Underwater Localization and Mapping for Cost-Effective Robots

Submitted in partial fulfillment of the requiurements for

the degree of

Doctor of Philosophy

in

Mechanical Engineering

Akshay A. Hinduja

B.Tech., Production Engineering, Veermata Jijabai Technological Institute
M.S., Mechanical Engineering, Carnegie Mellon University

Carnegie Mellon University
Pittsburgh, PA

August 2024

Keywords: Simultaneous Localization and Mapping; Underwater Navigation; Acoustic Localiztion; Feature-based Methods; Machine Learning; Deep Learning; Computer Vision

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my advisor Dr. Michael Kaess for hiring me as staff, and later as a PhD student in his group. His mentorship and guidance over the years has made all the work in this thesis, and research outside of it possible. This opportunity allowed me to develop my passion for field and underwater robots, and his feedback, insights and wisdom at numerous times over the years greatly helped me develop as a researcher. I could not have asked for a better advisor in this journey than Michael.

I would also like to thank the members of my thesis committee, Dr. Kenji Shimada, Dr. Matthew Johnson-Robertson, and Dr. Joshua Mangelson, for their invaluable insights, constructive feedback, and support for my thesis.

I extend my heartfelt appreciation to my colleagues and co-authors, whose collaboration has been both a privilege and a pleasure. Their intellectual contributions, stimulating discussions, and camaraderie have profoundly enriched my academic pursuits and personal growth. I am particularly grateful to my friends and fellow researchers at the Robot Perception Lab, who have made my doctoral journey not just memorable, but truly unforgettable. Our lab group meetings were a constant source of invaluable feedback, critical insights, and creative inspiration, significantly shaping and elevating the work presented in this thesis. The bonds forged and experiences shared in this collaborative environment are ones I will cherish throughout my career and beyond.

Special thanks to Chuck Whittaker and Tim Angert for their technical expertise and assistance, which were crucial in overcoming numerous experimental challenges during my research. I gratefully acknowledge the financial support provided by the Office of Naval Research, and the National Oceanographic Partnership Program under awards N00014-16-1-2103, N00014-18-1-2843, N00014-18-1-2843, N00014-21-1-2482, and N00014-24-1-2272 which made this research possible. My appreciation also goes to the staff at the department of Mechanical Engineering and the Robotics Institute for their administrative support and assistance throughout my academic journey.

I want to thank my friends, both within and outside academia, who have provided emotional support, laughter, and necessary distractions during the ups and downs of this journey. To my friends like family, Purva and Rohan, thank you for being on this journey with me, which started as we navigated through the Master's program at CMU together. To my parents and sister, Ashok, Resham, and Anupa, thank you for always believing in me and supporting my dreams since even before my CMU journey. I would not have come this far without your constant love and support. And finally, to my fiancée Rachel, whose love, understanding, and support have been an anchor at every step. Meeting her has been a blessing, and the future chapters of life together are eagerly anticipated.

Abstract

Autonomous Underwater Vehicles (AUVs) have become integral tools to solve real world tasks of surveying, mapping and inspection of human made infrastructure, as well as monitoring in natural environments. AUVs are often used in areas considered dangerous for human intervention, making the need for robust methods of perception and navigation paramount. Depending on the task, AUVs vary in size and form-factor which in turn also affects the onboard sensors used. For most infrastructure inspection tasks the most common types of sensors include Doppler Velocity Logs (DVL) and Inertial Measurement Units (IMU) for inertial navigation and acoustic sonars for perception. While these sensors represent the best options for AUVs, they have shortcomings such as often a prohibitively high cost, and a lack of information rich perceptual data when considering imaging sonars. We discuss the challenges faced in underwater localization and mapping with respect to imaging sonars, such as the lack of a general framework which works across different types of sonar makes, and also obstacles to engaging in research due to high setup and operational costs for AUVs.

In this dissertation, we explore localization and mapping techniques designed for cost-effective underwater robots. We begin with looking at the problem of performing accurate Simultaneous Localization and Mapping (SLAM) when using sonar data due to featureless environments causing degenerate situations. I present a factor graph based solution to this problem. We then look at a framework for onboard acoustic localization of multiple ultra-low-cost underwater robots using off the shelf equipment. For improving feature-based SLAM using imaging sonars, I present a pose-supervised network to learn sonar image correspondences called SONIC. Lastly, this work is further expanded in C-SONIC to allow cross-sonar image correspondence. This allows cheaper robots with low frequency imaging sonars to find feature correspondences in maps made with high frequency imaging sonars.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Importance of Underwater Robot Localization and Mapping

Mobile robotics research has advanced to a level where it is possible to deploy these robots in a variety of situations and has enabled them to work autonomously or with minimal human intervention. Examples of such robots are, but not limited to, - robots used for warehouse automation, self-driving vehicles, exploration, inspection, and search and rescue. Most mobile robots are developed keeping in mind the kind of environment people are most likely to find them useful in, such as indoors, on roadways, or in open air areas. However, there is an equally important need for the development for challenging areas, where human presence is usually minimal, be it due to potential hazards or infrequent need of access.

Underwater environments, be it indoor or open water, are an example of such a situation where continued research is important. Despite covering the majority of the Earth's surface, having more artifacts than museums, and 94% of all life on Earth, it remains relatively unexplored. Autonomous robotic exploration would allow us to better monitor offshore infrastructure such as undersea cables, oil pipelines and drill rigs, and also monitor aquatic life continuously, something which is ever more important in today's world with the collapse of ecological harmony.

Robot autonomy is highly pursued as these robots are likely to be more efficient, safe, and adaptable to changes in scenarios. For a robot to be truly autonomous, it needs to possess the capability of estimating its location in known or unknown environments. This is commonly referred to as *localization.* In unknown environments, it may be efficient for the robot to make a note of the layout of its environment, landmark positions and objects in its vicinity. This process is called *mapping.* The ability of a robot to perform localization and mapping enables the robot to perform several tasks such as navigating to points of interest, planning paths for best viewpoints, and returning to visited areas efficiently. In other words, this allows the robot to perform tasks and interact with the environment in a meaningful way.

1.2 Sensing when Underwater

As in the case of most autonomous robots, remotely operated vehicles (ROVs) and autonomous underwater vehicles (AUVs) also need a combination of sensors for navigation and perception. The sensor selection is dependent on the suitability of use when under water. For example, the attenuation of electromagnetic signals under water makes the use of GPS or other wireless signal based localization methods impossible unless when at the surface. Underwater robots are thus commonly equipped with inertial measurement units (IMUs) or inertial navigation systems (INSs), which use accelerometers and gyroscopes to measure translational accelerations and rotational velocities of the vehicle. To measure translational velocities in a plane, AUVs can be equipped with Doppler velocity logs (DVLs). Similarly, pressure sensors can provide a depth estimate to provide a vertical measurement. A fusion of these sensors can be integrated over time to provide state estimate based on dead reckoning. However as these are interoceptive sensors, the pose estimates they provide are prone to drift from the true pose of the robot. As such, the use of exteroceptive sensors to observe external objects, such as landmarks, are used to correct for the sensor drift.

For most open-air applications, robots are equipped with exteroceptive sensors such as optical cameras and laser-based lidars. These sensors are able to capture vast amounts of information, whether from two-dimensional pixel information, or spatial 3D features from lidars and stereo camera pairs. Underwater robots too have specialized versions of watertight cameras and lidars [70]. While these sensors are very useful, and have an abundance of research support from open air robotic platforms, they are plagued with loss of useful information in water due to attenuation of visible and infrared spectrum in water, and also turbidity in open water areas, especially in areas near the shoreline. This limited visibility severely shackles the usefulness of these sensors.

Thus, acoustic sensors are often the logical sensor of choice for underwater applications. Sonars have been used for decades on ships as well as remotely operated vehicles (ROVs) as well as AUVs, mostly for bathymetry and survey purposes. The commonly used sonar types are the side scan (SSS) and synthetic aperture sonars (SAS). These are typically long range sonars, meant to capture as large an area as possible with each scan, while compromising on resolution. On the other hand, imaging sonars, also referred to as acoustic cameras or forward looking sonar (FLS), are a different class of sonar which focus on short range, but higher resolution. This makes imaging sonars ideal for tasks such as inspection, 3D reconstruction and high resolution surveys.

