scan sonar data gathered during surveying to bound the possible error in AUV position estimate, and increase overall position accuracy, using only information already gathered during the survey mission.

One problem in using SLAM to improve localization is that data from a preliminary or route survey on the sea floor may be inaccurate due to changes in the sea bed or merely be differently detected due to different side-scan sonar surveying patterns or equipment.

This thesis' focus is an integrated on-board SLAM system using automated target recognition system to extract objects for SLAM data association, data association algorithms for MLOs (joint compatibility program), and finally change detection on the SLAM results to determine if new objects have been introduced to the sea floor.

LIST OF ABBREVIATIONS USED

ADCP	Acoustic Doppler current profiler
ASW	anti-submarine warfare
ATR	automated target recognition
AUV	autonomous underwater vehicle
C3I	command, control, communications, and intelligence
CD	change detection
CSAIL	Computer Science and Artificial Intelligence Laboratory
DRDC	Defence Research and Development Canada
DVL	Doppler velocity logger
EKF	extended Kalman filter
EM	electromagnetic
GPS	Global Positioning System
iSAM	incremental smoothing and mapping
ISR	intelligence, surveillance, and reconnaissance
IvP	Interval Programming
KF	Kalman filter
MCM	mine countermeasures
MIT	Massachusetts Institute of Technology
MLO	mine-like object
MOOS	Mission Oriented Operating Suite
NMCM	naval mine countermeasures
PF	Particle filter
ROV	remotely operated vehicle
SLAM	simultaneous localization and mapping
UAV	unmanned aerial vehicle
USV	unmanned surface vehicle
UUV	unmanned underwater vehicle
WHOI	Wood's Hole Oceanographic Institute

ACKNOWLEDGEMENTS

I would like to acknowledge the support and assistance of my direct supervisor, Dr. Mae Seto. I would also like to acknowledge the assistance of co-supervisor Dr. Ya-Jun Pan, and the other members of my committee Dr. Darrel Doman and Dr. Thomas Trappenberg. I would like to thank DRDC Canada staff and personnel for all their assistance and help and access to facilities, especially the technicians Sean Spears, Owen Shuttleworth, and Tim Murphy. I would like to thank the US Office of Naval Research (ONR) for financial support for this work. Finally, I would like to thank my parents Bill and Gloria Pohajdak for their continual support during this project.

CHAPTER 1 INTRODUCTION

1.1. AUV BACKGROUND

There are two broad classes of Unmanned Underwater Vehicles (UUVs). The first are the free-swimming Autonomous Underwater Vehicles (AUVs), while the second are the tethered Remotely Operated Vehicles (ROVs). AUVs, as their name suggests, are robots capable of traversing underwater environments (similar to submarines) and performing scripted tasks to make measurements more-or-less independently of human operators. Due to the difficult underwater environment for acoustic communications and navigations, AUVs must be self-sufficient, and possess on-board power, navigation, and control systems. AUVs, available commercially in a variety of different configurations, sizes and payload sensors, are capable of independent missions once deployed from a ship or dock. This independence sharply contrasts against Remotely Operated Vehicles (ROVs) that require communications, control and power from the surface operator, via a tether from the surface. ROVs have human operators at the top side of the tether [1]. AUVs are more flexible – they are not tethered, so their range is limited only by the on-board energy they carry, and they require less top-side support for their scripted missions. However, AUVs require on-board autonomy (controls programming) to operate in ever more complex environments and missions. The underwater environment has limited range and bandwidth for acoustic communication with human operators, in complex missions like surveying beneath Arctic ice [2]. Scripted missions are very much limited and unable to adapt to changes in the environment, robot, or mission. This thesis' research focusses on programming the AUV to generate maps and navigation plans from on-board sensor data and is thus capable of adaptive, autonomous navigation.

AUVs are used for civilian, military, and commercial applications. As AUV autonomy advances, the missions AUVs can be applied to, becomes more complex. Missions that AUVs are presently tasked to, or development is underway for, include:

- Mapping underwater bathymetry as in Canada's UNCLOS efforts to map the Canadian Arctic continental shelf to substantiate sovereignty claims under-ice [3].

Also, research in studying ocean chemistry and biology are underway by the Canadian Excellence Research Chair at Dalhousie University [8].

- Surface vessel or submarine surveillance/observation (intelligence, surveillance and reconnaissance or ISR) or anti-submarine warfare (ASW) [10].

- Naval mine countermeasures (NMCM) involves the detection, classification, identification and neutralization of mines targeting manned ships and submarines. It is desirable to use unmanned ROVs or AUVs for these tasks [10]. The work of this thesis is applied to mine countermeasures.

- Command, control, communications, and intelligence (C3I) relay platforms for capital ships. AUVs can act as sensor platforms, and have on-board acoustic modems or in-air radios to communicate with ships and submarines [10] either underway or on the surface, respectively.

- Searches for shipwrecks and downed aircraft as in the successful localization and recovery of the Titanic. Downed aircraft are recovered, among other reasons, to determine the causes of failures [4].

- Hull inspection of ships to assess the condition of the wetted hull for corrosion and to search for attached objects [5].

In underwater mapping tasks such as bathymetry surveys, NMCM, and downed aircraft searches, AUVs have notable differences from previously used methods where side-scan sonars are towed by manned ships. AUVs:

- are slower (typical speeds ~ 3-4 knots, or 1.5-2 m/s) compared to minimum ships' speeds of 5^{+} m/s or more;

- can reduce the exposure of manned ships to potential naval mines by allowing the ships to serve as re-supply / pickup / relay points for the AUVs that perform the actual NMCM surveys [10];

- may have greater inaccuracy in determining their submerged location compared to a ship towed sonar's which is bound by the tow cable scope. The towed sonar's range and range and bearing is relative to the ship's, which has GPS;

- do not facilitate human operators analyzing their acquired side-scan sonar data until the AUV is recovered. A towed side scan sonar relays the imagery in near real-time to the ship via the tow cable. This is addressed through the emerging capability, implemented in this thesis, to analyze the side-scan sonar imagery in-situ on-board the AUV [6].

AUV teams can mitigate their disadvantages over previously proven NMCM methods and other survey missions. Multiple AUVs controlled by cooperative algorithms allow for more efficient surveys compared to an individual AUV. Further, AUVs can use one or two AUVs periodically surfacing, or one or more unmanned surface vehicles (USVs) to obtain a GPS fix that can then be "shared" amongst the AUVs in a cooperative manner [7]. This 'accuracy' is propagated to submerged AUVs through time-of-flight measurements of acoustic modem pings and local sound velocity profile to determine range and bearing from the AUVs/USVs with GPS fixes. Techniques which depend on distance between the sensor and robot include short baseline, long baseline, and ultrashort baseline [9]. Another useful feature of working with multiple AUVs is the ability to transmit locations of targets or mine-like objects from one AUV to another over underwater acoustic modem. This allows one AUV to make a preliminary survey of a large area at lower resolution to identify targets and then have another AUV, with a higher resolution sonar and directed mission to more closely inspect the targets [6].

The primary focus of the research is to use AUVs to map seabed objects in an area with little or no *a priori* information. This is achieved by integrating on-board Automated Target Recognition (ATR) into Simultaneous Localization and Mapping (SLAM) algorithms. This research is also relevant to multiple AUVs surveying large areas. This thesis' work allows AUVs to better localize themselves which lays the ground work for AUVs to localize other AUVs as in long baseline navigation to establish accurate relative

locations underwater. The work of this thesis enhances localization of underwater assets in the global reference frame.

The AUVs used at Defence R&D Canada (DRDC) for research are ISE Explorers and OceanServer Technology Ivers (formerly Iver2s, now Iver3s). The former are large, working class, long endurance AUVs used successfully on multi-week missions to map the Canadian continental shelf [3]. The latter are small man portable AUVs on the lighter end of the four man portable classes identified by the US Navy [10]. The OceanServer Ltd. Iver2 AUVs have custom autonomy software tools developed and implemented by DRDC and are used at DRDC as testbeds for validating experimental algorithms. In the control and sensor respects they are very similar to the large, working class AUVs but but lack the power, range, or sea-keeping – which is not as critical as an autonomy testbed. The Iver2 AUV (5.8in dia, Figure 1.1) was the primary testbed for the research and is described next.

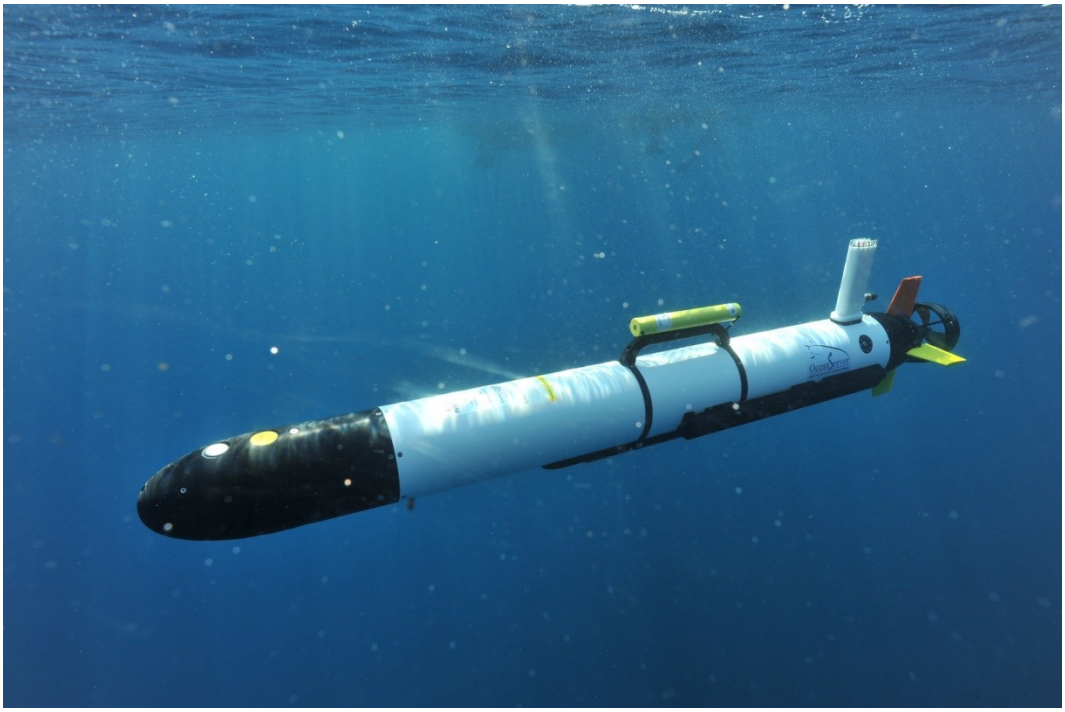


Figure 1.1: Iver2 AUV. Source: Oceanserver, http://www.iver-auv.com/photogallery.html.

The Iver2 AUVs have four rear control surfaces, which are actively controlled hydrofoils, that maintain the AUV depth/altitude, pitch, roll, and heading control. A single shrouded

propeller provides the forward thrust. The Iver2s are small, light, and deployable by one or two operators with no specialized equipment like the crane needed to launch and recover larger AUVs. The Iver2's are easy to handle, inexpensive, and have an open architecture on the payload computer which makes them excellent testbeds for developing control algorithms. Iver2 AUVs are available in a variety of acoustic and non-acoustic sensor configurations. Possible sensors include side-scan sonars, Doppler velocity logs (DVL), acoustic Doppler current profilers (ADCP), acoustic modems, control computers, payload computers, inertial navigation systems, etc. The specific Iver2 AUV used for this research has an Imagenex Yellowfin 3-frequency (260 kHz/ 330 kHz / 800 kHz) side-scan sonar, 4-beam Doppler velocity log, compass, acoustic modem, etc. Iver2 AUVs are rated for operating to 100 meters depth. The battery endurance is about 14 hours at 2.5 knots in a 'best case scenario', though the battery endurance decreases with increased duty cycle of the sonars, underwater modems, Doppler velocity log, on-board computers, and other hardware.

A distinction is made here between the terms feature, object, target, and landmark. A feature is a generic structure or shape. In imagery it can be a simple collection of pixels that has not been given an identity as an entity or object. Examples of objects include a rock, a mine, or an underwater docking station. The automated target detection tools (discussed later) convert features in imagery into objects. A target usually refers to an object with a specific geometry in mind. Object and target is sometimes used interchangeably in the literature. A landmark is an object (or target) that could be a milestone in reference to a map or mapping as in SLAM.

This particular system is limited in that it possesses no sophisticated inertial measurement unit. In an AUV with a really excellent IMU, accuracy loss of location would be decreased and for tests on the scale of this project SLAM would not help substantially as during these tests the AUV would not build up much error to be detected. However, even with an IMU, without references to fixed locations, positional error still grows unboundedly over time and with sufficient time underwater when returning to the same area the AUV would benefit from SLAM in theory.

1.2. INTRODUCTION TO UNDERWATER SLAM

Simultaneous localization and mapping (SLAM) is a solution to the probabilistic robotic control problem – robotic navigation and localization in unfamiliar surroundings with noisy sensors and actuators. The SLAM problem, in the context of AUVs, is to enter an environment where the AUV has no a priori information and to map objects of interest in that environment and to localize itself within the map at the same time. The objects of interest in this thesis are targets that appear mine-like in geometry and referred to as mine-like objects (MLO). For the AUV-based solution a variety of sensors are needed to navigate and localize the robot and to detect and localize the targets).

The on-board inertial navigation system, compass, Doppler velocity log (measures speed over ground), and propeller rotations are inputs towards a refined dead-reckoned navigation and localization solution for the AUV. When possible, the AUV reduces its cumulated dead-reckoned position error by coming to the surface for a GPS calibration (or fix). The SLAM algorithm uses iterative sensor measurements and models, of which the extended Kalman filter (EKF) is an example of the latter. SLAM methods have been used for AUVs in ocean environments to navigate [11] but, for the underwater environment this has been problematic and does not work well, repeatedly.

The side-scan sonar is used to acquire imagery of the seabed. From this imagery objects can be extracted using the on-board DRDC automated target recognition (ATR) software tools. The navigation and localization stream of the AUV is synchronized in time with the side-scan sonar imagery stream in order to geo-reference the targets. This means the targets' locations are only as good as the AUVs'.

SLAM provides a solution for mapping with minimal risings for a GPS calibration by re-observing or re-visiting targets (not necessarily from the same mission) that were observed when their position error was small. This could be for a target that was observed soon after a GPS or other position calibration. For this to occur the SLAM 'loop closure' or data association, where recently observed targets are associated or matched with previously observed targets, has to be performed accurately.

In the probabilistic formulation of SLAM the uncertainties and noise in the robot measurements of attitudes, altitudes, and position; control actuation, and target geo-referencing are explicitly captured through probability distribution functions. Information on the inaccuracies in the robot's measurements and actuations are determined from previous in-water experience with the system.

Commonly used SLAM algorithms are based on two methodologies – EKFs and Particle Filters. In both cases they have mathematical properties that converge the map and robot pose to the system's collective accuracy given data sets of estimated AUV and measured (or calculated from measurement) target positions [13]. The SLAM approaches have their trade-offs for computational efficiency in terms of number of calculations to produce updates, accuracy, implementation difficulty, adaptability to complex environments and fitness to an application [14] [19][21]. These can be difficult to judge. For example, map quality has competing requirements like robustness to initial inaccurate data [22], feature detection, loop closure [23], etc. Usually computationally efficient solutions yield less accuracy than less conservative solutions [19], such as PowerSLAM [24], a highly efficient computationally SLAM technique. Consequently, SLAM solutions are diverse and vary in their implementation.

For example, a large region to be mapped can be broken into smaller regions that are individually less computationally intense to analyze as with the ATLAS SLAM framework [25]. Its drawback is that it is not suited for SLAM problems with known data associations and Gaussian error distributions. Cumulative SLAM methods that do not require re-computing the entire map to achieve loop closure are consequently, computationally efficient though potentially less accurate [26]. Sparsification methods reduce the number of non-zero elements in the covariance matrices and thus reduce computational requirements [19] [27] [28]. This allows multiplications and inversions of large matrices in SLAM updates to compute faster and reduces memory and subsequent read-write delays. Methods that were once too computationally intractable to be practical have become less so over time. Moore's Law (suggested in 1965) postulated that computer performance will double roughly every eighteen months [29]. This has broadly held true as manufacturers have used Moore's law as targets for their design of future

CPUs. Presently, incremental progress in computer performance is from more processor cores which SLAM implementations like iSAM2 (discussed later) exploit through parallelizing tasks.