Acoustic beacons are commonly used for underwater localization and communication. Depending on the travel-time system used, there are typically multiple beacons with transmitters and or receiver, with the robotic platform having its own module as well. Popular methods are the long-baseline (LBL), short-baseline (SBL), and ultrashort-baseline (USBL) methods. Recently, more development has gone into the development of inverted USBL methods (iUSBL) which allow for one-way travel time localization and communication. These will be expanded upon in later chapters.

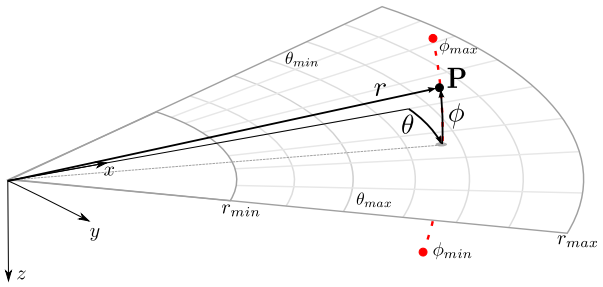


Figure 1.1: The basic imaging sonar sensor model of a point feature. Each pixel provides direct measurements of the bearing / azimuth (θ) and range (r), but the elevation angle (ϕ) is lost in the projection onto the image plane - analogous to the loss of the range in the perspective projection of an optical camera. The imaged volume, called the frustum, is defined by the sensors limits in azimuth $[\theta_{min}, \theta_{max}]$, range $[r_{min}, r_{max}]$, and elevation $[\phi_{min}, \phi_{max}]$. Reprinted from [90].

1.3 Barriers to Underwater Robotics Localization and Mapping Research

Underwater environments are possibly the most challenging for autonomous robots. While cameras and lidars remain the most widely used and supported sensing modalities for in-air platforms, they are largely ineffective underwater. Due to attenuation and turbidity, these sensors have visibility limited to a few meters at best. Although imaging sonars are used underwater to circumvent the issue of visibility in turbid waters, there are several drawbacks associated with them as also described by [57, 30]. The drawbacks can be summarized to:

- Noise: Imaging sonars, like other acoustic sensors suffer from low signal to noise ratios. There is also persistent speckle noise, which arises from coherent and random patterns of constructive and destructive interference of back-scattered signals, creating pixels with low and high intensity in the image[33, 19, 38]. This greatly effects image quality, making it harder to get well defined features.
- Low and nonuniform resolution: Common imaging sonars like the SoundMetrics DIDSON[79] and the Blueprint Subsea Oculus M1200d[5] that are used in our experiments have their image resolutions determined the range bins and azimuthal beams. The maximum resolution achievable is close to 250,000 pixels. In contrast, even cheap consumer grade cameras are able to give several million pixels in information. Aside from the number of pixels, the pixel footprints are not spread out uniformly. As seen in Figure 1.1, as the range increases, the footprint of the pixel extends, resulting in lower resolutions at larger ranges.
- Acoustic artifacts: Similar to how speckle noise is formed, it is possible that cross-talk between transducers can result in erroneous measurements in the images. The bounce back of acoustic waves from multiple surfaces before returning to the transducers (multipath reflections) also generates incorrect bearing and range measurements, often looking like reflections of the true object.
- Variance from viewpoint and object materials: Images produced by optical cameras pos-

sess a property called as photometric consistency. This means that points or objects viewed from different angles would provide similar color frequencies or intensities across views. This property is what makes multi-view geometry and structure from motion using optical cameras possible. However when it comes to sonar images, pixel intensities are dependent on the viewing angle. That is the angle between the sonar center and the normal from the surface patch being viewed. Aside from the viewing angle, the material of the object also dictates the uniformity of the image. Smooth metallic surfaces are likely to create specular reflection compared to more rough or porous material like concrete.

Keeping aside difficulties in underwater perception, another hurdle for underwater robotics research is the cost associated with it. Inspection and observation class ROVs can be purchased in the range of $15,000 to $250,000. These are suitable for depths upto a few hundred meters. Light work and work class ROV's which can operate in depths beyond thousand meters can range from a quarter million to millions of dollars. The size of the robots, and available budget also dictate the type, accuracy and resolution of onboard sensors. For navigation, onboard DVL or USBL networks can cost anywhere between $10,000 to $100,000 and more. Similarly, sonars can also come in the range of $6000-$300,000. While the cheaper alternatives are cost-effective, they can often come at the compromise of resolution and accuracy.

In addition to this, there are operating costs for field experiments, which include hiring a boat and crew to deploy the robot. The bigger the ROV, the more equipment and hands would be needed. Unsurprisingly, this limits a large chunk of research capabilities to a small section of researchers who have the monetary support for supporting this research.

1.4 Contributions

As broached upon in the previous section, cost is a significant barrier to underwater robotics research. As such, the focus of this document is towards algorithmic techniques and tools developed for cheaper underwater robots to perform localization and mapping at levels comparable to more expensive ROVs and AUVs.

As such the contributions of this thesis can be summarized as follows:

- A degeneracy-aware simultaneous localization and mapping (SLAM) framework, capable of negating the effects of inaccurate state estimates from navigational sensors.
- A multi-robot, on-board, pseudo-ranging based acoustic localization framework, developed for small, low-cost bio inspired robotic fish.
- A learned feature descriptor for imaging sonar, allowing for noisy, lower resolution images to be sufficient for feature-based SLAM.
- Expanding the learned feature descriptor to work for different sonar frequencies and parameters, allowing for robots with lower resolution sonars to use maps made by high resolution maps to localize themselves.

1.5 Outline of Thesis

In **Chapter** 2 we first go over preliminaries of the fundamental mathematics and models we use throughout our developed techniques. We present the maximum a posteriori estimation framework and inference using factor graphs. We then introduce the imaging sonar sensor model, which is the sensor a majority of this thesis work is prepared for.

In **Chapter** 3 we describe our solution to failure in SLAM optimization arising from degeneracies in feature-poor areas, which are very common in underwater environments. To this effect we introduce a two pronged approach, first, a degeneracy aware iterative closest point algorithm for scan matching, and second, a degeneracy aware factor for loop closures. This chapter is based on work from Hinduja et al. [24].

In **Chapter** 4 we explore the capability of low-cost and low-power underwater robots to localize themselves without an external observer and present our acoustic localization method inspired from global positioning systems. This chapter is based on work from Hinduja et al. [25].

In **Chapter** 5 we look at semi-supervised, learned methods to generate feature descriptors for imaging sonar by leveraging sensor pose information. This would help perform more robust feature-based SLAM considering most traditionally used feature point descriptors are formulated keeping optical images in mind. This chapter is based on our work, SONIC [18].

In **Chapter** 6 we look towards extending the work from the previous chapter, focusing on cross-sonar feature matching. The network is able to provide correspondences between images differing in frequency modes as well as fixed parameters like maximum range.

Finally, in **Chapter** 7 we look at discussions, and directions for the future of the works presented in this thesis.

Chapter 2

Foundations

In this chapter, we provide an overview of probabilistic inference and factor graphs for SLAM as well as introduce the imaging sonar sensor model. To interact with and make meaningful decisions in the world, a robot needs to infer information about its environment through its sensors by relying on a priori knowledge. Observations or measurements recorded by sensors are provided periodically at each time step. Individually, these measurements are typically noisy and do not reveal enough useful information. However, robots can accumulate and combine multiple such measurements to determine the "state" of the robot in the world. Due to noise and uncertainties in the measurements received from the sensors, it is difficult to obtain the true state, but we can obtain a probabilistic inference of the state. The state is typically modeled as a belief over continuous, multivariate random variables $X \in \mathbb{R}^n$. In SLAM, we use sensor measurements Z to infer the unknown states X which are the unknown robot and landmark positions.

2.1 Notation

We will attempt to adhere to the following notation in our mathematical equations throughout the rest of the document. Bolded capital letters (e.g. $\mathbf{A}$) will denote matrices, and bolded lower-case letters (e.g. $\mathbf{v}$) will denote vectors. Scalars are represented with lower-case letters (e.g. c). We use capital letters (e.g. X) to denote sets. 6-DOF pose variables are denoted as x_i, with an appropriate subscript.

If any change in notation specific to a Chapter occurs, it will be explicitly stated so.