SLAM involves landmarks the robot measures range and bearing to. If recently observed landmarks are correctly associated with previously observed ones, the robot gains confidence about the landmarks' relative layout from the multiple observations. This confident association can be used to deduce landmark positions relative to the AUV [19]. However, if multiple landmarks are close together and similar in appearance this can cause incorrect associations [30]. If the landmarks have other features that can be sensed (e.g. colors or bar codes) then this can be used to jointly associate with landmark location with the aim to reduce incorrect associations. In some of the simpler AUV localization tests by the DRDC Iver2, an acoustic modem serves as a stationary landmark broadcasting its known GPS position. Synchronized clocks that measure the time-of-flight between between sending and receiving acoustic signals on assets with modems can be used to measure the underwater range between them. SLAM algorithms that use visual data from cameras [32], sonar data in artificially constructed environments [33], or environments with specifically placed sonar reflectors [31] have been implemented with AUVs.

1.3. EKF STATE ESTIMATION

With state estimation noisy data is extracted from measurements of robot localization or landmark ranges and bearings (with an AUV) to describe system state. For an AUV executing SLAM, this state is relevant to map building and estimates of its position within that map. Kalman Filters are an example of a state estimation algorithm. The most basic Kalman Filters require that the measurement and motion (or state transition) model's uncertainty be described by a normal distribution – which may or may not be accurate. If these conditions are not exactly correct, but the robot and sensor performance could be approximated in this manner, then the system could be modeled using Kalman filters. The extended Kalman Filter uses the first term in the Taylor-series expansion to describe a nonlinear function as a linear function tangent to the nonlinear function over the region of interest. The EKF approximation is efficiently computed even for multi-

dimensional systems. However, if the linear approximation is invalid because the mean of the nonlinear function is discontinuous or the system is quite nonlinear, this will cause the EKF to fail and the map and AUV/targets positions will not converge. EKFs can be part of a larger SLAM implementation with GPS access and proprioceptive navigation sensors to periodically reduce error [35].

1.4. PARTICLE FILTER STATE ESTIMATION

Particle filters (Pf) use Monte Carlo point mass distributions to capture probability density functions for nonlinear sensors, actuation, or robot motion where the uncertainty is not normally distributed. Monte Carlo localization uses particle filters for robot localization. The robot state space is represented by particles which are each a hypothesis of the robot state. Each particle is weighted to reflect the likelihood of the robot being in that state. The (Markov) recursive assumption and the iteration between predicted (motion) and measurement update states in Kalman filters is maintaine here. As particle filters are not parametric they can more flexibly support multiple state hypotheses unlike Kalman Filters.

Initially, particles are distributed uniformly over the robot's state space as it has no information on its location so every state is equally likely. The measurement update occurs when the robot makes a measurement. In resampling, the particles are resampled based on recursive Bayesian estimation. Particles (state hypotheses) which support the measurement are weighted more so less likely hypotheses are removed from the state space and more likely hypotheses are generated. In the motion update, the particle distribution changes its density as the robot moves. Particles move in the robot state space to support hypotheses of where the robot might be and noise is added. The particle filter solution converges to dense clouds of hypotheses/particles that are denser during measurement and diffuse when the robot moves.

An example of the particle filter's merit is with environments that yield similar sensor measurements (e.g. an office building with similar offices). Compared to Kalman Filters, particle filters underperform where both could be used (discussed in 2.1). The computational overhead of calculating likelihoods for many particles (~ order of

hundreds) is greater than the Kalman Filter solution which is a single particle. However, Kalman Filters are not applicable to all SLAM scenarios and the advantage of PFs is with scenarios where the Gaussain approximation is invalid [36]. While the number of PF particles is on the order of hundreds it can be scaled – more particles yield better localization and navigation but require more computations. A robot that is moving and updating its range and bearing measurement every ten seconds needs more particles than if it updated every one tenth of a second. Large kinematic movements cumulate more uncertainty which manifests in the state space as greater dispersion dispersion. Linearly increasing the number of particles in a particle filter yields diminishing returns in accuracy and precison and requires linearly more processing cycles. A typical implementation of such a case is FastSLAM [19]. The next sub-section describes the particular SLAM implementation used in this work.

1.5. iSAM: Incremental Smoothing and Mapping

iSAM (incremental smoothing and mapping) is a SLAM implementations developed by Georgia Tech and MIT researchers [37]. iSAM is efficient and can execute on-line. On-line means the SLAM solution is calculated in near real-time during the robot mission rather than in post-processing after the mission. The iSAM implementation is efficient as it is only a few mathematical steps to execute, sparse matrices are used to reduce the computational burden, and only those portions of the covariance matrices that change is recalculated. Research and development has been on-going since iSAM's introduction [37]. Additional computational efficiency is achieved by recovering covariance information, of information related to localization, from a square root information matrix that is already maintained by iSAM. This recovered data is the relative uncertainties between the position of a single AUV pose and a landmark's position. This is used for the data association which helps to solve the data association problem [39]. iSAM was designed to work in sparse environments with as few as one landmark. Marine environments are often either sparse or cluttered with objects. iSAM uses both proprioceptive measurements such as DVL, compass, modem acoustic time-of-flight measurements, as well as inclinometers and internally logged propeller rotations to localize AUVs [40].

The latest iSAM version, iSAM2, has as its innovation the use of a new data structure, the Bayes trees – a set of interconnected nodes that represent the underlying Bayes network of probability densities [41]. iSAM2 achieves smoothing and remapping of the map with the Bayes tree. The nodes are calculated from Gaussian elimination on matrices of the measurement and motion data. The Bayes tree facilitates insight through data processing with well-known linear algebra operations. The Bayes tree allows iSAM2 to be more computationally efficient than the original iSAM. In particular, iSAM requires a batch updating that re-linearizes the model periodically which causes a delay (proportional to the amount of data) in the processing. This iSAM batch update occurs once every 100 cycles of the move-sense-update cycle. The Bayes tree facilitates fluid re-linearization so a periodic batch process to relinearize the model is not needed. Consequently, this eliminates the spikes in time to process data that appear during the batch job – further increasing iSAM2 efficiency.

iSAM2 produces accurate maps compared to other SLAM algorithms with improved on-line accuracy. On occasion, iSAM will cumulatively increase error after each recalculation caused by new data during the time period between two of the batch step relinearizations mentioned earlier. This accuracy loss is ameliorated when the batch step occurs and relinearizes the model, so the final map accuracies, or accuracies of maps generated by post-processing of data, are similar. By comparison, iSAM2 does not have this so iSAM2's average accuracy is better than that of iSAM. The iSAM2 algorithm can recover marginal covariances as iSAM does, which as stated earlier facillitates better data association. Finally, iSAM2 can exploit parallelization in multi-core processor architectures better than earlier SLAM implementations for faster computing [42].

SLAM allows an AUV to determine its position with a certain confidence and its surroundings relative to that position, even from an initial state of ignorance. That knowledge may be used to plan paths dynamically, or maps may be built from data observed while following pre-set paths. This is discussed in Section 3, on path planning.

1.6. SLAM: IMPACT OF ENVIRONMENT

SLAM solutions are diverse. Not surprisingly 2D implementations are ill-suited to 3D applications. Similarly, SLAM methods developed in one environment are not easily applicable to another.

Compared to terrestrial SLAM, which is a well-studied problem [12], there are differences for SLAM in the underwater environment. The closest other field of robotic SLAM to underwater is aerial SLAM as both have the same degrees-of-freedom for the motion. An underwater (or aerial) environment means the robot is free to manoeuvre in three dimensions rather than two, so there is greater complexity to describe the environment for navigational purposes. In underwater (or aerial) SLAM there are also three degrees of rotational freedom (pitch, roll, and yaw) that are used to navigate through environments rather than the single yaw in conventional two-dimensional land-based SLAM. This makes for a more complex robot state.

In other respects, SLAM for aerial robots differs from underwater ones. The issues involved with a robot making measurements above a terrestrial terrain is not entirely unlike those of an AUV measuring above the seabed. Notably different is that in underwater SLAM communications (between other robots or operators) are limited in range and bandwidth. This impacts the ability for the navigating AUV to localize by communicating with other AUVs, beacons, or modems that may know their positions well.

Another notable difference from aerial robot SLAM is that there is no global positioning system to aid in AUV localization or navigation. When the seabed is deficient of distinctive geo-referenced landmarks to navigate against, deployed geo-referenced sonar targets, acoustic modems or beacons could be used. However, there is expense and logistics in deploying, localizing, and recovering these devices. AUVs can also reduce their cumulated dead-reckoned position error by surfacing for GPS fix. This however, compromises mission efficiency and especially so in deep water. For localization, AUVs could range off an unmanned surface vehicle that broadcasts its known GPS position to the underwater AUVs (via an underwater modem on-board the unmanned surface vehicles). The AUVs use one- or two-way time-of-flight acoustic pings with a

synchronized clock to determine their range from the unmanned surface vehicle to localize themselves. Yet another method is to use landmarks that are identified by geo-referenced side-scan sonar images, These landmarks can be natural features like distinct rocks on the sea floor or man-made debris. Identifying landmarks from side-scan sonar images is not without its challenges especially as it impacts data association. Different insonification aspects, AUV altitudes, and environmental conditions can cause the same landmarks to appear different on subsequent side-scan sonar scans. These challenges are discussed more in Section 5.

The underwater environment changes over time due to tidal action, currents, climate change, movement of ships and activity by man through removing or inserting landmarks. If these features are incorporated as landmarks in a SLAM, their changes have to be identified, tracked, or at least noted in the SLAM map. This is currently accomplished by using a stationary sensor SLAM update followed by tracking moving targets [19]. This dynamically changing environment makes the underwater SLAM, an already difficult problem, even more so. This tracking of changes is referred to as change detection. This thesis addresses underwater SLAM with the aim of mapping mine-like objects in a *dynamic* environment so change detection is of interest. Consequently, the next sub-section briefly reviews change detection as relevant to the work here.

1.7. CHANGE DETECTION

Change detection (CD) is, in most general terms, "the process of identifying differences in the state of an object or phenomenon by observing it at different times" [44]. The goal of CD is to identify where parts of a surveyed area have significantly altered, despite noise and background changes, while identifying changes of significance. In the underwater environment CD has been successful with underwater video [45][46] but, sonar proves more difficult. The challenge is to interpret the sonar data and find feature correspondences between sonar surveys as well as identify where change occurred. CD is used in data analysis but only in a limited fashion with sonar and even less so for side-scan sonar. This is due to the difficulty in identifying features in sonar imagery across surveys. Side-scan surveys of the same area performed during the same hour can appear different due to the sonar's sensitivity to acoustic returns based on ensonification angle (or aspect), range from the AUV, underwater acoustic ambient, susceptibility to noise

corruption, and AUV attitude. Accurate geo-referencing of underwater locations across side-scan sonar surveys is vital for CD. One application of underwater CD is to determine whether a new object has been added or, if a previously observed object has been removed or altered as in naval mine counter-measures. There are also other applications as in searches for downed aircraft or deliberately laid AUV docking or charging stations. If the same geo-referenced objects consistently appear across sonar surveys (i.e. there is no change at those locations) then those objects could be used for local robot localization as in terrain-based navigation.

This comparison of features across side-scan sonar surveys could be implemented as a simple differencing of geo-referenced images. Another approach is to identify regions where significant changes are detected (compared to the background) and examining those regions to extract significant features with which to compare against in other surveys to determine the level of correspondence across temporally separated images [47]. Recent DRDC efforts [48][49] use this approach for mine-like object detection. Specifically, this involves identifying the feature in localized regions of the side-scan sonar image. The feature geometry is identified by extracting 'patches' of bright sonar returns and shadows to define that pixel collection as a feature and to determine attributes like geo-referenced location, length, width, and height above the sea bed. Then, the CD applies hypothesis testing to determine the likelihood of whether the feature is a change at that location or not. CD is applied in this thesis to determine exactly that – to identify features that are consistent, have altered, or are new detections across survey that could be separated by months (or years). In this manner, the CD is referred to as persistent. In the long term, persistant change detection is of interest. This thesis applies it near the end. Given all the on-board processing described so far, the next sub-section describes the on-board processors and the data flow for the SLAM.

1.8. ON-BOARD PROCESSING

The Iver2 AUV has two on-board networked computers, both Intel Atom CPUs, the front seat or Original Equipment Manfacturer (OEM) computer and the 'back seat' or payload computer. The front seat computer, running Windows XP Embedded, is where scripted mission plans are created, stored and executed. This computer is provided by the OEM, OceanServer Technology Ltd. Side-scan sonar data arrives on this computer from the sensor as data streamed to a file. The 'back seat' computer, which is user-configured is networked to the front seat computer, and used for more deliberative high level planning/learning /etc like in the SLAM. In this case, Ubuntu 10.04 is installed for compatibility with the original Iver2 AUV. The front seat computer

executes the missions, implements the navigation solution, and administers the fault diagnostics and sonar data-gathering, It also performs the low level inner loop control of the AUV, for example to maintain yaw, pitch, roll, altitude/depth, and speed over ground.

As mentioned earlier, the back seat computer is for higher-level deliberation tasks like mission-planning, decision-making, and functions like interpretation and parsing of side-scan sonar data, sophisticated navigation like SLAM, etc.
This front seat/back seat configuration is a standard setup for Iver2 AUVs [20] and has its benefits. Isolation of the front seat system from the back seat means that, if the advanced and often prototype-level software on the back seat fails or causes total system failure the front seat system is isolated from it and continues to run and could implement some recovery. As well, the processing speed and memory available on the back seat computer is not tied to the front seat, and therefore back seat programs can be optimized without being bottlenecked by the front seat for disk read-write availability or memory as sonar data is streamed to the front seat.

There are a few drawbacks to this approach – duplication of hardware increases both hardware costs and increases power consumption compared to a single proccessor. Also, programming and using the Iver2 is more complex as multiple systems must work in concert to transfer data via e.g. FTP between the two computers, rather than having all data on the same hard drive. However, the net effect, particularly in the safety of trialing experimental software on an isolated system as opposed to the main navigational and control system, outweighs the disadvantages.

This thesis runs software on both CPUs. A flow chart of the software is shown in Figure 1.2. It illustrates the data flowing through the system from sensors, the layers of analysis on the AUV, and how the data passes between components.

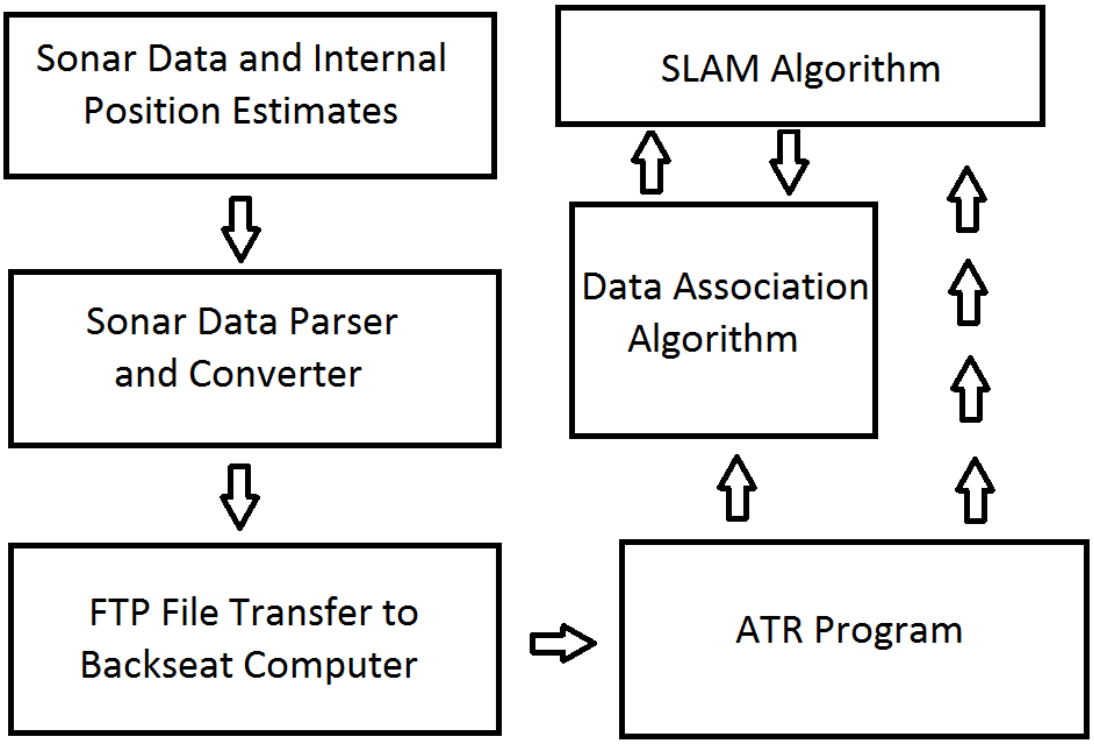


Figure 1.2: Data flow to implement on-board SLAM. The portions on the left side of the diagram execute on the front seat computer and the portions on the right execute on the back seat computer. The FTP program transfers sonar data from frontseat to the backseat.

Initially, the AUV used for this project's demonstration and testing gathers side-scan sonar data in the .872 format for side-scan sonar files (Imagenex, OEM for the Yellowfin side-scan sonar). At each side-scan sonar ping there is an updated position estimate based on the predicted Iver2 motions through water from its navigation solution which uses a DVL, 3-axis compass, and propeller rotation measurements in a proprietary dead-reckoning algorithm. This telemetry information is likewise collated into *.872* Yellowfin files on the front seat.

On the front seat computer, these files are read and imported as binary data then exported as *.xtf* files which is then read by the on-board automated target recognition (ATR) tool. The ATR processes the side-scan imagery using filters to detect landmarks from the background of the sea floor based on geometric similarity to known MLOs. The landmark position is then associated by a data association algorithm against previously observed landmark positions (with their uncertainties) in the SLAM algorithm. Then,

this processed data is input to the SLAM algorithm to produce refined location estimates for the Iver2 AUV as well as the detected landmarks. This is performed on-line during the mission. Processing is on the front and back seats and are initialized during the mission's setup phase. Upon initialization a specific folder on the front seat computer is periodically polled for a size change indicating a new sonar data (*.xtf*) files to upload to the back seat from the front seat. The parsing and analysis of a sonar data file starts after it is detected in the polled directory. For the cases tested, the parsing and analysis were complete prior to the survey for the next leg. This prevents overlap leading to delay of the sonar data processing build-up over multiple legs of a mission. This means that even with the current embedded computer the system can run in near-real-time as opposed to entirely in post-processing.

1.9. PARAMETER TUNING AND ENVIRONMENTAL EFFECTS

In general, this project was created with a number of adjustable parameters that were later tuned and refined to create the best possible results for the environment of the Bedford Basin, the specific hardware (especially AUV and Sonar type used), and the type of mission carried out (5m altitude above sea bed sweeps). If these particulars changed – if, for example, surveying a much deeper area, using a different AUV and Sonar, etc – then a substantial recalibration would have to be manually done by a researcher.

The major parameters that were created and tuned for this program are subdivided into two distinct things: ATR Parameters, and SLAM parameters.

The ATR parameters refer specifically to parameters that the ATR program uses to analyze a sonar data stream and identify targets within it. The key thing to optimize for is low numbers of false positives (e.g. times the system identifies noise as a minelike object / landmark) while maximizing the number of true landmarks. In decreasing order of importance to optimize, the ATR parameters include the target object size along-track and across track (parallel to and perpendicular to the AUV's motion), the target object height above sea floor, the detection threshold parameter, the smoothing sizes and background sizes used in processing, and the minimum and maximum ranges objects can

be found in relative to the AUV path. These parameters were part of a configuration file that is fed into the ATR program to adjust it's targeting parameters every time it is run.

For the internal SLAM parameters, these parameters were part of another configuration file that was pulled into the SLAM program, and adjust the mathematical model used to create the map. These parameters include a covariance matrix of the AUV sensors & AUV kinematics; time in seconds between AUV poses in the map, and several data association parameters that adjust e.g. maximum distances between proposed associated landmarks & likelihood requirements for data association. These parameters were optimized for avoiding incorrect associations (primarily) while making as many correct associations as possible (secondarily). The covariance matrices were estimated based on best estimates of sensor performance and accuracy of the Iver2 motion, and revised to increase the accuracy of the output maps.

To do parameter tuning, there were several stages. Initially, a sonar data stream from the Bedford Basin from a different AUV sonar type but similar mission (survey at 5 m altitude above sea floor) was used to do SLAM testing and establish initial values for parameters. This was using a Minegarden data set made available from DRDC. These values were used for initial tests in the Bedford basin. After experiments collected their own data set using the actual sonar and AUV for this project, the parameters were retuned for later tests using actual data collected in situ using actual equipment on the AUVs.

In general, to prevent overfitting and incorrect tuning, the shortest mission available at any particular time was used as 'validation' data – parameters were tuned using the other mission data sets, and then the data that was not used to tune the parameters was tested against to validate that it would work on unknown data. Some of the ATR parameters are likely environmentally sensitive and dependent on the location used for testing. The other parameters are more hardware dependent, particularly on the sonar and the Iver2 navigation software/hardware performance as a gestalt.

In actual useage of a AUV SLAM system for military / commercial / etc use, ideally a large number of datasets will be gathered and appropriate parameters for the various environments and sonar types that are going to be used, and parameters created for each

that the AUV system could be made to switch between appropriately. This sort of large-scale data-gathering endeavor is both beyond the scope of this project and also not valuable on the Iver2 AUVS which are a research testbed rather than a practical commercial or military AUV for large surveys.

The AUV system as a whole is sensitive to ATR parameters, and it is very possible to create false and erroneous maps that give misleading information (from false positive target recognition) or maps that are entirely useless due to sparsity of type of object (from not having appropriate matches). The SLAM parameters are likely to be perfectly fine for all Iver2 AUVs with these sonars, but should be remade for use with other hardware/software; Only a few might be recalibrated for different mission types – in particular, the number of poses on a multi-day mission would likely be excessive to calculate with if a poses is taken as happening every 10 seconds as in this research.

The methods used to gather the parameter tuning data, and gather data to assess the project as a whole, are discussed below.

1.10. IN WATER TESTING PROCEDURE

In water testing was done in the Bedford Basin using the DRDC Acoustic Calibration Barge as a base, and launching and recovering the AUV from a rib. In general, the basin has a wide variety of sea floor types with some areas rocky, others sandy, etc. This variety allowed researchers to assess the problems and limitations imposed by different sea terrains. The basin also had some human made debris such as boxes that served as already emplaced minelike objects for the survey.

The testing was done in several short missions that were taking place in different parts of the Bedford Basin, but all between the Acoustic Calibration Barge and the nearest coast, due to wanting to avoid difficulties with local traffic in the Basin and to simplify recovery in case of vehicle failure (by making it much more likely to beach on some sandy area near the Barge). For additional safety and AUV loss prevention, the AUV was programmed to surface at each ‘leg’ of a sonar survey - after completing a duration of surveying in a straight line, it would surface and update it’s GPS, turn, and resubmerge for the next leg of a sonar survey.

There were 4 different missions run in somewhat different locations in the basin, and they were run repeatedly over the course of several days that were spread out over a few months, during which time the basin itself changed and moved around. The missions were programmed to run on the front seat computer of the Iver2 while the other computer carried out SLAM but did NOT feed this information to the front computer in any respect – this meant that the mission was entirely governed by OceanServer's own onboard software to follow waypoints as best it could with no updating position based on SLAM estimates. This was done for three reasons:

- reduced risk to the AUV in event of SLAM failure
- simpler programming and engineering
- data processing limitations meant that updates to vehicle position estimates for relatively small runs in shallow waters like the Basin tests would only finish processing after the vehicle had already surfaced and gotten a GPS fix anyways.

As every time the AUV completed a transect, it surfaced for a GPS fix and update, this allowed a primary statistic used to assess performance was a metric of error parameterized as ε.

ε was defined as follows:

$$\varepsilon = \frac{|\theta_{GPS} - \theta_{slam}|}{|\theta_{GPS} - \theta_{dr}|}$$

Where θ_{GPS} is the GPS-verified true position when the vehicle has surfaced, θ_{slam} is the SLAM estimate of position when surfacing, and θ_{dr} is the dead-reckoning estimate of position when surfacing, and the differences are taken as the absolute distance between the positions as a length. This ratio being less than one means the positional estimate accuracy was increased, while it being one means it's error is unchanged. If it was to be greater than one, it would mean the SLAM had actively reduced positional accuracy – a failure the system was carefully tuned to avoid.

1.11. THESIS ORGANIZATION

The remainder of the thesis is organized as per Table 1.

Table 1: Thesis Organization – Contents by Chapter Number.

Chapter Number	Contents
Chapter 2	Discussion of SLAM algorithm selection rationale and iSAM internal details.
Chapter 3	Discussion of ATR program selection details and rationale. Discussion of ATR parameter tuning and optimization.
Chapter 4	Discussion of data association method selection and tuning and optimization
Chapter 5	Testing and results. Demonstration of functionality. Demonstration of processing in real time.
Chapter 6	Contributions
Chapter 7	Conclusions and recommendations for future work.
Appendices	Sample code and sample raw data and more detailed results.

CHAPTER 2 SLAM ALGORITHM

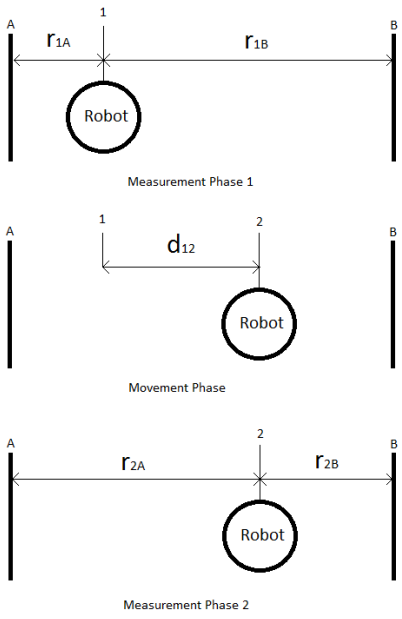
The objective of SLAM is for "an autonomous vehicle to start in an unknown location in an unknown environment and, using relative observations only, incrementally build a map of [a] world and to compute simultaneously a bounded estimate of the vehicle location." [15]. Sensor observations are separated by motion of the AUV. This means that the autonomous vehicle whether on land, under the sea, in the air or any environment, must measure from its surroundings ranges to landmarks (e.g. walls) and use these measurements, along with measurements of its motion such as speed and pose, to build a map of the environment in which the vehicle localizes itself. The map is probabilistic if it captures the uncertainty in the measurements, actuation, and subsequent localization. The robot uses this information to navigate within the environment to achieve its mapping goals. The constructed map may be in the global reference frame (like GPS is) or relative to the robot's initial location or other origin [16]. Under certain circumstances SLAM assumes a Gaussian distribution [17] for the uncertainty in sensor measurements and motion actuations. Such an assumption makes it mathematically convenient to compute a solution though in some cases it can be a limitation. The general SLAM problem is complex and once a system is capable of such a solution it is simpler to find a solution for systems with fewer restrictions, e.g. navigating when a map or GPS is available for the robot position [18]. There are a simpler SLAM problems that are more tractable than the general solution which encompasses nearly all real-world applications that AUVs might require. [10]

2.1 iSAM: Incremental Smoothing and Mapping

iSAM and iSAM2 are leading SLAM algorithm implementations initially developed at Georgia Tech [37] with further development at MIT's Computer Science and Artificial Intelligence Laboratory (CSAIL) [38]. These SLAM algorithms execute efficiently and on-line. The latter meaning as the robot moves, the SLAM estimates are calculated at each time step rather than in post-processing at the end of the robot's trajectory. iSAM is computationally efficient because it transforms matrices of estimated positions into sparse matrices, works with the information matricies in triangular form, and uses back solving and forward solving to decompose the model of the robot poses and landmark

positions. Consequently, subsequent recalculations from new observations are performed only in those parts of the matrices that changed. In this way, it is possible to retain the majority of the benefit from previous computational effort in solving each new time step. Other SLAM methods solve the entire robot trajectory at every time step. This re-ordering and transformation of the matricies relies heavily on Suitesparse's column approximate minimum degree (COLAMD) and other refactoring techniques [34]. They allow the overall performance of iSAM to not deteriorate as rapidly as other approaches with arbitrarily increasing AUV poses and landmarks – COLAMD in particular creates factored representations that are numerically sparse, where as many entries as possible are zero, and thus the matrices are more efficient to solve. There are other iSAM strengths such as its ability to add and remove associations between landmark observations in response to updates in data association. Research and development on iSAM has been on-going since its introduction [38]. One part of iSAM was not perfectly suited to other parts of the SLAM problem – the 'square root' representation of the information matrix that iSAM relies upon obscures the covariance information (information about the strength of knowledge on the relative locations of various points) that may be used for data association. This covariance information is more explicitly available in other SLAM algorithms and so could be incorporated into the data association (Chapter 4). This was remedied in iSAM by designing mathematical methods to efficiently recover covariance information [39] from a square root information matrix that is maintained by iSAM of the landmark and AUV pose information. This recovered data, information regarding the relative uncertainties between the position of a vehicle pose and particular landmark's position, helps to solve the data association problem, and is used when the vehicle achieves loop closure (i.e. reacquires a previously observed landmark) [39]. iSAM works well with few navigational landmarks which is an advantage as marine environments can be either sparse or very cluttered with indistinct landmarks. iSAM has been validated to work with as few as one landmark. As with other localization methods, iSAM integrates measurements from Doppler velocity logs, compass, propeller thrust estimates and rotation rates with micromodem time-of-flight range measurements, to a modem of known position, to localize the AUV [40].

Fundamentally, iSAM uses a square root information matrix as the basis of its process. In a SLAM approach following the methods of for example, FastSLAM, the main process that is computed repeatedly is a representation of landmark positions relative to one another in the form of an information matrix. This is composed of the sensor measurement covariances and an associated measurement vector. These factorized representations are made up by adding each measurement's individual matrix of covariances into a complete matrix representation. For example, Figure 2.1 below shows a robot with two distances to measure in a single-vector world. Phase 1 measures those distances. In Phase 2, it moves in its 1-D world. In Phase 3, it measures once again from its new location. The two robot poses are x_1 and x_2, the landmarks are L_A and L_B, and the measurements are r while the distance moved is d and finally error estimated in movements and measurements are e.



Measurement Phase 1 Matrix:

$$\begin{bmatrix} r_{1A}+r_{1B} & -r_{1A} & -r_{1B} \\ -r_{1A} & r_{1A} & 0 \\ -r_{1B} & 0 & r_{1B} \end{bmatrix} \begin{bmatrix} x_1 \\ L_A \\ L_B \end{bmatrix} = \begin{bmatrix} e_{r_{1A}}+e_{r_{1B}} \\ -e_{r_{1A}} \\ -e_{r_{1B}} \end{bmatrix}$$

Movement Phase Matrix:

$$\begin{bmatrix} d_{12} & -d_{12} \\ -d_{12} & d_{12} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} e_{x_{12}} \\ -e_{x_{12}} \end{bmatrix}$$

Measurement Phase 2 Matrix:

$$\begin{bmatrix} r_{2A}+r_{2B} & -r_{2A} & -r_{2B} \\ -r_{2A} & r_{2A} & 0 \\ -r_{2B} & 0 & r_{2B} \end{bmatrix} \begin{bmatrix} x_2 \\ L_A \\ L_B \end{bmatrix} = \begin{bmatrix} e_{r_{2A}}+e_{r_{2B}} \\ -e_{r_{2A}} \\ -e_{r_{2B}} \end{bmatrix}$$

Figure 2.1: Information matrix example – parts of a simple information matrix

Combined Matrix for all 3 Phases of Measure – Move - Measure:

$$\zeta X = error\ vector = \begin{bmatrix} d_{12} + r_{1A} + r_{1B} & -d_{12} & -r_{1A} & -r_{1B} \\ -d_{12} & d_{12} + r_{2A} & -r_{2A} & -r_{2B} \\ -r_{1A} & -r_{2A} & r_{1A} & 0 \\ -r_{1B} & -r_{2B} & 0 & r_{1B} + r_{2B} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ l_A \\ l_B \end{bmatrix} = \begin{bmatrix} e_{r_{1A}} + e_{r_{1B}} + e_{x_{12}} \\ e_{r_{2A}} + e_{r_{2B}} - e_{x_{12}} \\ -e_{r_{1A}} - e_{r_{2A}} \\ -e_{r_{1B}} - e_{r_{2B}} \end{bmatrix} \quad (1)$$

The combination of the information from movements and measurements is, as shown above in Eq.(1), represented as an information matrix ζ and an error matrix in an equation that represents the robot distances measured and distances moved in a compact form as shown in Eq.(1). The best solution, X, to Eq.(1) is the least-squares solution to the system of equations represented in matrix form. This is computed by inverting the square matrix, ζ and multiplying it by the error vector. The inverse of the information matrix ζ multiplied by the error matrix solves the least-square problem and minimizes the error, giving the most likely solution to a set of sensor measurements where the 'solution' consists of ranges of all landmarks and poses to the initial pose.