2.2 Maximum a Posteriori Estimation

We will now introduce the nonlinear least squares optimization that is used to solve the maximum a posteriori (MAP) formulation of a SLAM problem. The objective of MAP estimation is to achieve the most likely state X of a system given a set of measurements Z. We begin by

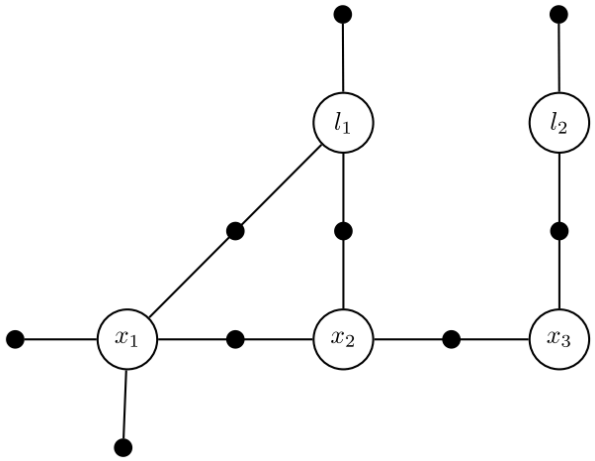


Figure 2.1: Factor graph representation of a toy SLAM problem. Here the state X is comprised of poses x_1, x_2, and x_3 as well as landmarks l_1 and l_2. The measurements are represented by the smaller black vertices and are not labeled for simplicity. Figure reprinted from [9].

maximizing the posterior of the state:

$$
\begin{aligned}
X^* &= \underset{X}{\operatorname{argmax}}\ p\left(X|Z\right) && (2.1)\\
&= \underset{X}{\operatorname{argmax}}\ p\left(X\right) p\left(Z|X\right) && (2.2)\\
&= \underset{X}{\operatorname{argmax}}\ p\left(X\right) l\left(X;Z\right) && (2.3)\\
&= \underset{X}{\operatorname{argmax}}\ p\left(X\right) \prod_{i=1}^{N} l\left(X;\mathbf{z}_i\right) && (2.4)
\end{aligned}
$$

where $l\left(X;Z\right) \propto P\left(Z|X\right)$ represents the likelihood. Here we assume conditional independence of measurements, which is encoded in the connectivity of the factor graph. Note that although we use the notation $p\left(z_i|X\right)$, the measurement $\mathbf{z}_i$ is only conditioned on the subset of variables from the state X to which it is connected in the factor graph. The prior $p(X)$ can be dropped on the assumption that we have no prior information about the state. We also assume additive Gaussian noise in the measurement model such that $l\left(X;\mathbf{z}_i\right) \propto p(\mathbf{z}_i|X) = \mathcal{N}(h_i(X), \mathbf{\Sigma}_i)$ and can be expanded as:

$$
l(X;\mathbf{z}_i) \quad \propto \quad \exp\left\{-\frac{1}{2}\left\|h_i(X) - \mathbf{z}_i\right\|^2_{\mathbf{\Sigma}_i}\right\} \tag{2.5}
$$

Here $h_i\left(X\right)$ is function that predicts a value $\hat{\mathbf{z}}_i$ of the measurement $\mathbf{z}_i$ based on the state estimate X. To represent the uncertainty in the measurement $\mathbf{z}_i$, we use the covariance matrix $\mathbf{\Sigma}_i$ which may be derived from sensor specifications or found empirically. The monotonic logarithm function and Gaussian noise model allow us to simplify the optimization into a nonlinear least

squares (NLS) problem:

$$X^* = \underset{X}{\operatorname{argmin}} - \log \prod_{i=1}^{N} l\left(X; \mathbf{z}_i\right) \tag{2.6}$$

$$= \underset{X}{\operatorname{argmin}} \sum_{i=1}^{N} \left\| h_i\left(X\right) - \mathbf{z}_i \right\|_{\mathbf{\Sigma}_i}^2 \tag{2.7}$$

where the notation $\|.\|_{\mathbf{\Sigma}_i}^2$, is the Mahalanobis distance with covariance $\mathbf{\Sigma}_i$.

While many iterative solvers are available, the Gauss-Newton algorithm is most commonly used to solve the NLS problem in Equation 2.7. The prediction function can be linearized as

$$h_i\left(X\right) = h_i\left(X^0 + \mathbf{\Delta}\right) \approx h_i\left(X^0\right) + \mathbf{H}_i\mathbf{\Delta} \tag{2.8}$$

$$\mathbf{H}_i = \left.\frac{\partial h_i\left(X\right)}{\partial X}\right|_{X^0} \tag{2.9}$$

where X^0 is an initial state estimate, $\mathbf{\Delta} = X - X^0$ is the state update vector and $\mathbf{H}_i$ is the Jacobian matrix of the prediction function $h_i\left(X\right)$. This approximation can be substituted into Equation 2.7 as

$$\mathbf{\Delta}^* = \underset{\mathbf{\Delta}}{\operatorname{argmin}} \sum_{i=1}^{N} \left\| h_i\left(X^0\right) + \mathbf{H}_i\mathbf{\Delta} - \mathbf{z}_i \right\|_{\mathbf{\Sigma}_i}^2 \tag{2.10}$$

$$= \underset{\mathbf{\Delta}}{\operatorname{argmin}} \sum_{i=1}^{N} \left\| \mathbf{H}_i\mathbf{\Delta} - \left(\mathbf{z}_i - h_i\left(X^0\right)\right) \right\|_{\mathbf{\Sigma}_i}^2 \tag{2.11}$$

$$= \underset{\mathbf{\Delta}}{\operatorname{argmin}} \sum_{i=1}^{N} \left\| \mathbf{A}_i\mathbf{\Delta} - \mathbf{b}_i \right\|^2 \tag{2.12}$$

$$= \underset{\mathbf{\Delta}}{\operatorname{argmin}} \left\| \mathbf{A}\mathbf{\Delta} - \mathbf{b} \right\|^2 \tag{2.13}$$

Through whitening, the Jacobian and error vector can be simplified as $\mathbf{A}_i = \mathbf{\Sigma}_i^{-1/2}\mathbf{H}_i$ and $\mathbf{b}_i = \mathbf{\Sigma}_i^{-1/2}\left(\mathbf{z}_i - h_i\left(X^0\right)\right)$. Stacking all the terms of $\mathbf{A}_i$ and $\mathbf{b}_i$ together gives us a single matrix and vector $\mathbf{A}$ and $\mathbf{b}$, respectively. With this linear approximation, the three mandatory components required from each ith factor are summarized as:

1. The square-root information matrix $\mathbf{\Sigma}_i^{-1/2}$, obtained from the covariance matrix.
2. The Jacobian matrix $\mathbf{H}_i$ of the prediction function.
3. And last, the error vector $\boldsymbol{z}_i - h_i\left(X^0\right)$, which is based on the prediction function.

Taking derivative of Equation 2.13 and equating it to zero results in the familiar "normal" equation:

$$\left(\mathbf{A}^T\mathbf{A}\right)\mathbf{\Delta}^* = \mathbf{A}^T\mathbf{b} \tag{2.14}$$

where by using the pseudo-inverse we can solve directly for the current iteration's update $\mathbf{\Delta}^*$:

$$\mathbf{\Delta}^* = \mathbf{A}^\dagger\mathbf{b} = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{b} \tag{2.15}$$

or by using the Cholesky or QR decomposition. The update $\boldsymbol{\Delta}^*$ is applied to compute the updated state estimate, which is used as the linearization point for the next iteration in the Gauss-Newton solver. The solver terminates after the magnitude of the update vector falls below a threshold or after a maximum number of allowed iterations.

2.3 Imaging Sonar Sensor Model

Imaging sonars are active acoustic sensors which work on the principle of measuring the intensity of the reflections of sound waves emitted by it. They are analogous to optical cameras in the sense that they interpret a 3D scene as a 2D image. However, the images provided by these sensors are very different. The imaging sonar sensor model uses a slightly different coordinate frame convention compared to the standard pinhole camera model for optical images. The x axis points forward from the acoustic center of the sensor, with the y axis pointed to the right and z axis pointed downward, as shown in Figure 2.2. This coordinate frame is preferred due to the way the image formation occurs for imaging sonars. Unlike in the pinhole camera model, where the scene from the viewable frustum is projected onto a forward-facing image plane, for imaging sonars the scene is projected into the zero elevation plane, which is the xy plane in the sonar frame.