However, the matrix inversion step in the most straightforward solution is computationally expensive and scales with matrix dimension (i.e. number of landmarks and poses – though sparsification can reduce matrix density somewhat). This means that as the robot acquires more poses and landmarks, the matrix inversion requires more processing (each update step requires O(M log K) operations where there are M particles with K landmarks). As an amelioration of this problem, iSAM's unique insight is Q-R factorization of the square information matrix, and keeping the data represented in a Q-R factorized form even as more measurements are added. This is achieved through a Givens Rotations of the matrix – an efficient and numerically stable QR decomposition method that allows the information matrix to be broken down. Givens rotations are more computationally stable compared to for e.g. a Householder transformation with the same effect.

A Givens Rotation is a simple procedure to rotate a matrix about a plane and keep all information and relationships in the matrix while reducing one element of the matrix to zero. When applied sequentially and repeatedly it reduces multiple elements to zero, by multiplying the matrix by a transforming Givens rotation matrix is an identity matrix except for *i*th and *j*th rows and columns, where the *ii* and *jj* rows have cos(θ) and the *ij* and *ji* rows have sin(θ) and -sin(θ). This rotation angle and the row/column numbers *i* and

j are chosen to zero a specific element each time. This creates a matrix that is triangular and easier to solve since it can be back and forward solved, efficiently. This representation is the 'square root information filter' representation as it is comprised of a QR decomposition into an upper and lower triangular QR factorization that reflects *r* about the matrix main diagonal – the two 'square roots' of the information matrix. This representation can be stored and updated efficiently, as before, and remain fairly sparse by periodic variable re-ordering. This grows less quickly in computational intractability relative to the size of the poses and landmarks compared to other methods that use the same sort of matrix-based approach to SLAM. It is bound by $O(n \log(n))$ or $O(n)$. The simple inversion has a computational cost in n poses and landmarks between $O(n^3)$ and $O(n^{2.373})$, depending on how the inversion is implemented. This means the computationally expensive parts of the SLAM are performed only on the measurements' equations as measurements are added rather than to the entire matrix.

Additionally what is not shown (well) by the simple example here is in real situations the matrix will be fairly sparse, i.e. the number of non-zero entries will be less then the number of zero entries. This is because not all landmarks are observable from all robot poses. Each pose is only related to the previous and next pose it is measured relative to (as well as landmarks the robot observes at those positions). Consequently, the square matrix of landmarks and poses will necessarily have more zeros than non-zeros where landmarks are observable from only some poses (or aspects). The zero elements, matrix sparsity, and symmetry allow for more optimized computations to solve matricies repeatedly as new measurements or poses are added to the matrix.

The approach ISAM uses for the matrix inversion is to find a square root information matrix. If the covariance matrix is A and the vector of measurements as B, while the resulting information we desire is X, the general case is:

$$X = A^{-1}B.$$

However, instead of finding this directly through matrix inversion every time, this can be converted by Q-R factorization to

$$X = (Q * R)^{-1} * B.$$

Where R is an upper triangular matrix and Q is an orthogonal matrix. Thus, the best estimate of X is:

$$X = R^{-1}(Q^T B).$$

Since matrix R is upper triangular, the matix inversion can be calculated by back substitution, which is numerically efficient.

The solution will be found multiple times and as the matrix expands through addition of new columns and rows (poses and landmarks) the factorized QR representation of the matrix also expands. However, the Givens rotations used for the QR factorization for previous steps can be used again for all but the new columns/rows of information and the new matrix can be merged with the already rotated one, thereby saving on computation effort. The matrix is recalculated from scratch, occasionally, by default in ISAM every 100 new total pose/landmark information sets; this recalculation is to reorder variables to increase the sparsity of the matrix and thereby speed up computation performance for subsequent processing. The entire matrix may be reordered to make it sparser (COLumn Approximate Minimum Degree [COLAMD] matrix reorganization algorithm of Larrimore and Davis [97]), which reduces the overall computational requirements of the algorithm from the larger number of zeros in the factorized matrix representation after LU decomposition. This allows for faster computation of the least squares solution when using more-or-less the same matrix repeatedly with incremental new data from a pose and landmark measurement.

This square root information filter (SRIF) approach is suited for SLAM implementation on embedded processors with landmarks that do not number in the hundreds or thousands for every pose – true for underwater systems using sidescan sonar measurements. SRIF works on-line, is computationally efficient, and produces pose and map updates every time step. At the same time, the iSAM representation of the information matrix as a QR-decomposed square root information matrix allows one to add and remove data associations and still calculate final estimates for all landmarks and poses efficiently. The

iSAM algorithm grows in computational time without bounds, linearly, in the number of landmarks and poses processed. However, this happens mostly with larger numbers of poses that observe all the landmarks, eventually. As the matrix density increases, iSAM's advantage decreases. Some SLAM implementations require the robots stay within an area and see the same landmarks thousands of times as they move around in a small area repeatedly. However, for underwater MLO surveying, targets in an area are unlikely to be re-visited thousands of times. The targets are more likely surveyed twice or have a few survey paths overlap for the purposes of re-acquiring the target. This means the increasing matrix density as the mission progresses and the consequent iSAM slowdown will not have a significant impact on its performance for MLO surveys.

iSAM needs landmarks associated with robot poses and other landmarks. This landmark information is extracted by processing side-scan sonar imagery and AUV telemetry using on-board automated target recognition (ATR) tools, as described in the next section.

The motion data estimate is extracted from the Iver2 logs created as the mission runs. These log files are created and stored as semicolon separated values programs with a header as follows:

Table 2: Log File Data

Latitude	Dist To Next (m)
Longitude	Next Speed (kn)
Time	Vehicle Speed (kn)
Date	Motor Speed CMD
Number of Sats	Next Heading
GPS Speed (Kn)	Next Long
GPS True Heading	Next Lat
GPS Magnetic Variation	Next Depth (m)
HDOP (horizontal dilution of precision, GPS accuracy)	Depth Goal (m)
C Magnetic Heading	Vehicle State
C True Heading	Error State
Pitch Angle	Distance to Track (m)
Roll Angle	Fin Pitch R
C Inside Temp (c)	Fin Pitch L
DFS Depth (m)	Pitch Goal
DTB Height (m)	Fin Yaw T
Total Water Column (m)	Fin Yaw B
Batt Percent	Yaw Goal
Power Watts	Fin Roll

Watt-Hours	DVL-Depth (m)
Batt Volts	DVL -Altitude (m)
Batt Ampers	DVL -Water Column (m)
Batt State	DVL-FixType
Time to Empty	DVL-FixQuality
Current Step	

These data are all stored to the log file every second. The most important data extracted from the log file is latitude, longitude, and heading data for every second.

TARGET RECOGNITION FOR SIDESCAN SONAR

This section describes how objects or targets for iSAM (Chapter 2) are 'detected' or extracted from side-scan sonar imagery. There are inherent issues with using side-scan sonar imagery for SLAM. The most important is that the appearance of a feature/object/target to a side-scan sonar image is a function of:

1) sonar insonification aspect (or angle). The side-scan sonar insonification of an object creates highlights and shadows that represent the object. An object with shadow is illustrated in Figure 2.2 below. The length of the shadow is a function of ensonfication angle, range and altitude to the AUV. Objects close to the AUV will have shorter sonar shadows than ones further away much as terrestrial structures at noon have smaller light shadows than near twilight or dawn.

2) sonar range to the object which affects intensity/brightness of a sonar return in addition to the object shadow length.

3) ambient conditions (noise underwater, obscuring objects like fish). Noise can be detected by the ATR as objects, and adds occasional bright lines to sonar imagery that can be removed by filtering.

Figure 2.2: Side-scan sonar object (center) with shadow (right).

With side-scan sonar imagery, size, shape, and depth information can be extracted but not color as with optical imagery. Optical imagery at depth is not of much use as it is difficult to carry an intense enough light source on-board, have it work at large range, and not be obscured by underwater scattering or absorption. The resolution of side-scan sonar images can be as good or better then optical imagery at higher insonification frequencies.

Therefore, the estimated location of the object is a key attribute – it is simple to work with, and parametize, and during a survey the 'true' location of an object remains relatively constant (i.e. no changes in appearance).

Several object extraction tools were examined in this work.

1) A peak detector made from open source software (OpenCV) [51] finds spikes and intensity drops towards the outer edge around a detected object – essentially looking for bright patches of variable pixel size that are joined together and accompanied by a dark patch away from the sonar ensonification direction. This approach applied to side-scan sonar imagery picks up errors from out-of-range glitches due to noise in the sonar data stream and has difficulty determining an accurate background intensity to reference dark and light levels against. This had to be recomputed for every sub-section examined in the imagery. On faster processors (AMD A6-3420) than is present on the Iver2 (1.6 GHz Atom processor) this approach required 4+ minutes to analyze a 3-minute sonar data stream which is marginal for on-line processing.

2) SIFT/SURF feature extraction [52] proved unreliable across known objects when those objects were viewed multiple times from different aspects. The unique object shadows and shapes that depend on proximity and angle usually were not geometrically similar enough across multiple data sets for correct data association based on SIFT/SURF feature extraction. The subsea features were identified as new objects. It also proved sensitive to noise in the data stream which it detected as new SIFT/SURF features. It was however, computationally efficient. The large number of false positive from interpreting noise as new SIFT/SURF features was the most problematic part – noise as lines of bright returns were found regularly. Conversely, often features observed from the context of absences and irregularities in specific locations against a more regular background were not found.

3) DRDC proprietary ATR was created to detect mine-like objects on the sea floor from side-scan sonar imagery. It was designed to be robust and efficient, and invariant as possible to insonification (observations) from different aspects. The processing is efficient. A single run of 3+ minutes of side-scan sonar imagery required twenty to

thirty seconds to process on-line on the Iver2 embedded processor. This solution also proved more robust to noise in the data as well.

The DRDC proprietary ATR tool was chosen for this project. However, one of the other methods, may also work if the algorithms used are more computationally efficient and better filtering and noise-reduction applied to the side-scan sonar imagery.

The DRDC ATR is configurable to bottom type (e.g. sand, gravel, mud, etc.) or sensitivity by adjusting parameters. In some circumstances (where the sea floor was cluttered or exceptionally sparse) this was adjusting the size threshold of the object to detect. This was effective in reducing the number of objects detected in a cluttered area and increasing the detected objects in sparse areas. Figure 2.3, shows the results of applying ATR at the default 1-meter size threshold compared to other methods.

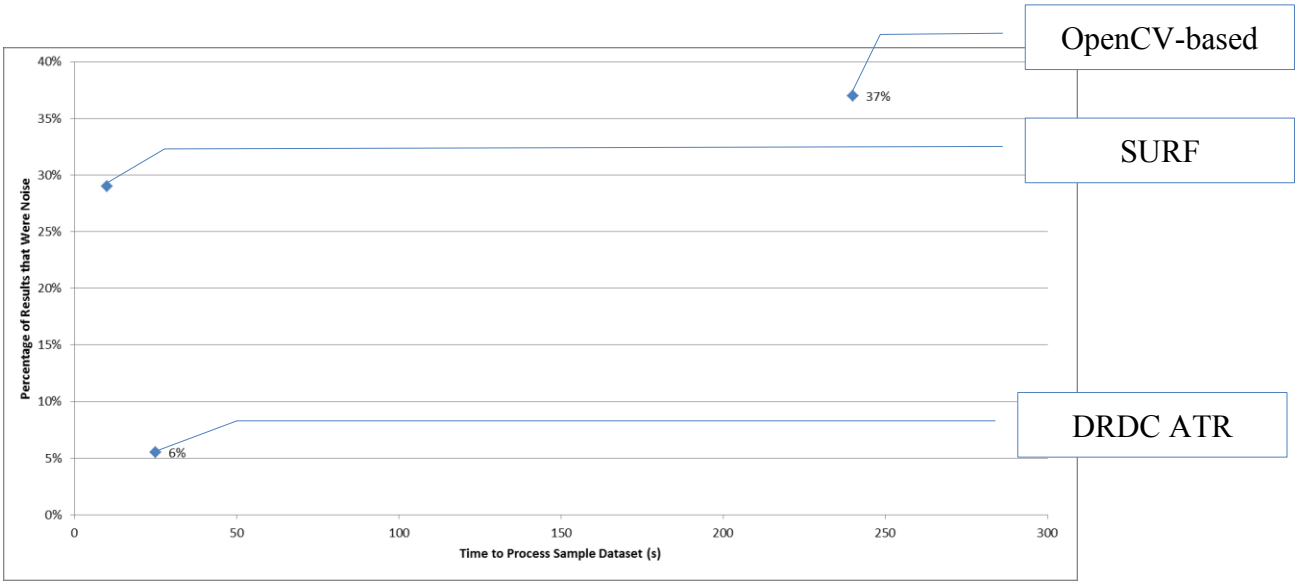


Figure 2.3: Comparison of side-scan sonar imagery extraction tools. Results are averaged performance across three transects of duration 192 seconds, 190 seconds, and 194 seconds.

By comparison, Figure 3.3, shows how adjusting the size threshold of detected object affects the detection rates over a sea floor area of Bedford Basin (Nova Scotia).

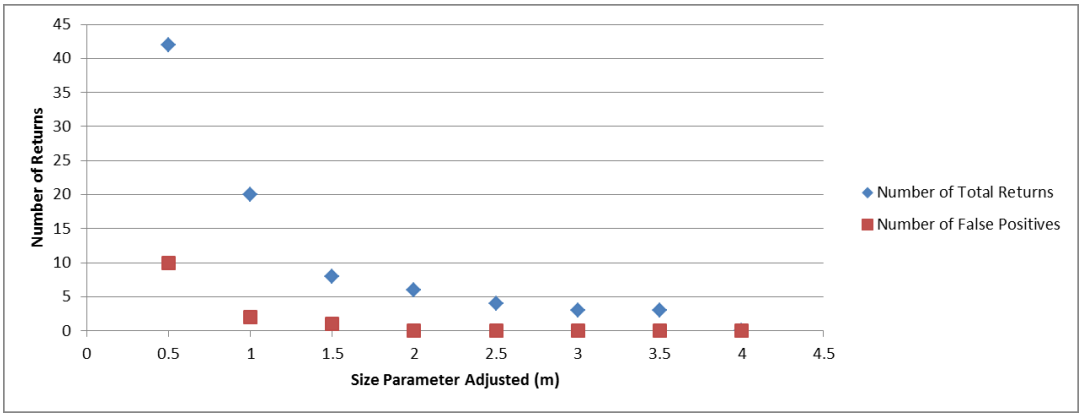


Figure 2.4: Effect of ATR size parameter threshold on detection rate.

There was no measureable time difference to process of the ATR based on this detection characteristic. The possibility of noise being detected increased dramatically at sub-1m size threshold, so those were not used. Such small thresholds are not representative of the objects of interest. The ATR was tuned to run at a default size parameter of two meters as a minimum with the parameter decreasing to 1m if thre were no detections. The size of objects to detect was greater than 1 meter.

One has to keep in mind that the SLAM objective in this work is to map targets of a certain size. However, distinct objects of any size are valuable for the data association and navigation SLAM objectives. This should be kept in mind when the detection size threshold is adjusted.

Once objects/targets are detected in the side-scan sonar imagery, the SLAM data association is the next problem.

CHAPTER 3 DATA ASSOCIATION

Suppose the ATR initially identifies an object at location *X*, when the robot is at pose *P*. If the system then later identifies an object at pose ~*X* from a vantage *P*, how should the system determine whether this is a new object *X2* or the previously observed object *X*? Several data association algorithms were examined to answer such questions:

1) Nearest Neighbour;

2) Individual Compatibility, and

3) Joint Compatibility.

The first, Nearest Neighbour, considers which of the object locations previously observed is closest (in the Euclidean sense) in range to the recently observed object, and associates them unless the range threshold is exceeded, in which case the detection is considered a newly observed object.

Individual compatibility association assigns the new object's association based on its range against a probabilistic position distribution that describes the location of previously observed local objects. Whichever one has the recently observed object best falling into its position distribution (in the Mahalanobis sense [53]), is associated, unless all are very unlikely in which case none are associated.

The final association, joint compatibility, assesses not only the association of a single object, but the cumulative probabilistic chances of each possible set of associations for multiple objects, simultaneously. Fundamentally, it may be less likely for a set of old objects *A*, *B*, *C* and new object set *X*, *Y*, and *Z*, that *X* matches *C* than that *X* matches *B*. However, if it is more compatible that *X* matches *B* AND *Y* matches *C* AND *Z* matches *A*, then this incorrect association can be omitted.