Following the notation in [30], consider a point $\mathbf{P}$ with spherical coordinates (r, θ, ϕ) - range, azimuth, and elevation, with respect to the sonar sensor. The corresponding Cartesian coordinates are then

$$\mathbf{P} = \begin{bmatrix} X_s \\ Y_s \\ Z_s \end{bmatrix} = r \begin{bmatrix} \cos\theta\cos\phi \\ \sin\theta\cos\phi \\ \sin\theta \end{bmatrix} \tag{2.16}$$

When considering an image formed from the pinhole camera model, an arbitrary pixel location can provide us information like the azimuth and elevation angle, but the range information is ambiguous. Every point lying along the ray in 3D would project onto the same pixel location in the image. In the imaging sonar model, a 3D point is projected onto the image plane, or zero elevation plane as

$$\mathbf{p} = \begin{bmatrix} x_s \\ y_s \end{bmatrix} = r \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix} = \frac{1}{\cos\phi} \begin{bmatrix} X_s \\ Y_s \end{bmatrix}. \tag{2.17}$$

In the projective camera model, each pixel has a corresponding ray that passes through the sensor origin. In contrast, the sonar sensor model has a finite elevation arc in 3D space, as seen in Figure 2.2 as a red dotted line. A pixel in a sonar image can provide us the azimuth and range information of the arc, but loses the elevation of the arc entirely, just like how range information is lost for the projective camera model. This nonlinear projection adds significant complexity, coupled with the very narrow field of view most imaging sonar sensors have makes seemingly straightforward tasks for optical cameras appear rather difficult for imaging sonars. Another complication arises from the information these pixels hold. For optical images, pixels measure the intensity of the light reflected towards the sensor from a particular surface patch along the corresponding ray. Thus, each pixel would likely correspond to a single unique surface patch. This allows for different viewpoints to have similar pixel values when accounting for small

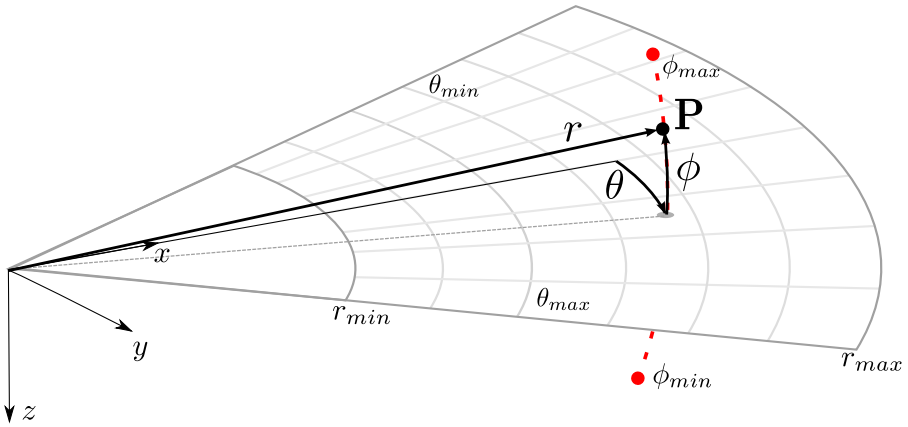


Figure 2.2: The basic imaging sonar sensor model of a point feature. Each pixel provides direct measurements of the bearing / azimuth (θ) and range (r), but the elevation angle (ϕ) is lost in the projection onto the image plane - analogous to the loss of the range in the perspective projection of an optical camera. The imaged volume, called the frustum, is defined by the sensors limits in azimuth $[\theta_{min}, \theta_{max}]$, range $[r_{min}, r_{max}]$, and elevation $[\phi_{min}, \phi_{max}]$. Reprinted from [90].

motions of the camera origin. On the other hand, when it comes to acoustic images, each pixel need not correspond to a single surface patch. All surfaces along an elevation arc may reflect sound emitted by the sensor, and as such a single pixel may contain the compounded intensity of multiple surfaces, which may not be consistent even with slight changes to the sensor origin.

2.4 Test Vehicle Configuration

Our experiments were performed using the Hovering Autonomous Underwater Vehicle (HAUV) [16]. The HAUV is equipped with five thrusters enabling control of all degrees of freedom barring roll and pitch. For onboard navigation, the HAUV houses a 1.2MHz Teledyne/RDI Workhorse Navigator Doppler velocity log (DVL), a Paroscientific Digiquartz depth sensor, and an attitude and heading reference system (AHRS) fitted with a Honeywell HG1700 IMU. The onboard computer of the HAUV is capable of providing odometry estimates by fusing readings from these sensors. The depth sensor, and AHRS are capable of providing highly accurate and drift free measurements of Z, pitch, and roll. However, in comparison the measurements for X, Y and yaw rotation are subject to drift as they are estimated through dead reckoning from the DVL and AHRS readings. For the purpose of mapping, we utilize a dual-frequency identification sonar (DIDSON) [79]. This sonar has 96 azimuthal beams and 512 range beams, using a 2D transducer array.

2.4.1 On-board odometry

A proprietary navigation algorithm fuses measurements from the AHRS, DVL, and depth sensor to provide odometry estimates for the vehicle, in the coordinate frame of the DVL sensor. It

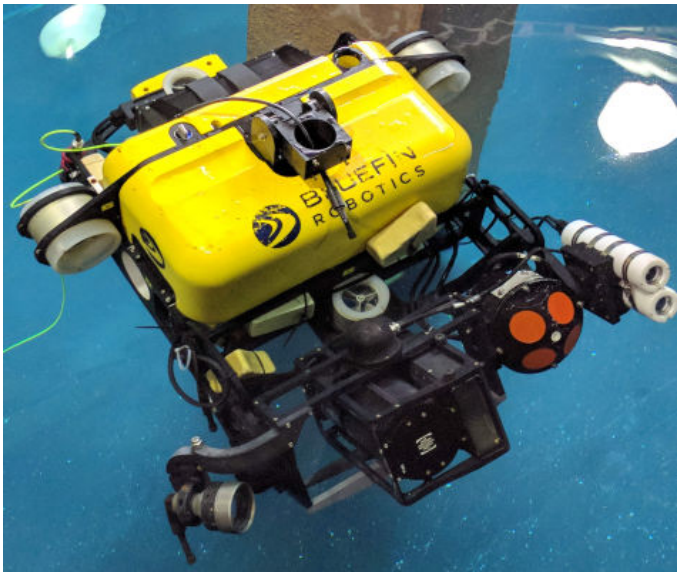


Figure 2.3: Bluefin Hovering Autonomous Underwater Vehicle (HAUV)

is important to note that the depth sensor gives direct measurements of the vehicle's depth, or Z position. The AHRS is also capable of producing very accurate, drift-free estimates of the vehicle's pitch and roll angles by observing the direction of gravity. The X and Y translation and yaw rotation are not directly observable, and are therefore estimated by dead reckoning of the DVL and IMU odometry measurements. The pose estimate will inevitably drift in these directions over long-term operation. This naturally leads to a formulation that treats the vehicle's odometry measurements as two separate types of constraints: a relative pose-to-pose constraint on XYH motion (**X** and **Y** translation and **H**eading / yaw) and a unary ZPR constraint (**Z** position, **P**itch and **R**oll rotation) [83].

For these XYH and ZPR measurement factors, we make use of the Euler angle representation of 3D rotations to represent a pose x_i as the 6-vector $\left[\psi_{x_i}, \theta_{x_i}, \phi_{x_i}, t^x_{x_i}, t^y_{x_i}, t^z_{x_i}\right]^\top$, where ψ_{x_i}, θ_{x_i},ϕ_{x_i} and are the roll, pitch, and yaw (heading) angles, respectively. All Euler angles are normalized to the range $[-\pi, \pi)$ radians.