The most complex of the above, joint probability, is also the most complex to implement. However, with a limited field of objects the computation time to associate the object after each run is manageable – 30 seconds instead of ~100 milliseconds for the simplest association (nearest neighbor). This time is of less concern as the SLAM calculation

takes 2-4 minutes depending on the sonar run length (this 30 second average includes time for iSAM to make the covariance matrix accessible – a significant time not incurred for other data association methods). For more details on computation performance, see Chapter Five.

The worse joint compatibility data association performance in terms of speed is not a significant problem compared to simpler algorithms, because it is part of a whole ensemble that takes an order of magnitude longer anyways. However, the more sophisticated data association is better for situations where the ATR result is ambiguous, the sonar imagery is noisy, or otherwise difficult to work with. If the situation had distinct targets that were clear and less noisy (not usually), then it would be better to use a simpler and faster algorithm.

The actual data association performance in terms of accuracy (low frequency of incorrect association hypotheses) is difficult to evaluate since many subsea MLOs appears similar. The primary method to assess accuracy is twofold. The first is to examine the proposed associations by comparing side-scan sonar imagery by eye visually. Examples of visually distinct objects are Figure 3.1 and Figure 3.2 – the former consists of a large (~2 m) boulder greater than almost all other matches in size and with a distinct shadow, while the latter is a sharp-cornered object (may anthropic in origin).

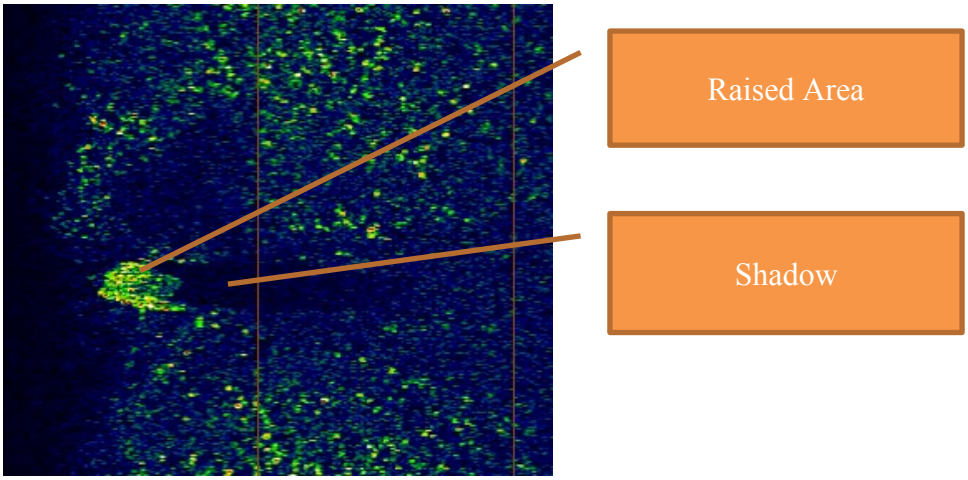


Figure 3.1: First example of visually distinct object from side-scan sonar imagery

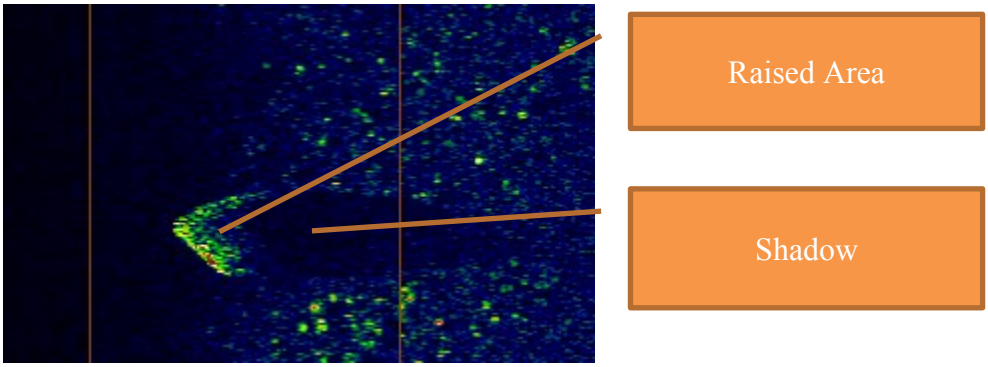


Figure 3.2: Second example of visually distinct objects from side-scan sonar imagery

The second way to assess data association accuracy is to examine how well the SLAM increased the AUV's position estimate. This is discussed in Chapter 5. The results of visual inspection alone were significant enough to the assessment that the time requirements of getting covariance matrix information from ISAM and associating it was considered necessary and beneficial.

Figure 4.3 is an example of data associations compared from visual examination.

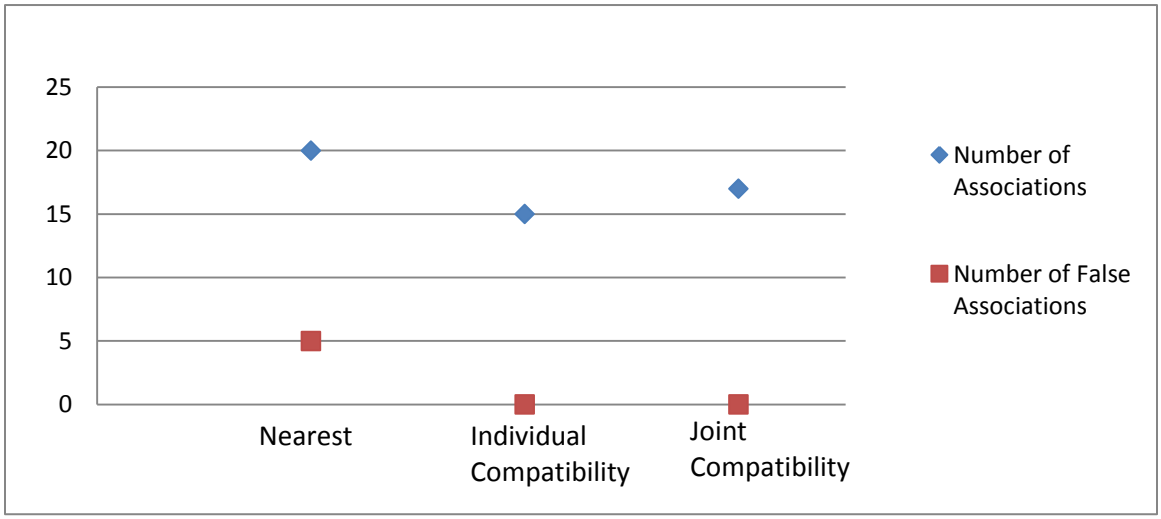


Figure 3.3: Data association accuracy assessment. False associations fall with more sophisticated data association algorithms. Correct associations increased by joint compatibility algorithm. Nearest Neighbour produced undesireable false positive associations.

As shown in Figure 3.3, the performance in number of total associations was similar for all data association methods considered – possibly a result of having a large dependency

on perceived object locations for all tested examples. However, the large number of false associations with Nearest Neighbour makes it undesirable as false associations unknowingly distort the map, especially when there are few correct associations. It is not possible to restrict a Nearest Neighbour algorithm to output a large number of correct associations without incorrect associations.

The difference in joint vs individual compatibility was largely the latter having slightly better results in the rate of data association but at a ~50% performance penalty in processing time (from ~1 to ~1.5 seconds). However, the primary computational burden is in extracting the covariance data information from iSAM and in both individual and joint compability this is an identical duration.

Having studied all the subsystems in Figure 1.2, the next task is to assess their computation time and localization performance when integrated.

CHAPTER 4 RESULTS

The results overall showed that the sub-systems discussed in Chapters 2-4 are tenable for increasing localization accuracy and detecting moved/newly placed/altered MLOs automatically on sea floor bottoms. In areas where detections were sparse (1 detection or less per transect) it was unsatisfactory. This was compensated, to some extent, by decreasing the object detection size threshold in the configurable ATR parameters. However, such a measure is limited by sensor quality and noise. When the object detection size threhold in the ATR is too small the false detection rates increase and thus reduces the value of the detections.

5.1. TEST RUN CASES

There were four in-water survey missions to gather data and test the system in this thesis. These runs are labelled missions A, B, C, and D.

Mission A is a 40-minute survey that ran northeast-southwest as shown in Figure 4.1.

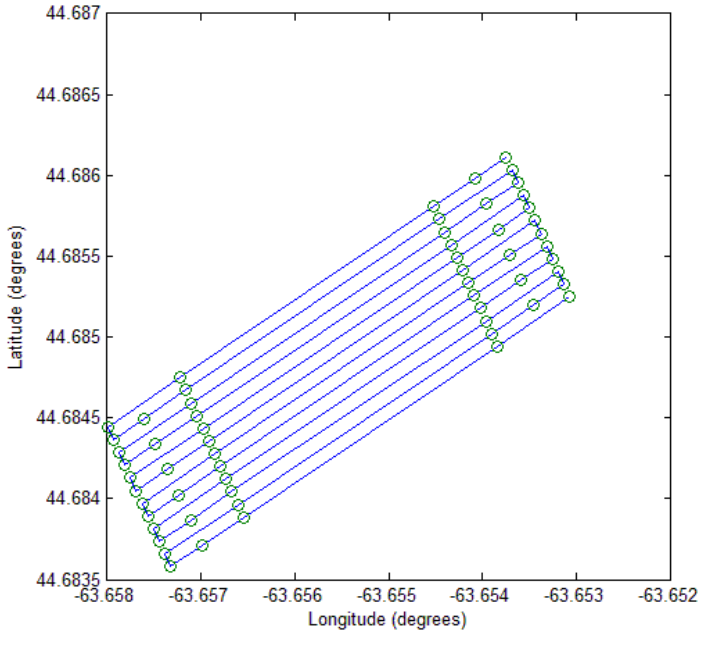


Figure 4.1: Planned route (waypoints) in Mission A. Transects have dive/rise segments to obtain a GPS fix after turn segments at the ends.

In Figure 4.1 there are idealized straight parallel lines as these are GPS coordinates for the AUV waypoints to be traversed – the exact execution is not as precise. The set of outer and inner GPS points in the transect show how at the end of a transect the robot surfaces for a GPS calibration, then dives between two GPS points going in the forward direction, and begins surveying at depth until it reaches the the final submerged waypoint.

Illustrated in Figures 5.2 - 5.4 are the three other mission areas surveyed. These are shown for comparative purposes – Missions A, B, and D are all in different locations within Beford Basin, at different angles and with some variety in the length of run. In Mission C, the area was surveyed at a 90° aspects to assess the effect of sonar ensonification direction on target perception.

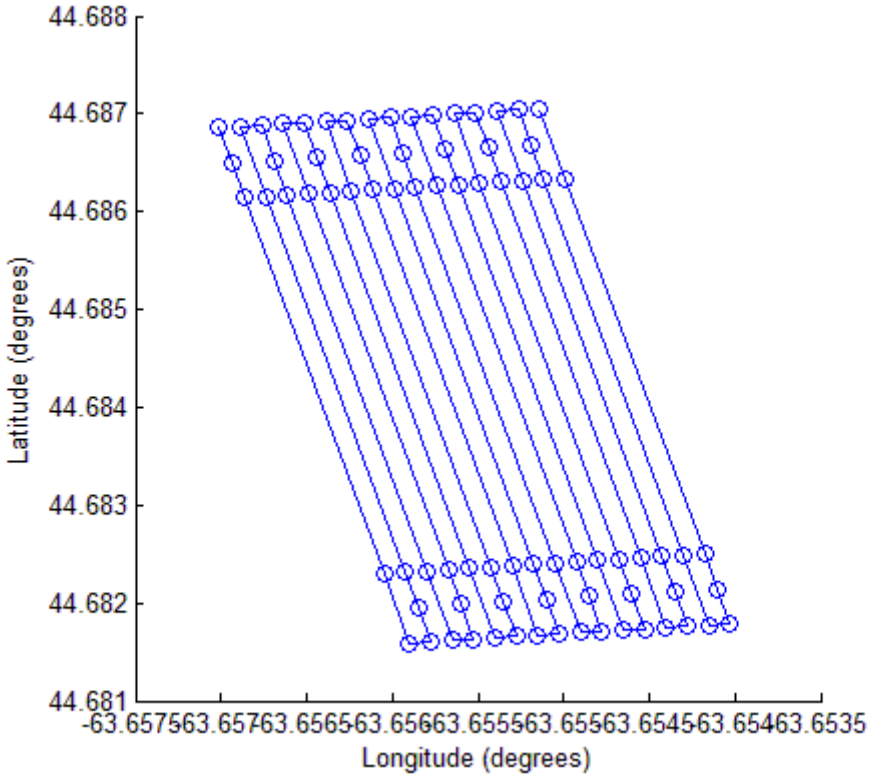


Figure 4.2: Mission B planned route. Transects are at different location and length than A.

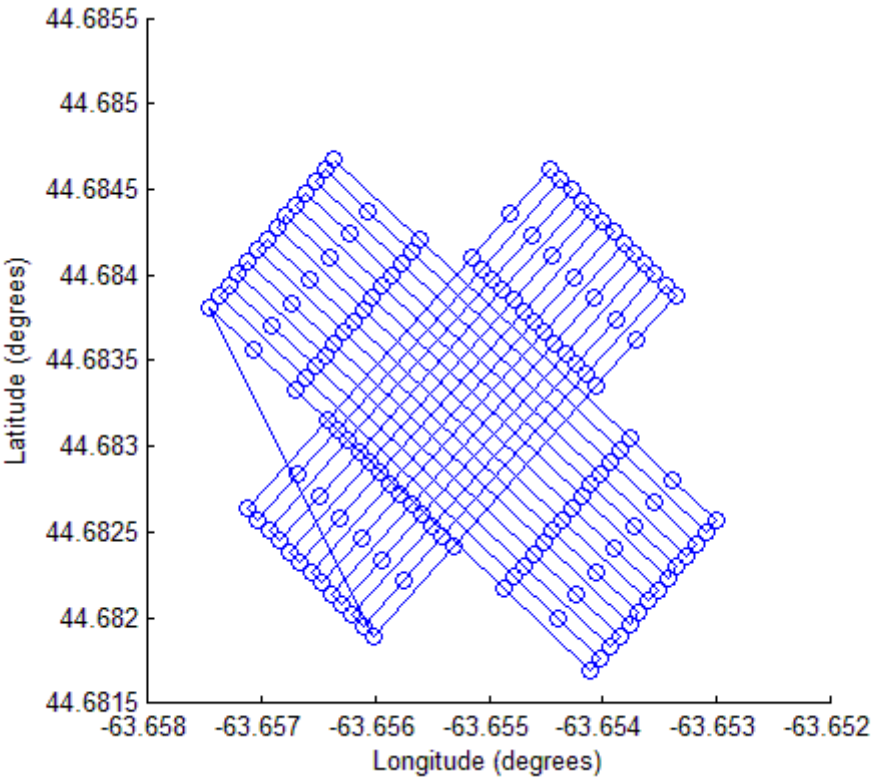


Figure 4.3: Mission C planned route. Transects are at right angle aspects and substantial area is ensonified from two directions.

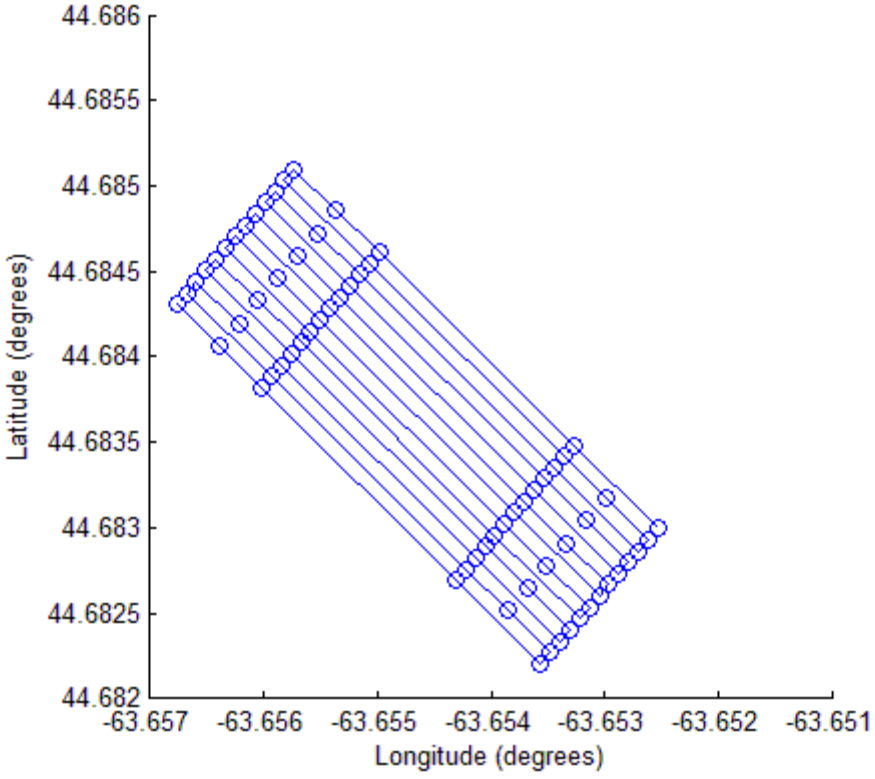


Figure 4.4: Mission D planned route. Similar to other tests but again different location / aspect.

The missions were set-up as standard underwater side-scan sonar surveys in different areas. The only different mission was C, which examined data association methods and ATR in the case of perpendicular sonar surveys of the same area. Otherwise, the goal of multiple missions was to have different sea floor bottom types to reduce overfitting.

Side-scan sonar imagery and associated AUV navigation telemetry data was gathered. This was deliberately tested over three days in October 2013, a day in December 2013, and two days in January 2014 for change detection purposes. The runs are summarized in Table 5.1

Table 3: Summary of when missions in Fig. 5.1 – 5.4 were performed

day runs performed	*mission test runs (by letter)*
October 22, 2013	C, D
October 23, 2013	A, C
October 24, 2013	A, B
December 12, 2013	B, D
January 20, 2014	B, D
January 21, 2014	C, A

The preceding Figures of ideal paths for the AUV to take do not match up to the actual Iver2's path achieved. This is due to the Iver2's limited ability to dead-reckon without inertial navigation, steer in the presence of water currents at low speed, and inability to dive instantly and perfectly to depth. As a result, the actual path for a reasonably good run appears as Figure 4.5.

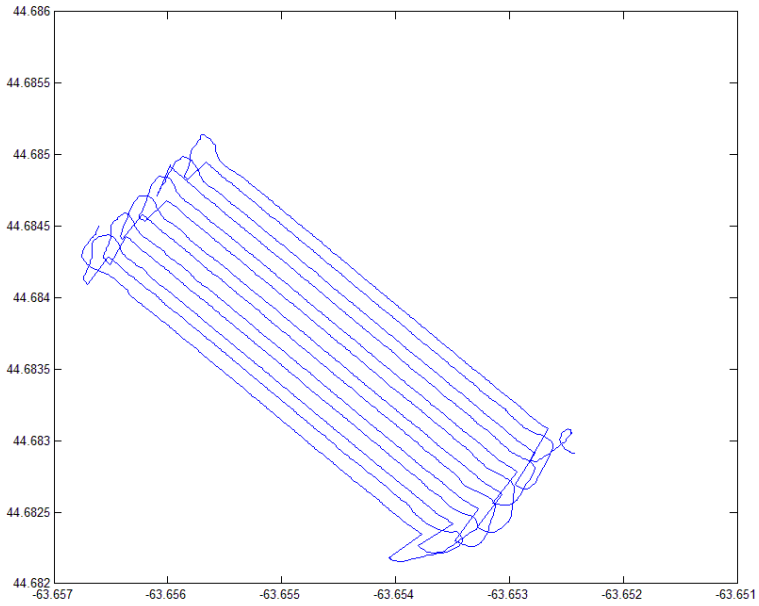


Figure 4.5: Mission D dead-reckoned AUV positions underwater and on the surface (at the end of transects), The underwater dead-reckoning solution does not integrate water current measurements so the underwater paths are straighter than they actually are. GPS measurements occur at the end of each transect.

At the ends of each transect, where it is possible to compare the GPS and dead-reckoned AUV position upon surfacing, there is a substantial deviation between actual GPS location and estimated dead-reckoned location. This causes the robot's position to 'jump' between the last estimated dead-reckoned location and the new GPS measured one.

When the side-scan sonar data is analyzed through the ATR and iSAM system shown in Figure 1.2, the map appears as in Figure 5.6.

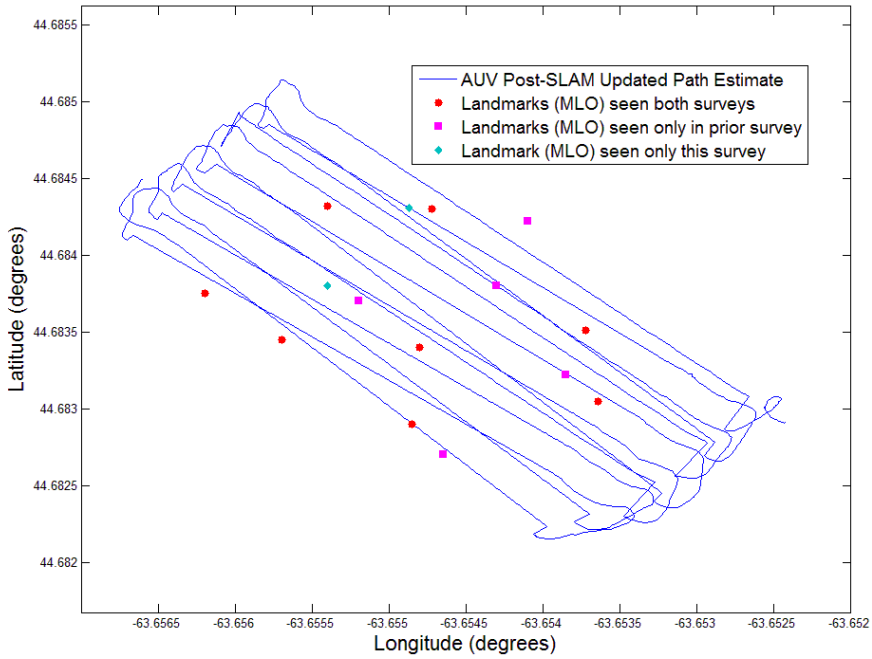


Figure 4.6: SLAM position updated runs for Jan 2014 runs. Paths underwater are still mostly straight, due to limited data points for developing curvature, but paths are not parallel - angles correspond to cumulated drift from dead-reckoning.

Of note, the red circles are landmarks found in both the Jan 2014 and the prior runs over the same area, while the blue circles are landmarks found only in the new survey, and the green diamonds are landmarks detected only in the older survey. Manual examination of the side-scan sonar imagery suggests that this is more accurate than the previous estimates of position, and further the GPS update 'jumps' are substantially reduced on almost all transects of the run (though three transects detected no objects and therefore did not change). This inference of increased accuracy is supported by the reduced error in position from a GPS update jump upon surfacing. The results for a single transect before and after adjustment by the SLAM algorithm are shown in Figure 5.7.

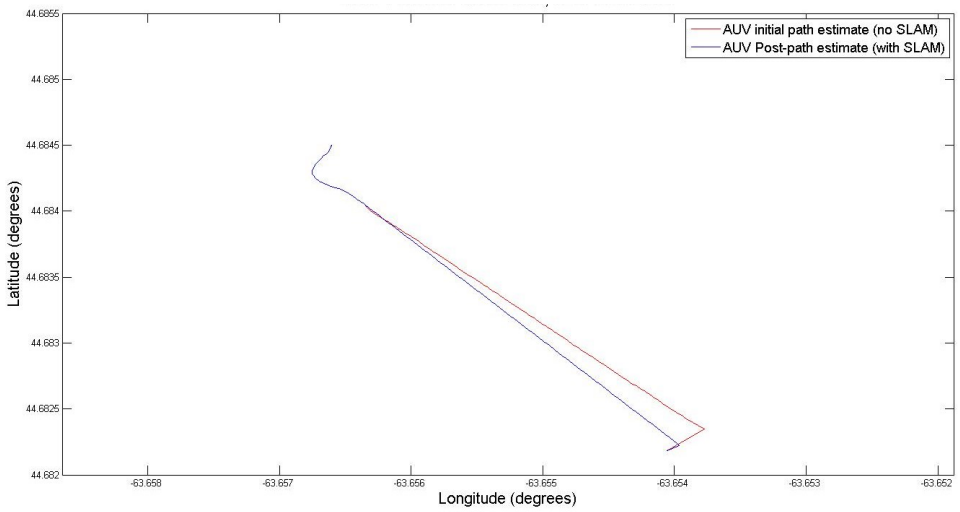


Figure 4.7: Difference in AUV position before and after SLAM position update from the Jan 2014 mission. Note the 'jump' at the end of the transect (bottom of diagram) upon surfacing is reduced as the path with SLAM is closer to the GPS ground truth than the dead reckoned one.

In general, this difference in location accuracy was parameterized as ε. ε was defined as follows:

$$\varepsilon = \frac{|\theta_{GPS} - \theta_{slam}|}{|\theta_{GPS} - \theta_{dr}|} \qquad (1)$$

Where θ_{GPS} is the GPS-verified true position at surfacing, θ_{slam} is the SLAM estimate of position when surfacing, and θ_{dr} is the dead-reckoning position when surfacing, and the differences are taken as the absolute distance between the positions as a length in meters. The unitless ratio, then between the error in the SLAM estimate and the dead reckoning estimate is less than 1 if the estimate improves our accuracy (reduces error) and more than one if the estimate increases our error in position estimate. This is shown in the table below with an average error value. (Survey A is omitted for reasons discussed immediately hereafter):

Table 4: Error results

survey	pre-loaded target pos age	Survey Type	avg ε (Eq. 1)
B	none	Original detection	0.79
B	3 wks	re-visit	0.57
B	5 wks +8 wks	re-visit	0.40
C	none	Original detection	0.70
C	8 wks	re-visit	0.30
C	8 wks	re-visit	0.34
D	none	Original detection	0.83
D	3 wks	re-visit	0.56
D	5 wks +8 wks	re-visit	0.52

These average values decreased as surveys incorporating previous data were done, because previous estimates could be used to increase positional accuracy. The original detections had localization improvements that were highly dependent on seeing the same object from multiple orientations in the course of their transects – and objects which were only passed by once could not improve things. In the re-visit condition, when using localizations made previously, the error in position was substantially reduced because objects seen only once in the course of the entire mission could still be used to localize if they had been seen before in the course of the previous mission.

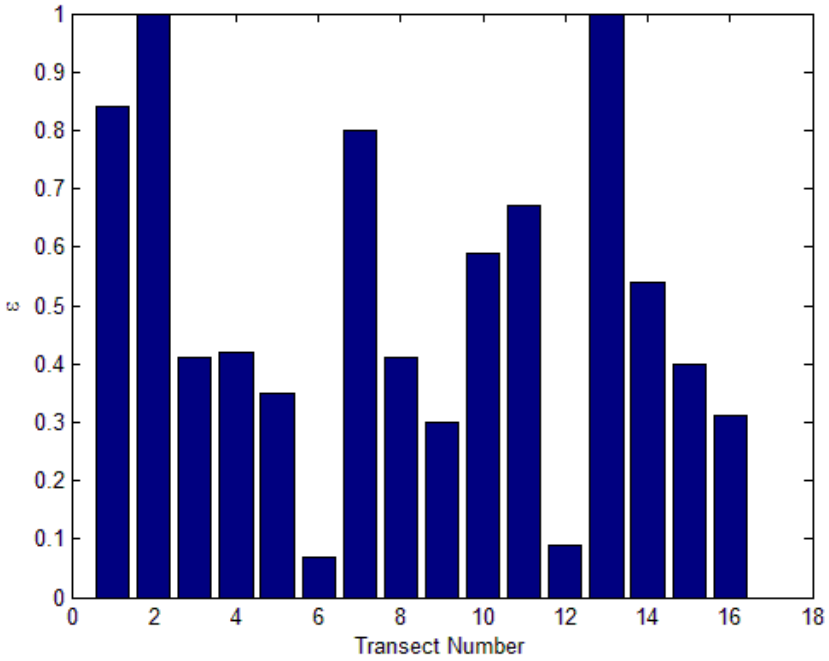


Figure 4.8: Survey leg comparison for Survey B, second survey, ε by transect number. The average error decreased to 0.57, but there was variance in performance in individual transects.

As shown in Figure 5.7 and 5.8, the SLAM updated path is closer to the GPS ground truth position at the end of the transect, and therefore the error metric ε is reduced for most transects. Figure 5.8 illustrates how on some transects there were no landmarks observed (2, 13) and no update to position could be applied over the dead-reckoning estimate. Several others, e.g. transects 1 and 7, observed landmarks near the beginnings of their transects only. As they traversed further, their localization update did not help nearly as substantially as a few others that observed known landmarks just before rising like 6 and 12.

The following comparisons were made:

1. Mission A using data from October 24rd on October 22nd
2. Mission A using data from October 24th on January 21st

3. Mission B using data from October 24th on December 12th
4. Mission B using data from December 12th and October 24th on January 20th
5. Mission C using data from October 22nd on October 23rd
6. Mission C using data from October 22nd and 23rd on January 21st
7. Mission D using data from October 22nd on December 24th
8. Mission D using data from October 22nd and December 12th on January 20th.

Of these eight comparisons, the following was observed:

1 and 2: Mission A was too sparse. The number of landmarks that could be observed was small and not repeatable. There was also noise detected and interpretted as landmarks. No real targets were observed. Results were therefore $\varepsilon=1.0$ (no accuracy increase).

3,4 and 7,8: Reasonable detections with good estimates of position updates. Note that on these missions the AUV survey direction was the same. However, #4 performed better without the data from October 24th included in the analysis as it had an incorrect association causing one transect to be off noticeably, though no such association was made between the December to January data. This may be due to the three-month interval between missions and hence much had changed in the environment. Nevertheless, the overall total error in position estimates decreased more without the October data due primarily to this incorrect association.

5-6: When using data that is only measured in one direction similar performance between 3,4 and 7,8 mission pairs was observed. When using the full mission, which included orthogonal heading side-scan sonar measurements, there was a number of landmarks (3 of the ~30 overall landmarks) which were found in one direction of the survey and not in the other. This did not, however, seem to impair the overall SLAM performance.

It was difficult to assess if the landmarks were better than non-landmarks for ground truth, as the areas involved are dynamically changing and surveys of them yield accurate results but also capture the dynamic changes every time, with the change increasing as the time between surveys grows. However, the increase in accuracy of robot position estimates indicates that the accuracy of detections on its transects would also increase. In summation, the overall accuracy of the map increases.

5.2. PARAMETER TUNING RESULTS

In general, parameter tuning results were as follows:

Table 5: ATR Parameters

Parameter Name	Parameter Number
ObjectSizeAcrossTrackMeters	1.5
ObjectSizeAlongTrackMeters	1.5
ObjectHeightMeters	0.75
SmoothingSizeAcrossTrackMeters	4.0
SmoothingSizeAlongTrackMeters	4.0
BackgroundSizeAcrossTrackMeters	10.0
BackgroundSizeAlongTrackMeters	2.0
DetectionThreshold	0.6
MinimumRangeMeters	6.0
MaximumRangeMeters	50.0

The various settings were adjusted manually by comparing resulting outputs to the input stream as well as SLAM matches, adjusting parameters as seemed good to remove bad identifications. This was done manually rather than by searching automatically and thoroughly because the number of parameters to tune was so high as to render any proper complete exploration difficult when processing times for an individual run were in excess of 10 minutes.

The most sensitive and important parameter is the Object Height Meters parameter, which the ATR uses to filter out a minimum size of an object protruding from the sea bed. At about 0.4 m and below, there are too many false positives and landmarks that are less consistently placed over time are detected. As it goes higher, though, fewer and fewer landmarks qualify and there are no detections above ~1.5m in the whole dataset. The object size along and across track (in the vertical and horizontal orientations in the sonar data) are also fairly important and sensitive. The other parameters are less important, and are more about filting out noise from the sonar results.

The SLAM algorithm results were simpler: A time of 10 seconds between poses while underwater was used to generate practically computable SLAM maps. The parameters were tuned by adjusting relative covariances in matrices, starting with an initial assumption that the kinematic model of the Iver2 was probably very inaccurate to real world results but the sidescan sonar distances were likely to be correct. The maximum distance between estimated positions and possible matches in position was chosen to be 30 meters based on estimates of maximum AUV deviation in location during these field tests, which proved accurate.