Chapter 3

Degeneracy-aware Factors for Pose Graph SLAM

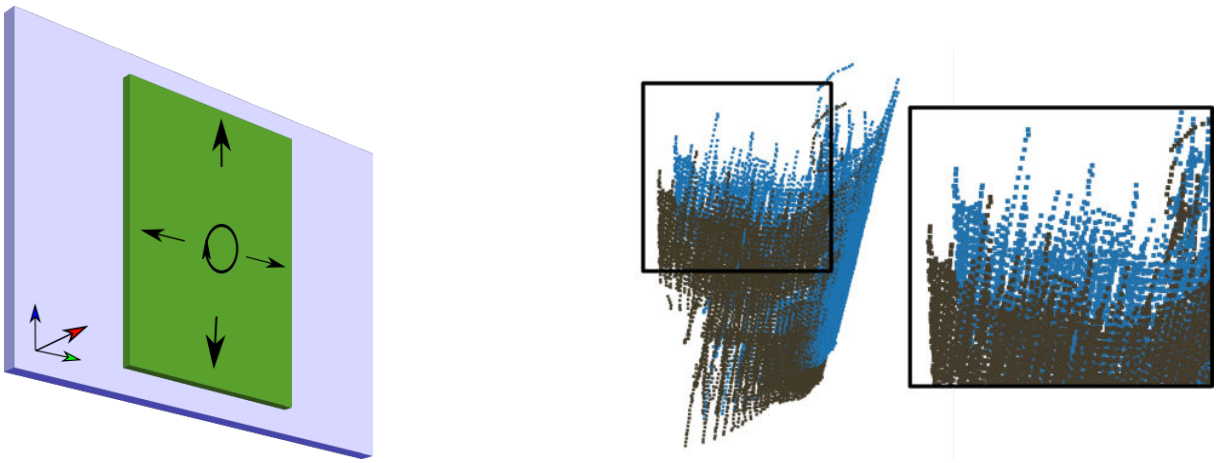
3.1 Problem Statement

SLAM systems utilizing laser range finders or RGB-D sensors have become commonplace over the past decade. A multitude of algorithms have been developed using these sensors to provide accurate state estimates and consistent maps in real-time [95, 94, 102], without the need for absolute measurements from sensors such as a GPS. However, most of these methods are prone to failure in certain scenarios. One particular failure mode is when the scene in consideration has degenerate geometry or insufficient features for the method to get correspondences. Cadena et al. [6] note that the absence of observable features, even temporarily, can hinder the SLAM solution and necessitates the need for techniques to deal with the lack of observability. There are broadly two approaches to handling degenerate situations in SLAM, which can be categorized as either (1) limiting or (2) actively handling degenerate situations. Limiting the occurrence of degenerate situations can be achieved by effectively supplementing the system with more information to minimize failure. On the other hand, actively handling degeneracy in a SLAM system requires analyzing the constraints imposed by measurements, identifying the degeneracies, and appropriately compensating for them. Our approach focuses on actively handling degeneracy and the contributions of this work presented are as follows:

1. A degeneracy-aware point-to-plane ICP algorithm, based on the solution remapping technique [101].
2. A new degeneracy-aware partial factor, which allows for the optimization of the pose graph only in the well constrained directions, by using the results from the degeneracy-aware ICP.

3.2 Related Work

Limiting degenerate situations: As highlighted in Section 3.1, supplementing the system with added information can reduce the occurrence of degeneracy. Hsiao et al. [27] use an IMU in their planar SLAM framework to maintain the sensor pose for short durations when faced with



(a) (b)

Figure 3.1: Examples of degeneracy:- (a) Planar registration: The smaller plane while constrained in X (into the plane), pitch and yaw, is unconstrained in the Y, Z and roll directions. (b) Degeneracy in mapping simulated pilings: The optimization is unconstrained in the vertical direction as well as along the curved surface, resulting in an incorrect registration when using point-to-plane ICP.

texture-less scenes. In their point-plane SLAM system, Taguchi et al. [82] analyze the possible correspondences found to satisfy certain properties, and reject the pairs likely to be degenerate based on those properties. Tribou et al. [84] describes the use of multiple cameras with different fields of view such that at a given time at least one camera's observation can be used for localizing the system effectively, even if in a visually poor environment. Other ways to limit degeneracy could be by switching to different features or methods. For the mapping of texture-less pipe systems, Ma et al. [52] substitutes visual features with voids detected via ultrasonic scanners as landmarks, effectively avoiding degeneracies which would occur through visual measurements. Hu et al. [28] switch between feature matching methods for RGB-D and RGB to deal with different environments during mapping.

Active handling of degenerate situations: These techniques work by identifying degenerate situations and dynamically compensating for them. Pathak et al. [63] describe a novel method of scan matching using planes, where they determine and minimize the geometric uncertainty of the configuration space. Cho et al. [8] build upon Pathak et al.'s method and propose the detection of degeneracy between two sets of plane features by comparing the ratios of eigenvalues in the second moment matrix. They then project the value of the IMU recorded orientations in the directions estimated to be degenerate, which is then used to optimize their state estimate. Rong et al. [71] use the eigenvalues of the Empirical Observability Gramian to represent the observability of the system parameters. They then use the local condition number to determine the degeneracy online, and detect future motions which may result in degenerate cases. As Zhang et al. [101] define, an optimization problem in SLAM is degenerate if there exist one or more directions in the state space of the optimization which are not well constrained. However, even if some directions in the optimization are degenerate, the well constrained directions can still be used to update in a subspace of the original optimization. They separate the degenerate

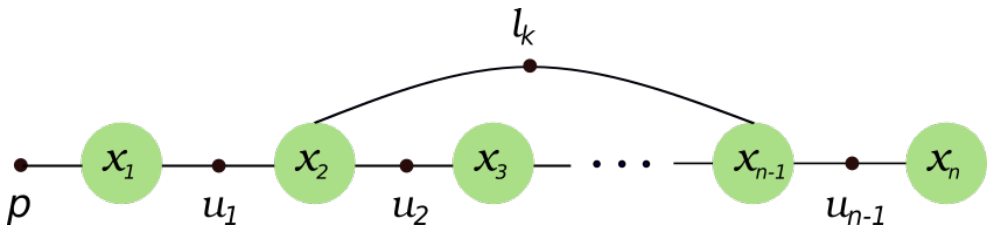


Figure 3.2: Pose graph representation of a SLAM problem: Green circles represent the pose nodes x_k to be estimated. The small black dots represent measurement factors (odometry u_i or loop closures l_k) that are connected through edges to one or more pose nodes. The prior p constrains the first pose to remove a gauge freedom.

and non-degenerate directions and introduce a technique called solution remapping to perform incremental updates only in the non-degenerate directions.

3.3 Pose Graphs for SLAM and Iterative Closest Point

In Section 2.2, we introduced MAP formulation for SLAM to be solved as a least-squares optimization. This optimization can be represented in graphical form using a pose graph or factor graph [9]. Pose graphs are a special form of factor graphs where all variable nodes are robot poses, and the factors represent pose-to-pose constraints. A typical pose graph is shown in Fig. 3.2 with robot poses $x_1, ..., x_n$. Between adjacent poses we have pose-to-pose odometry constraints, $u_1, ..., u_{n-1}$. Loop closure constraints l_k connect two arbitrary poses. The variables nodes in the graph are estimated using nonlinear least-squares using Gauss-Newton or a similar algorithm.

Equation 2.7 gives us the general nonlinear least squares form of the pose graph optimization. We now represent an expanded version of the nonlinear least squares formulation for the SLAM system represented in Fig. 3.2. We assume a motion model $\mathbf{f}(\mathbf{x}, \mathbf{u})$, where $\mathbf{u}_i = u_1, u_2, ..., u_n$ are the added odometry factors between poses x_{i+1} and x_i, along with a measurement model for loop closures $\mathbf{g}(x_i, x_j)$. Then, for a sequence of poses $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$, the MAP estimate can be written as:

$$\begin{aligned} \mathbf{x}^* \quad = \quad & \underset{\mathbf{x}}{\operatorname{argmin}} \Big(\sum_{i=1}^{N} \left\| \mathbf{f}_{i,i+1}\left(\mathbf{x}_i, \mathbf{x}_{i+1}\right) - \mathbf{u}_i \right\|^2_{\mathbf{\Sigma}_i} + \\ & \sum_{i=1}^{N} \left\| \mathbf{g}_i\left(x_i, x_j\right) - \mathbf{l}_{ij} \right\|^2_{\mathbf{\Sigma}_{ij}} \Big) \end{aligned} \tag{3.1}$$

To acquire a loop closure constraint $\mathbf{l}_{ij}$ between poses x_i and x_j, a popular method for measurements from range scanners and RGB-D sensors is the point-to-plane ICP algorithm [7]. Here we introduce the linear least-squares optimization for point-to-plane Iterative Closest Point (ICP) as described by Low [50].

Let $\mathbf{s}_i = (s_{ix}, s_{iy}, s_{iz}, 1)^\top$ and $\mathbf{d}_i = (d_{ix}, d_{iy}, d_{iz}, 1)^\top$ represent the source and their corresponding destination points. $\mathbf{n}_i = (n_{ix}, n_{iy}, n_{iz}, 0)^\top$ is the unit normal vector at $\mathbf{d}_i$. The resulting

error metric can be represented as follows

$$T_{opt} = \underset{T}{\operatorname{argmin}} \sum ((T \cdot \mathbf{s}_i - \mathbf{d}_i) \cdot \mathbf{n}_i)^2 \tag{3.2}$$

where T is the 3D rigid body transformation for transforming the source points such that the total error is minimized. As derived in [50], on further simplification our original metric can be re-arranged into a matrix expression as

$$\mathbf{x}_{opt} = \underset{\mathbf{x}}{\operatorname{argmin}} \, |A\mathbf{x} - \mathbf{b}|^2 \tag{3.3}$$

Here, A is our covariance matrix obtained from our source points and the corresponding normals, and $\mathbf{x}$ is the 6×1 vector of the unknown transformation.

In controlled environments, coupled with only small increments in orientation, point-to-plane ICP converges well when a unique global minimum exists and thus provides an optimal solution. However, complications arise when the optimization might get trapped in a local minimum. This is common in cases of degeneracy due to planar and textureless environments, in which all 6 degrees of freedom (DoF) are not constrained. In these situations the registration either fails completely and aborts, or even worse, gives an incorrect solution in some directions. Visual examples of such degeneracies are shown in Fig. 3.1.

Using the background provided by Equations 3.3 and 2.7, Section 3.4 describes the formulation of the degeneracy-aware ICP and partial loop closure factor, which help in avoiding the situations described above.

3.4 Approach

3.4.1 Degeneracy-Aware ICP and Solution Remapping

As mentioned earlier in the previous section, point-to-plane ICP can provide an incorrect registration due to degeneracy, which can result in a very poor trajectory estimate when the loop closure is added to the pose graph. To combat these errors, we need to detect the degenerate, or poorly constrained, directions of the state space, and make use of only the well-constrained directions in the optimization step for ICP. Zhang et al. [101] show how to iteratively solve a nonlinear system by performing updates only in the well constrained directions. After determining the values of eigenvalues λ_i and eigenvectors v_i from the A matrix of a linearized system, the method of Zhang et al. constructs three matrices V_d, V_f and V. The three matrices are constructed such that V_d contains the degenerate direction's eigenvectors, V_f the fully constrained direction's eigenvectors and V the complete set of eigenvectors. This division is based on the comparison of the individual eigenvalues λ_i against a common threshold λ_t. The linearized system as usual is solved as $\mathbf{x}_f = (A^\top A)^{-1} A^\top b$. Then, this standard solution is projected onto the space of well-constrained directions. This incremental update is added to the state estimate, and the procedure repeats at the new linearization point.

In our degeneracy-aware ICP system, we only use the well conditioned directions to update the optimization solution, and ignore the directions determined as degenerate. The degeneracy-aware ICP algorithm is described in pseudo-code in Algorithm 3.1.

Algorithm 3.1 Degeneracy-aware ICP

Input: Initial relative transformation from odometry T_{odom}
Output: Relative ICP transformation T_{icp}

$T_{icp} \leftarrow T_{odom}$
while nonlinear operations **do**
    Linearize the optimization problem at T_{icp} to get $A^\top A$
    Compute eigenvalue λ_i and eigenvector v_i of $A^\top A$ for $i = 1...6$
    Determine an eigenvalue threshold λ_t
    Construct matrix V containing all the eigenvectors
    Construct matrix V_f containing only well conditioned directions based on λ_t
    Compute update $\Delta x_f \leftarrow (A^\top A)^{-1} A^\top b$
    $T_{icp} \leftarrow T_{icp} + V^{-1} V_f \Delta x_f$
end while
return T_{icp}

The degeneracy-aware ICP was implemented by extending the Point Cloud Library's [73] point-to-plane ICP method. The linearization of the point-to-plane error metric to obtain $A^\top A$ and $A^\top b$ was achieved by the method described in [50]. The fundamental difference between the solution remapping technique presented by [101] and our proposed degeneracy-aware ICP is that at each iteration we re-compute the well conditioned directions on the basis of a different covariance matrix ($A^\top A$). By doing so we acknowledge the fact that at each iteration where we linearize at an updated T_{icp}, the degenerate directions might differ.

3.4.2 Thresholding

To determine which directions are degenerate, we need to set a threshold λ_t to be compared to the eigenvalues we receive from the degeneracy-aware ICP. Zhang et al. [101] use a sample dataset with predetermined degenerate and non degenerate sections, and determine their threshold based on the mid point of the minimum eigenvalue distribution they get. However this method is not suitable for real-time evaluation. Cho et al. [8] compare the ratios of eigenvalues amongst the different directions and set up a ranked list, using their sensor properties to determine their fixed threshold. Similar to [71], we use the condition number, i.e. the ratio between the maximum and minimum eigenvalues, λ_{max} and λ_{min}, for each iteration as our eigenvalue threshold for the loop closure candidates. The benefit of this threshold is largely seen in highly degenerate cases, where λ_{min} might be an extremely small value, along with other directions, thereby making λ_t much larger. This automatically rejects that constraint on account of it being unreliable.

3.4.3 Degeneracy-aware Loop Closure Factor

After obtaining the matrix of fully constrained eigenvectors V_f through the degeneracy-aware ICP, we incorporate this result into the pose graph formulation with our proposed partial loop closure factor. Given a measurement model for loop closures $\mathbf{g}(x_i, x_k)$, and referring to our least squares approach to SLAM optimization in Section 3.3, we can use the matrix V_f as follows:

$$\mathbf{x}^* = \underset{\mathbf{x}}{\operatorname{argmin}} \sum_{(i,k)\epsilon L} \|V_f \left(\mathbf{g}(x_i, x_k) - \mathbf{l}_{ik}\right)\|^2_{V_f \Lambda_{ik} V_f^\top} \tag{3.4}$$

where L is the set of all tuples (i, k) for which we consider loop closure constraints, and Λ_{ik} is the covariance for the registration between x_i and x_k. This formulation of the measurement model enables the generation of a partial factor for the well constrained directions of the tuple (i, k). The covariance for loop closure constraints, Λ_{ik}, is updated at each iteration as $V_f \Lambda_{ik} V_f^\top$ to reflect the selected well constrained directions. Upon incorporating our partial factor, we can perform MAP inference on the resulting pose graph as to determine the value of unknown poses x_i that maximally adhere to the information given the uncertain measurements.

3.5 Application

The degeneracy aware frame work was applied for underwater SLAM for ship-hull inspection. Using the HAUV platform with the DIDSON sonar presented in Section 2.4. We modify the sonar to be used in "profiling" mode by fitting a concentrator lens which reduces its vertical field of view to 1°. Similar to [26, 83], we assimilate a fixed number of these sonar profile scans to form a single submap, which is used as the basis of our mapping system. The following subsections detail how the pose graph, and related parameters, including factors were set up.

3.5.1 Pose Graph Formulation

For each time instance t_i, we have a new pose estimate x_i and sonar scan S_i available. The odometry between two subsequent scans is given by the difference between their poses

$$u_{i,i+1} = x_{i+1} \ominus x_i \tag{3.5}$$

where $\ominus$ expresses the first pose in the local coordinate frame of the second pose.

As shown in Fig. 3.3, our factor graph is formulated such that each subsequent pose x_i, x_{i+1}, ... is represented as a node in the graph, and the odometry constraint between consecutive poses, $u_{i,i+1}$, is a factor that is connected to these pose nodes. The second row depicts the submap formulation where we combine a fixed number of scans together to form a submap. Each individual scan is registered to the coordinate frame for the first scan and is referred to as the base pose of submap m_i. The base poses for consecutive submaps m_i and m_k are connected via a factor representing the accumulated odometry constraints for submap m_i, $u_{i,k}$. The third row shows our degeneracy-aware loop closure factor $l_{i,j}$ that can be added between any two submaps satisfying the necessary conditions [83].