5.3. PROCESSING TIME AND COMPUTATION REQUIREMENTS

On-line processing requirements appear satisfactorily met for the missions studied on the Iver2 payload processor but could be improved. Figure 5.8 summarizes the processing time required for the cases studied. The total processing time is defined as the time between when the sonar stops logging to a .872 file and when the ATR has completed its final calculations for a run. Figure 5.8 pertains to a set of data between waypoints 64 and 69 (waypoints not explicitly shown) where the AUV surveyed on orthogonal headings as in Figure 4.10.

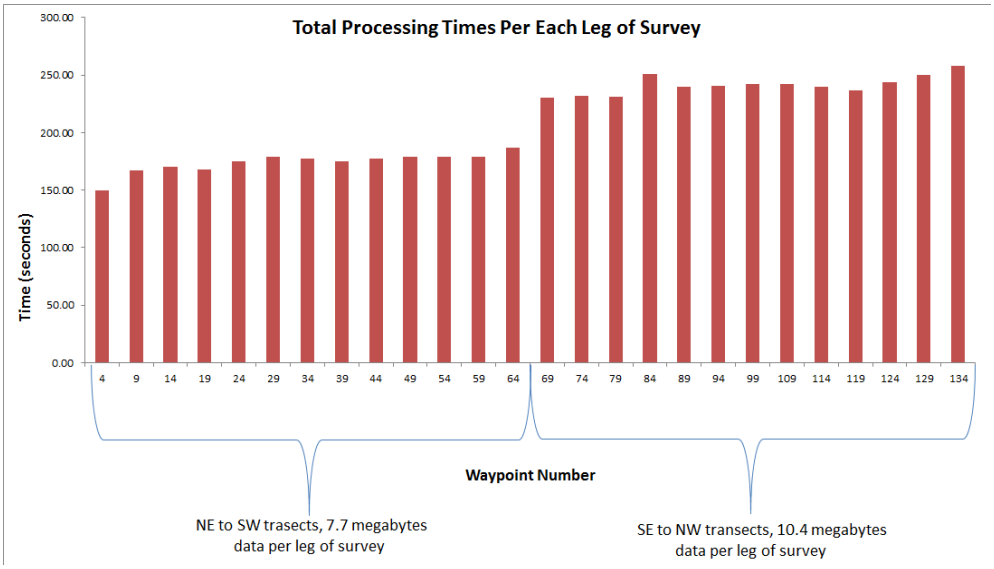


Figure 4.9: Total processing times for several side-scan sonar processing.

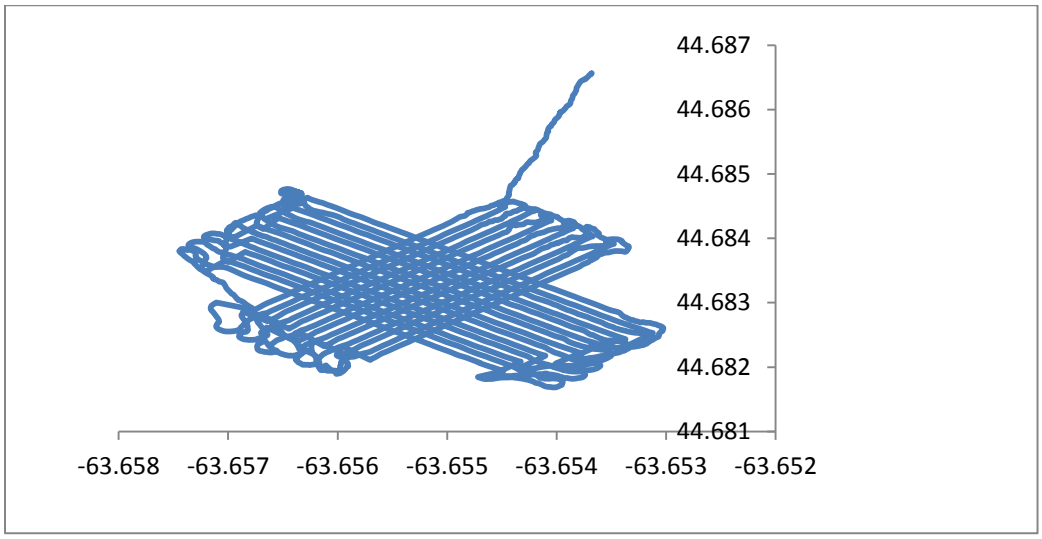


Figure 4.10: Position estimates of Iver2 AUV based on dead-reckoning (straight sections) and when on surface (end of transects) where it accesses GPS.

This was measured from an earlier surveyed area so the data was read in (adding to calculation time) and referenced for data association. Substantial amounts of the calculation time (> 25%) was consumed converting the sonar imagery from *.872* to *.xtf* format. If the side-scan sonar data was in the correct format or the ATR could read the *.872* data, this calculation time would be reduced. As-is, the program never ran longer than the time it took for the next transect's sonar imagery data to be logged. There can be different increased times between robot poses in the SLAM representation of robot

position – more poses increases processing time for iSAM and data association but yields a finer grained model.

The breakdown in processing time for individual tasks per transect is shown in Figure 4.11 for Mission E.

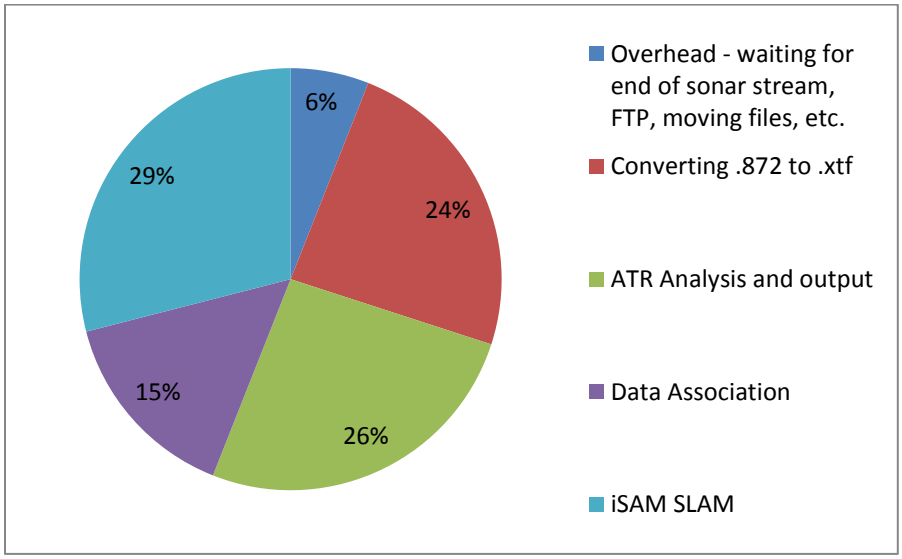


Figure 4.11: Distribution of average processing time for one transect in Mission E.

Figure 5.10 shows 6% of the time spent transferring files over FTP between the two on-board processors, monitoring sonar imagery file size to determine if a transect had completed logging so it could be accessed, and other file transfer tasks. A large portion of the remaining time, 24%, was spent converting sonar data formats in an inefficient script. The ATR analysis and output segment in Figure 5.10 includes time to perform the ATR analysis, parse the subsequently generated detected MLOs into a list, and synchronize the Iver poses for the data association. This data association time (purple) includes time for iSAM to extract covariance information used only in data association.

5.4. LIMITATIONS OF RESULTS

This project was undertaken without using sophisticated and expensive IMUs. IMUs could increase navigational accuracy substantially, but they add additional equipment costs and still do not prevent errors from growing unbounded. With substantially reduced error in navigation, it would take substantially longer underwater missions before SLAM became helpful and useful.

This project was undertaken with limited numbers of different ways of analyzing Sidescan sonar data for landmarks. The author created an image parser, used the DRDC Sidescan Sonar ATR, and also tried SIFT and SURF features in the image. There is a lot of data in the image not properly captured by these types of analysis, and incorporating e.g. relative elevation data into the process could improve localization results.

This project was undertaken in one relatively small geographical area, and some of the parameters are likely sensitive to the specific area testing was done in. Many of the specific parameters are calibrated to this specific AUV, Sidescan Sonar type, heigh above sea floor, and other features specific to the test environment – for actual widespread deployment, much more data could be gathered to help make a more general SLAM system. For some areas of the sea floor, where there is little recognizable terrain, even this is likely not terribly helpful.

The project assumed no use of sonar modems to localize via long baseline, short baseline, etc. acoustic communications. A base station with a known location could help localize the AUV using the time-of-flight of acoustic communications, and data from these could in future be incorporated into the AUV localization. Also, another AUV with sonar communications could relay sonar information back and forth for localization efforts to e.g. share landmark locations or relative positions.

CHAPTER 5 CONTRIBUTIONS

The contributions from this thesis are itemized in this section. They include the following:

The first time that the DRDC ATR tools are integrated into an on-board SLAM method where ATR is used for the detection of objects from a data stream. In fact, there are currently very few other SLAM methods that draw on ATR output. This ATR appears to be a fairly good 'perception' tool for the purposes of simultaneous localization and mapping for mine-like objects.

Based on the output of the ATR and the mapping requirements, joint compatibility branch and bound is the best of the commonly used data association algorithms in terms of accuracy for AUV and MLO localization. It is also the first test of ATR as a means for change detection as well. As a change detection tool, the DRDC ATR shows promise.

Validated concepts for improved AUV and landmark localization as well as change detection for underwater mine-like objects as per the overall project objective.

CHAPTER 6 CONCLUSIONS AND FUTURE WORK

Based on the integration, simulator testing, and in-water validation work reported here, there are follow-on components that could be developed to increase the efficiency and general accuracy of the SLAM system shown in Figure 1.2, namely:

- As mentioned earlier, a more efficient way to convert from the *.87*2 to *.xtf* sonar format to reduce the time needed for this task to decrease the overall processing time. To some extent the requirement for this task has been superseded by IVER3 UUVs that have networked access to the side-scan sonar imagery across processors and produces the imagery in a format that the ATR can read.

- The use of an sophisticated Inertial Measurement Unit (IMU) could potentially perform the same job of increasing accuracy underwater navigation. Compare accuracy to use of a sophisticated Inertial Measurement Unit (IMU), and test for SLAM navigation improvement with an IMU installed. An IMU-based system would still have unbounded error growing over time eventually, and would require additional expensive hardware to be added to an AUV.

- A compilation of more data sets to produce change detection plots like those in Figure 5.10. While this one set proves the principle of the system that was assembled this should be tested over different sized missions, seabed bottom types (to task the ATR in different environments), landmark distributions, etc. The ATR as a change detection tool would be further tested.

- Coordination with other vehicles for multi-vehicle SLAM, ranging through acoustic micromodem communications to increase accuracy of iSAM. This of necessity entails compression of data into as few bits as possible while still being useful to SLAM techniques, and may involve some sort of partitioning scheme for dividing up the area to be surveyed. Also, with multiple vehicles present they can send/receive acoustic signals to determine the range between Iver2 AUVs reliably.

- Use of more sonar data for different parts of the ocean than just Halifax Harbour, for increased generality. In general, this project surveyed five different areas within line

of sight of the DRDC testing barge in Bedford basin. To make the techniques developed more applicable to general underwater environments, requires extensive testing and calibration in other environments.

- Provisions for transferring data about changes detected during missions via e.g. acoustic modem communication relays to dock/ship/station for 'live' updates on presence of new mines. This could have practical applications, and again relies upon compressing relevant information and transmitting through an inherently noisy, lossy medium.

- Additional different types of landmarks could be investigated. This project focuses on minelike object landmarks using the DRDC ATR. While this is highly practical for several intended use cases, further projects could use other sidescan sonar features as landmarks.

- Related to the previous point, to fortify the accuracy of joint compatibility branch and bound for data association, the local seabed elevation imagery that could be extracted from side-scan sonary imagery be jointly associated with the MLO localization. This does not involve extra measurements but it does involve more processing on the same collected side-scan imagery. This topic is being pursued in another project.

BIBLIOGRAPHY

[1] Antonelli, G, *Underwater Robots*. pp 1-13. Springer, 2006.

[2] Stone, W., Hogan, B., Flesher, C., Guliati, S., Richmond, K. et al, "Design and Deployment of a Four-Degrees-of-Freedom Hovering Autonomous Underwater Vehicle for sub-Ice Exploration and Mapping". *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*. 2010.

[3] Crees, T., Kaminski, C., Ferguson, J., Laframboise, J., Forrest, A. Williams, J, MacNeil, E. Hopkins, D. and Pederson, R. "Preparing for UNCLOS - An Historic AUV Deployment in the Canadian High Arctic". *Proceedings of the Oceans / MTS Conference*, pp. 8-16. 2010.

[4] CBC News. "Malaysia Airlines Flight MH370: What's Needed to Find it." Online: <http://www.cbc.ca/news/canada/british-columbia/malaysia-airlines-flight-mh370-what-s-needed-to-find-it-1.2566309> March 10, 2014.

[5] Walter, M., Hover, F., and Leonard, J, "SLAM for Ship Hull Inspection using Exactly Sparse Extended Information Filters". *IEEE Conference on Robotics and Automation*, pp. 1463-1470. 2008.

[6] Paull, L., Sajad, S., Seto, M, and Li, H, "A Multi-agent Framework with MOOS-IvP for Autonomous Underwater Vehicles with Sidescan Sonar Sensors". *Autonomous and Intelligent Systems 2nd International Conference, AIS 2011.* pp. 41-50. 2011.

[7] Fallon, M., Papadopoulos, G., Leonard, J. and Patrikalakis, N. "Cooperative AUV navigation using a single maneuvering surface craft." *Intl. J. of Robotics Research*, vol 29 no. 12 pp. 1461–1474, October 2010.

[8] Doug Wallace, Professor, Dept. of Oceanography. Canadian Conference on Electrical and Computer Engineering (CCECE) Public Address. May 6, 2015.

[9] Christ, Robert and Wernlin, Robert. The ROV Manual: A user guide for remotely operated vehicles. 2013. 2nd edition. Elsevier. Ch.16. 2013.

[10] United States Department of the Navy, "The Navy Unmanned Undersea Vehicle (UUV) Master Plan." November 9, 2004. Retrieved June 13 2012 from <http://www.navy.mil/navydata/technology/uuvmp.pdf>

[11] Folkesson, J. and Leonard, J, "Autonomy Through Slam for an Underwater Robot." *Robotics Research*, pp. 55-70. 2011

[12] Siegwart, R., and Nourbakhsh, I, *Introduction to Autonomous Robots*. pp. 1-336. Apr. 2004.

[13] Durant-Whyte, H., and Bailey, T, "Simultaneous Localization and Mapping: Part 1". *IEEE Robotics and Automation Magazine,* pp. 99-109. Jun. 2006.

[14] Thrun, S., Burgard, W., and Fox, D, *Probabilistic Robotics*. MIT University Press, pp. 1-647. 2006.

[15] Dissanayake, M., Newman, P., Clark, S., Durrant-Whyte, H., and Csorba, M, "A Solution to the SLAM Problem". *IEEE Transactions on Robotics and Automation*, pp. 229-241. Jun. 2001.

[16] Moore, D., Huang, A., Walter, M., Olson, E., Fletcher, L., Leonard J., and Teller, S, "Simultaneous Local and Global State Estimation for Robotic Navigation". *International Conference on Robotics and Automation,* pp. 3794-3799. 2009

[17] Trappenberg, T, *Lecture Notes for CSCI 6905: Probability Theory*. pp. 1-30. 2011.

[18] Leonard, J, et al. "A Perception Driven Autonomous Urban Vehicle". *Journal of Field Robotics*, pp. 727-774. 2008

[19] Durant-Whyte, H., and Bailey, T. "Simultaneous Localization and Mapping: Part 2". *IEEE Robotics and Automation Magazine,* pp. 108-117. Sep. 2006

[20] Eickstedt, D.P., and S.R. Sideleau. “The backseat control architecture for autonomous robotic vehicles: A case study with the Iver2 AUV.” OCEANS 2009, MTS/IEEE Biloxi – Marine Technology for Our Future: Global and Local Challenges. pp. 1-8. 2009.

[21] Seto, M, "Autonomous Robotics Lecture Notes 6". pp. 1-78. 2011.

[22] Olson, E., Leonard, J., and Teller, S, “Fast Iterative Alignment of Pose Graphs with Poor Initial Estimates”. *IEEE International Conference on Robotics and Automation*. 2006.

[23] Olson, E., and Kaess, M, “Evaluating the Performance of Map Optimization Algorithms”. *Workshop on Good Experimental Methodology in Robotics*, pp 1-7. 2009.

[24] Nerurkar, E., and Roumeliotis, S, “Power-SLAM: a linear-complexity, anytime algorithm for SLAM”. *International Journal of Robotics Research*, pp. 772-788. Jan. 2011.