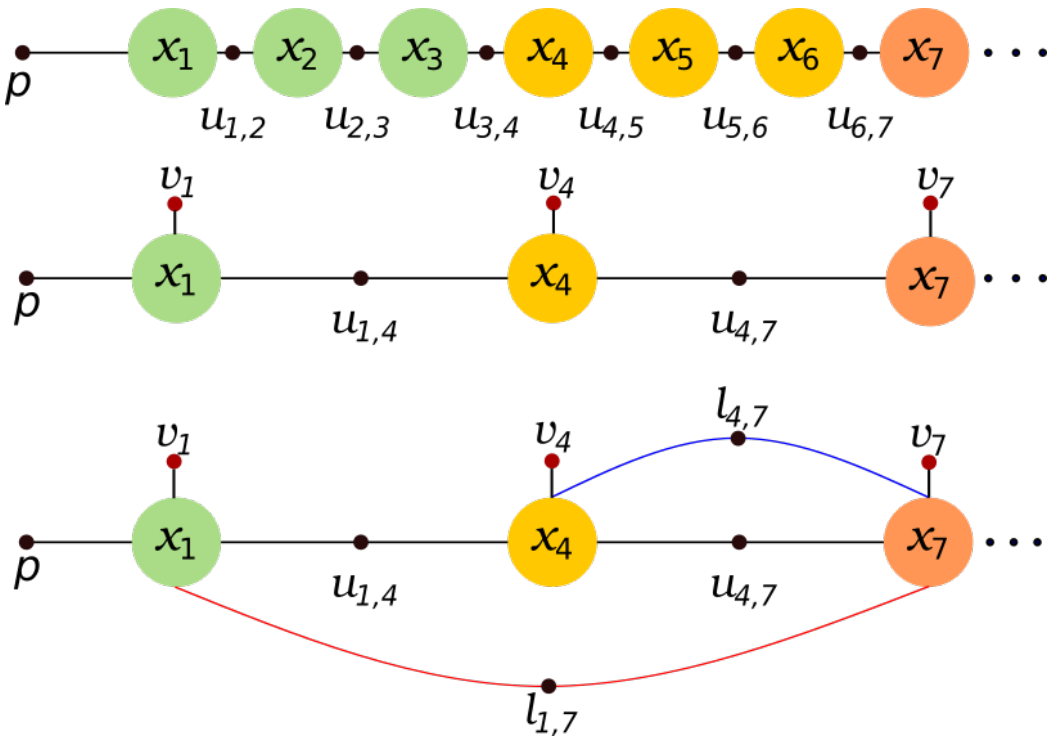


Figure 3.3: Pose graph formulation for underwater SLAM

We use an Euler-angle representation for a pose x_i as a vector $\left[\phi_{x_i}, \theta_{x_i}, \psi_{x_i}, t^x_{x_i}, t^y_{x_i}, t^z_{x_i}\right]^\top$, where $t^x_{x_i}, t^y_{x_i}$ and $t^z_{x_i}$ represent the translation in X, Y and Z, and ϕ_{x_i}, θ_{x_i}, and ψ_{x_i} are the heading, pitch, and roll angles respectively. Our odometry factors are split into $u_{i,i+1}$ and v_i. $u_{i,i+1}$ is a 3 degree of freedom (DoF) XYH factor, represented as $u_{i,i+1} = [t^x_{u_{i,i+1}}, t^y_{u_{i,i+1}}, \phi_{u_{i,i+1}}]$. v_i is a unary ZPR constraint linked to the base pose of the submap m_i, represented as $v_i = [t^z_{v_i}, \theta_{v_i}, \psi_{v_i}]$. A more detailed representation of the XYH and ZPR factors can be found in [93]. We use the iSAM library [37] for our factor graph framework optimization.

3.5.2 Odometry Prior

As previously mentioned in Section 2.4, the HAUV's AHRS and depth sensor give highly accurate and drift free measurements. To take further advantage of this information, we take the readings for depth $t^z_{x_i}$, pitch θ, and roll ϕ and improve the performance of the 6 DoF ICP algorithm by biasing the transform with a priori odometry information.

Section 3.3 introduced the linearization of the ICP transformation T_{icp} between two poses x_i and x_j, which can be represented as an over determined system $A\mathbf{x} = \mathbf{b}$.

We use $\mathbf{\Delta}$ to represent the vector difference between the depth z, pitch θ, and roll ϕ of the two poses. Thus $\mathbf{\Delta} = (t^z_{x_j} - t^z_{x_i}, \theta_{x_j} - \theta_{x_i}, \phi_{x_j} - \phi_{x_i})$. Let $\boldsymbol{X}_{pred}$ denote predicted measurements for the same $\mathbf{X}_{pred} = (t^z_{pred}, \theta_{pred}, \phi_{pred})$.

To add the prior odometry information in this case is equivalent to adding more constraints to the original matrix A and vector $\mathbf{b}$ from our linearized system. Let the matrix P and vector $\mathbf{d}$

be used to encode the prior constraints. Through whitening we simplify to get [9]:

$$P = \Sigma^{-1/2}\mathbf{X}_{pred}^{\top} \tag{3.6}$$

$$\mathbf{d} = \Sigma^{-1/2}\boldsymbol{\Delta}^{\top} \tag{3.7}$$

Thus, our system can now be represented as:

$$\begin{pmatrix} A \\ P \end{pmatrix} \mathbf{x} = \begin{pmatrix} \mathbf{b} \\ \mathbf{d} \end{pmatrix} \tag{3.8}$$

Upon solving this system, we get a 6 DoF ICP transformation estimation biased towards our odometry prior.

3.5.3 Partial Factor for Loop Closure

Consider two poses x_i and x_j with overlapping submaps which satisfy the conditions for a loop closure to be triggered. The relative transformation between them through odometry measurements can be represented as $l_{i,j}^{odom} = x_i \ominus x_j$. Let the resulting measurement from point-to-plane ICP (with solution remapping) be denoted as $l_{i,j}^{icp}$. As explained in Section 3.4.1, we can find the eigenvalues and eigenvectors of the resulting of $A^{\top}A$ matrix. Let V be the resulting matrix of eigenvectors for all 6 DoF, and V_f the well constrained matrix. To prevent double-counting due to the added odometry prior in the ZPR directions, we introduce a constant matrix C, such that the diagonal for X, Y and yaw are 1, and the rest are 0. Referring to Section 3.4.3, our final loop closure measurement model can be represented as:

$$\mathbf{g}(x_i, x_j) = \left\| V_f(C[(x_i \ominus x_j) \ominus l_{i,j}^{icp}]) \right\|^2_{V_f \Lambda_{ij} V_f^{\top}} \tag{3.9}$$

Our factor is constrained to be a maximum of 3 DoF for the X, Y and yaw directions on account of the matrix C.

3.6 Results and discussion

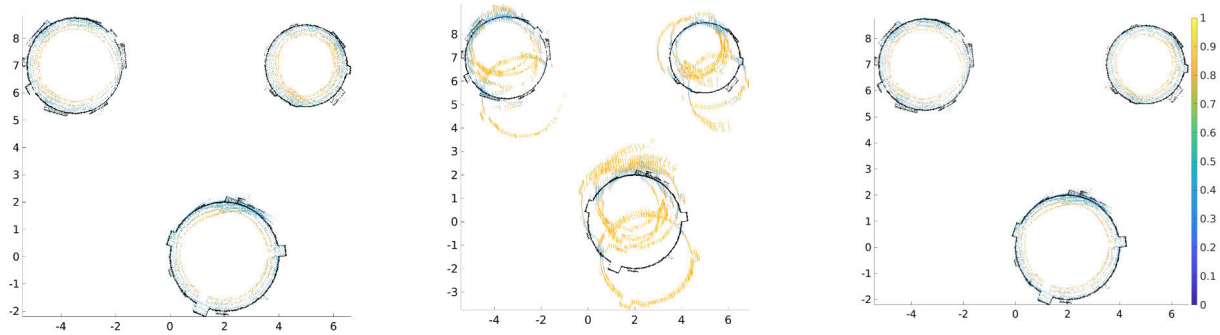
To evaluate the performance of the degeneracy-aware ICP and loop closure factor, we compare the maps generated by: (1) odometry only, (2) point-to-plane ICP with odometry prior (PTP-OP) and full loop closure factor, and (3) our proposed degeneracy-aware ICP with the partial loop closure factor, supplemented with the odometry prior. We perform these experiments on both simulation and real world datasets which are bifurcated into the following two subsections:

3.6.1 Simulated Datasets

The two datasets we used are the simulated pilings and propeller. The pilings dataset in particular is an extreme example of degeneracy given the nature of the loops made around the pilings and submaps formed by the system. We model the motion of our simulated robot to mimic the real HAUV. We also add additional Gaussian noise to the odometry to compare how well the

Table 3.1: RMSE values for simulated datasets. The proposed algorithm gives better results over the standard point-to-plane ICP formulation, as well as considerable improvements over the odometry solution.