[25] Bosse, M., Newman, P., and Leonard, J, “Simultaneous Localization and Map Building in Large-Scale Cyclic Environments Using the Atlas Framework”. *International Journal of Robotics Research*, Vol.23, No.12, pp. 1113-1139. 2004.

[26] Olson, E., Leonard, J., and Teller, S, “Spatially-Adaptive Learning Rates for Online Incremental SLAM”. *Proceedings of Robotics: Science and Systems.* 2006.

[27] Walter, M., Eustice, R., and Leonard, J, “Exactly Sparse Extended Information Filters for Feature-based SLAM”. *International Journal of Robotics Research*. vol. 26 no. 4 pp. 335-359. April 2007.

[28] Eustice, Ryan M., Singh, H., and Leonard, J, "Exactly Sparse Delayed-State Filters for View-Based SLAM". *IEEE Transactions on Robotics*, Vol. 22, No. 6, pp. 1100-1114. 2006

[29] Intel Corporation. “Excerpts from a Conversation with Gordon Moore: Moore’s Law.” 2005. Retrieved June 13 2012 from <ftp://download.intel.com/museum/Moores_Law/Video-Transcripts/Excepts_A_Conversation_with_Gordon_Moore.pdf>

[30] Fortman, T., Bar-Shalom, Y., and Scheffe, M, “Sonar Tracking of Multiple Targets Using Joint Probabilistic Data”. *IEEE Journal of Oceanic Engineering*, pp. 173-184. 1983.

[31] Folkesson, J. , Leederkerken, J. Williams, R., Patrikalakis, A., and Leonard, J, "A Feature Based Navigation System for an Autonomous Underwater Robot". *Field and Service Robotics*, pp. 105-114. 2008

[32] Mahon, I., Williams, S., Pizzaro, O., and Roberson, M, “Efficient View-Based SLAM Using Visual Loop Closures”. *IEEE Transactions on Robotics, Vol. 24, No. 5.,* pp. 1002-1014. 2008.

[33] Siciliano, B., Khatib, O., and Groen, F, "Underwater SLAM For Structured Environments Using an Imaging Sonar." *Springer Tracts in Advanced Robotics*. pp. 7-21, 113-119. 2010.

[34] Agarwal, Pratik and Olson, Edward. Variable Reordering Strategies For SLAM. *Proceedings of the IEEE International Conference on Intelligent Robots and Systems,* 2012.

[35] Loebis, D., Sutton, R., Chudley, J., and Naeem, W. “Adaptive tuning of a Kalman filter via fuzzy logic for an intelligent AUV navigation system”. *Control Engineering Practice 12 (2004)*, pp. 1531–1539. Aug. 2003.

[36] Arulampalam M., Maskell S., Gordon N., and Clapp, T, “A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking”. *IEEE Journal of Signal Processing*, pp. 174-188. Feb. 2002.

[37] Dellaert, F. and Kaess, M. "Square Root SAM: Simultaneous Localization and Mapping via Square Root Information Smoothing". *International Journal of Robotics Research*, vol. 25, no. 12, Dec. 2006, pp. 1181-1204.

[38] Kaess, M., Ranganathan, R., and Dellaert, F, "iSAM: Incremental Smoothing and Mapping". *IEEE Transactions on Robotics,* vol. 24, no. 6, Dec. 2008, pp. 1365-1378. 2008.

[39] Kaess, M., and Dellaert, F, "Covariance Recovery from a Square Root Information Matrix for Data Association". *Journal of Robotics and Autonomous Systems (RAS)*. vol. 57, pp. 1198-1210. 2009.

[40] Fallon, M., Kaess, M., Johannsson, H., and Leonard, J, "Efficient AUV Navigation Fusing Acoustic Ranging and Side-scan Sonar". *International Conference on Robotics and Automation*. pp. 2398-2405. 2011.

[41] Kaess, M., Ila, V., Roberts, R., and Dellaert, F, "The Bayes Tree: An Algorithmic Foundation for Probabilistic Robot Mapping". *Int'l Workshop on the Algorithmic Foundations of Robotics (WAFR)*. pp. 157-173. 2010.

[42] Kaess, M., Johannsson, H., Roberts, R., Ila, V., Leonard, J., and Dellaert, F, "iSAM2: Incremental Smoothing and Mapping with Fluid Relinearization and Incremental Variable Reordering". *International Conference on Robotics and Automation*. 2011.

[43] Lewis, F, *Autonomous Mobile Robots: Sensing, Control, Decision Making and Applications*. Taylor & Francis Group, LLC. 2006.

[44] Singh, A., "Review Article: Digital change detection techniques using remotely-sensed data", International Journal of Remote Sensing, vol. 10, no. 6, pp. 989–1003, 1989.

[45] Lebart, K. Trucco, E. and Lane, D. “Real-time automatic sea-floorchange detection from video,”. Proceedings of MTS/IEEE OCEANS, Sep. 2000, pp. 337–343.

[46] Edgington, D., Dirk, W. Salamy, K., Koch, C. Risi, M. and Sherlock, R. “Automated event detection in underwater video,”. Proceedings of MTS/IEEE Oceans Conference, 2003.

[47] Radke, R., Andra S., Al-Kofahi O. and Roysam B., ”Image change detection algorithms: a systematic survey” IEEE Transactions on Image Processing, vol. 14, no. 3, pp. 294–307 March 2005.

[48] Wei, S, Leung, H, and Myers, V. “An Automated Change Detection Approach for Mine Recognition Using Sidescan Sonar Data”. IEEE International Conference on Systems, Man, and Cybernetics. pp. 554–558. 2008.

[49] Midtgaard, Ø, Hansen, R., Sæbø, T., Dubberley, J, Quidu, I, and Myers, V. “Change Detection Using Synthetic Aperture Sonar: Preliminary Results from the Larvik Trial”. IEEE OCEANS 2011. pp. 1-8. 2011.

[50] Norvig, Peter and Thrun, Sebastian. "Introduction to Artifical Intelligence: Unit 2: Problem Solving." Retrieved Monday June 25th from <https://www.ai-class.com/home/>

[51] Itseez. “OpenCV Documentation”. 2016. Retrieved March 17th 2016 from <http://opencv.org/documentation.html>

[52] Bay, H. Ess, A, Tuytelaars, T, and Van Gool, L. “Speed Up Robust Features (SURF)”. *Journal of Computer Vision and Image Understanding*. vol. 110 no. 3, June, 2008
pp. 346-359.

[53] Mahalanobis, P. "On the generalised distance in statistics". *Proceedings of the National Institute of Sciences of India* vol. 2 no. 1, pp. 49–55. 1936.

[54] Stentz, A, "Real-Time Replanning in Dynamic and Unknown Environments". Retrieved June 18 2012 from <http://www.frc.ri.cmu.edu/~axs/dynamic_plan.html>.

[55] Guernane, R., and Achour, N, “Generating optimized paths for motion planning”. *Robotics and Autonomous Systems*, vol. 59, no. 10, pp. 789–800. 2011.

[56] Fenwick, J., Newman, P., and Leonard, J, "Cooperative Concurrent Mapping and Localization ". IEEE Conference on Robotics and Automation, pp. 1810-1817. 2002

[57] Chiddarwar, S., and Babu, N, “Conflict free coordinated path planning for multiple robots using a dynamic path modification sequence”. *Robotics and Autonomous Systems*, 2011.

[58] Rekleitis, I., New, A., and Chosit, H, “Distributed Coverage of Unknown/Unstructured Environments by Mobile Sensor Networks”. *Proceedings from the 2005 International Workshop on Multi-Robot Systems* (Springer). Pt IV, pp. 145-155. 2005.

[59] Zengin, U., and Dogan, A, “Cooperative target pursuit by multiple UAVs in an adversarial environment”. *Robotics and Autonomous Systems.* vol. 59, no. 12, pp. 1049–1059. 2011.

[60] Cortes, J., Martinez, S., Karatas, T. and Bullo, F, “Coverage Control for Mobile Sensing Networks”. *IEEE Transactions on Robotics and Automation*, pp. 243-254. Apr. 2004.

[61] Fallon, M., Leonard, J., and Bahr, A, “Cooperative Localization for Autonomous Underwater Vehicles”. *International Journal of Robotics Research*. vol. 28 no. 6 pp. 714-728. 2009.

[62] Fallon, M. , Papadopoulos, G., and Leonard, J, "A Measurement Distribution Framework for Cooperative Navigation using Multiple AUVs". *IEEE International Conference on Robotics and Automation*, pp. 4256-4263. 2010.

[63] Fallon, M., Papadopoulos, G., Leonard, J., and Patrikalakis, N, “Cooperative AUV Navigation using a Single Maneuvering Surface Craft.”. *International Journal of Robotics Research*, pp. 1461-1474. Oct. 2010.

[64] Bahr, A., Walter, M., and Leonard, J, "Consistent Cooperative Localization". *IEEE International Conference on Robotics and Automation*, pp. 3415-3422. 2009.

[65] Kim, B., Kaess, M., Fletcher, L., Leonard, J., Bachrach, A., Roy N., and Teller, S, “Multiple Relative Pose Graphs for Robust Cooperative Mapping”. *International Conference on Robotics and Automation.* 2010.

[66] Schwager, M., Rus, D., and Slotine, J, “Decentralized, Adaptive Coverage Control for Networked Robots”. *International Journal of Robotics Research*, pp. 357-373. 2009.

[67] Hudson, J, “Adaptive Path Planning for an Autonomous Marine Vehicle Performing Co-operative Navigation for Autonomous Underwater Vehicles”. Dalhousie University. 2012.

[68] Seto, M., Hudson, J. and Pan, Y, "Three Dimensional Path Planning for a Communications and Navigation Aid Working Cooperatively with Autonomous Underwater Vehicles". *Autonomous and Intelligent Systems 2nd International Conference, AIS 2011.* pp. 51-62. 2011.

[69] Sanderson, C., Gibbins, D, and Searle, S, "On Statistical Approaches to Target Silhouette Classification in Difficult Conditions". *Digital Signal Processing* No. 18, pp. 375-390. 2008

[70] Suvorova, S. and Schroeder, J, “Automated Target Recognition Using the Karhunen-Loéve Transform With Invariance”. *Digital Signal Processing*, pp. 295-306. 2002.

[71] Ravichandran B.,, Gandhe, A., Smith, R., and Mehra, R, "Robust automatic target recognition using learning classifier systems". *Information Fusion*, pp. 252-265. 2007.

[72] Chapple, P., Bertilone, D., Caprari, R., and Newsam, G, "Stochastic Model-Based Processing for Detection of Small Targets in Non-Gaussian Natural Imagery". *IEEE Transactions on Image Processing*, Vol. 10, No. 4, pp. 554-564. 2001

[73] Blondel, P, "Automatic Mine Detection by Textural Analysis of COTS Sidescan Sonar Imagery." *International Journal of Remote Sensing*, vol. 21, no 16, pp. 3115-3128. 2000.

[74] Reed, S., Petillot, Y., and Bell, J, "An Automatic Approach to the Detection and Extraction of Mine Features in Sidescan Sonar". *IEEE Journal of Oceanic Engineering*, Vol. 28, No 4., pp. 90-105. 2003

[75] Coiras, E., Petillot, Y., and Lane, D, "Multiresolution 3-D Reconstruction From Side-Scan Sonar Images". *IEEE Transactions on Image Processing*, Vol. 16, No. 2, pp. 382-390. 2007.

[76] Sidek, O., and Quadri, S, "A review of data fusion models and systems". *International Journal of Image and Data Fusion*, Vol 3, No 1., pp. 3-21. 2012

[77] Ainslie, M, *Principles of Sonar Performance Modelling*. Praxis Publishing Ltd. Pp 3-52, 361-512, 2010.

[78] Wille, P, *Sound Images of the Ocean in Research and Monitoring*. Springer-Verlag Berlin Heidelberg, 2005.

[79] Blondell, P., *The Handbook of Sidescan Sonar*. Praxis Publishing Ltd. 2009.

[80] Chapple, P, "Automated Detection and Classification in High-Resolution Sonar Imagery for Autonomous Underwater Vehicle Operations". Australian Government Department Of Defense. 2008.

[81] Fahimi, F, *Autonomous Robots: Modeling, Path Planning, and Control*. Springer Science+Business Media. 2009.

[82] Pastore, T., Fioravante, T, and Vermeij, A, "Buying and MOOSifying Two USVs in 2008/2009." MOOS-DAWG '10. Retrieved online from <http://oceanai.mit.edu/moos-dawg10/material/11-brief-pastore.pdf> on March 12, 2013. Aug. 2010.

[83] Linux for ARM on TS-7000 Embedded Computers. Technologic Systems. Online. Retrieved from <https://www.embeddedarm.com/software/software-arm-linux.php> August 16, 2013. 2009.

[84] TS-7800 Datasheet. Technologic Systems. Online. Retrieved from <https://www.embeddedarm.com/documentation/ts-7800-datasheet.pdf> August 16, 2013. 2009.

[85] TS-7800 Kernel Compile Guide. Technologic Systems. Online. Retrieved from <https://www.embeddedarm.com/about/resource.php?item=412> August 16, 2013. 2009.

[86] TS-7800 Manual. Technologic Systems. Online. Retrieved from <https://www.embeddedarm.com/about/resource.php?item=393> August 16, 2013. 2009.

[87] Li, B. et al, "MIMO-OFDM for High-Rate Underwater Acoustic Communications". *IEEE Journal of Oceanic Engineering*, pp. 634-644. Oct. 2009.

[88] Sozer, E., Stojanovic, M., and Proakis, J, "Underwater Acoustic Networks". *IEEE Journal of Oceanic Engineering*, pp. 72-83. Jan. 2000.

[89] Benjamin, M, "The Interval Programming Model For Multi-object Decision Making." MIT Computer Science and Artificial Intelligence Laboratory. Online. Retrieved from <ftp://publications.ai.mit.edu/ai-publications/2004/AIM-2004-021.pdf> May 2012. 2004.

[90] Benjamin, M. "MOOS-IvP Autonomy Tools User Manual." MIT Computer Science and Artificial Intelligence Laboratory. Online. Retrieved from <http://oceanai.mit.edu/moos-ivp-pdf/moosivp-tools.pdf> May 2012. 2008.

[91] Benjamin, M. "MOOS-IvP Autonomy Tools User's Manual Release 4.2.1." MIT Computer Science and Artificial Intelligence Laboratory. Online. Retrieved from < http://oceanai.mit.edu/moos-ivp-pdf/moosivp-tools.pdf> May 2012. 2009.

[92] Benjamin, M., Newman, P., Schmidt, H., and Leonard, J. "Extending a MOOS-IvP Autonomy System and Users Guide to the IvPBuild Toolbox." MIT Computer Science and Artificial Intelligence Laboratory. Online. Retrieved from < https://oceanai.mit.edu/svn/moos-ivp-aro/releases/moos-ivp-4.1/ivp/docs/MIT-CSAIL-TR-2009-037.pdf> May 2012. 2010.

[93] Benjamin, M., Newman, P., Schmidt, H., and Leonard, J. "A Tour of MOOS-IvP Autonomy Software Modules." MIT Computer Science and Artificial Intelligence Laboratory. Online. Retrieved from <https://dspace.mit.edu/bitstream/handle/1721.1/44590/MIT-CSAIL-TR-2009-006.pdf?sequence=1"> March 2016. 2009.

[94] Benjamin, M., Newman, P., Schmidt, H., and Leonard, J. "An Overview of MOOS-IVP and a Brief Users Guide to the IvP Helm Autonomy Software." MIT Computer Science and Artificial Intelligence Laboratory. Online. Retrieved from <https://oceanai.mit.edu/svn/moos-ivp-aro/releases/moos-ivp-4.1/ivp/docs/MIT-CSAIL-TR-2010-041.pdf> May 2012. 2009.

[95] Schmidt, H, Benjamin, M., Balasuriya, A., Cockrell, K., and Lum, R. "MOOS-IvP Undersea Autonomous Network Simulator User's Guide." Online. Retrieved from <http://acoustics.mit.edu/faculty/henrik/LAMSS/simguide.pdf> August 2013. 2008.

[96] Goby User's Manual 1.1. MIT. Online. Retrieved from < http://gobysoft.org/dl/goby1-user-manual.pdf> August 2013.

[97] T. A. Davis, J. R. Gilbert, S. Larimore, E. Ng. "A column approximate minimum degree ordering algorithm". *ACM Transactions on Mathematical Software*, vol 30, no. 3, Sept. 2004, pp. 353-376.