Dataset	Odometry	PTP-OP	Degeneracy-aware
Pilings	0.183	0.618	**0.173**
Propeller	0.346	0.429	**0.168**



(a) Odometry solution (b) PTP-OP solution (c) Proposed solution

Figure 3.4: Results for simulated pilings, top down view (RMSE)

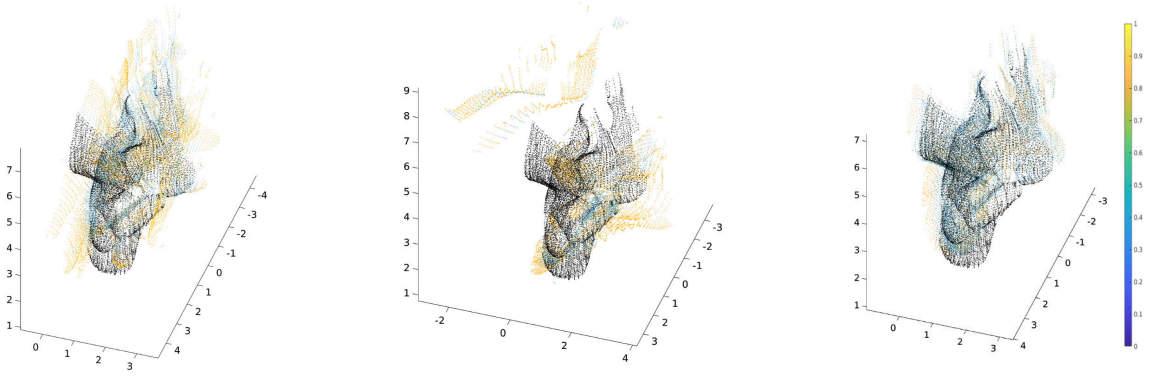
accumulated navigational drift can be corrected. Fig. 3.4 shows the top down view of the simulated pilings reconstruction. Here, the reconstruction in black corresponds to the ground truth, and the SLAM solution is shown in a progressive color scheme from blue to yellow, signifying increasing error.

Fig. 3.5 show the maps generated for the propeller dataset. The Root Mean Square Error (RMSE) comparisons for the two datasets is shown in Table 3.1. The results for our proposed algorithm show a marked improvement over the standard point-to-plane ICP formulation, which tends to substantially drift from the ground truth. For comparison, the PTP-OP results had 34 loop closures, whereas our proposed algorithm only accepted 9 for the simulated pilings dataset. Similarly, for the simulated propeller we see 45 loop closures for PTP-OP and 17 for the proposed algorithm. This is expected on account of the threshold we set, which rejects loop closures with very small λ_{min} values. Both algorithms were provided with identical parameters.

3.6.2 Real World Datasets

Our algorithm was also tested for real world datasets induced with additional Gaussian noise. We compare our results with an odometry only solution as well as results from Teixeira's [83] original submap SLAM formulation.

We first test against a similar dataset used by Teixeira which is the running gear (rudder + propeller) of the *SS Curtis*. We use the odometry only solution, without added noise, as the reference solution for each of the three SLAM solutions which are under additional noise. Fig. 3.6 shows the three results. The reference point cloud is in gray and the 3 solutions are represented



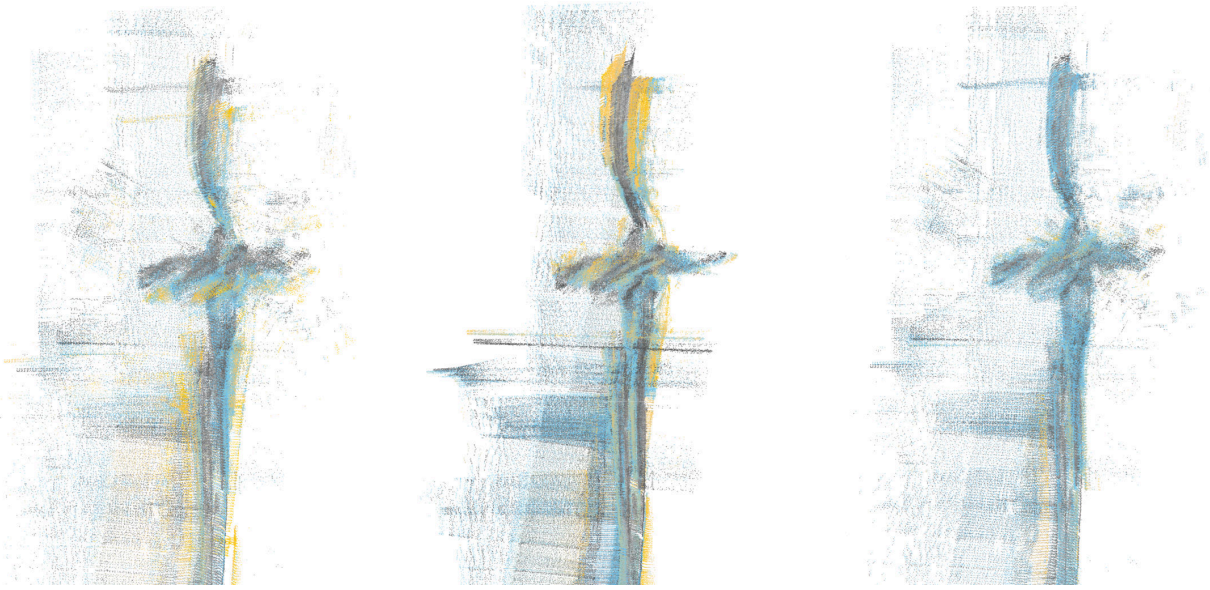
(a) Odometry solution (b) PTP-OP solution (c) Proposed solution

Figure 3.5: Results for simulated propeller (RMSE)

by a progressive color scheme ranging from blue to yellow, representing increasing error from the reference, similar to the simulated datsets.

The second dataset is of pilings from the pier. Fig. 3.7 depicts the environment of the dataset. Each piling is identical and rotational motion of the HAUV to navigate around them increases the chances of degenerate observations. The visual results are shown in Fig. 3.8. The same color scheme as before applies here as well. The original algorithm by Teixeira et al. was specifically formulated for the mapping of ship hulls which are continuous surfaces that are locally mostly planar, explaining the poor result for the pilings. Our method shows its versatility for the mapping of different types of structures, giving consistently better results.

Figure 3.7: Environment in which the pilings dataset was created



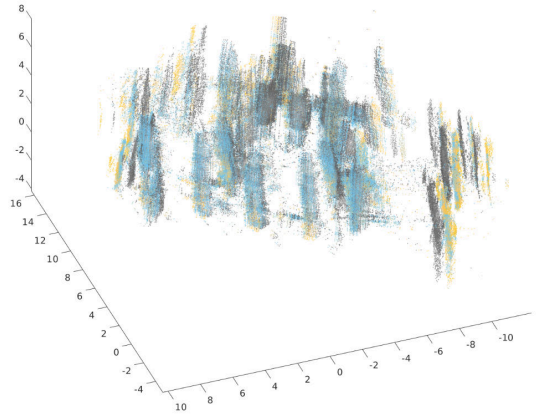
(a) Odometry only solution (b) Texeira et al. solution (c) Proposed solution

Figure 3.6: Results for running gear dataset, bottom up view

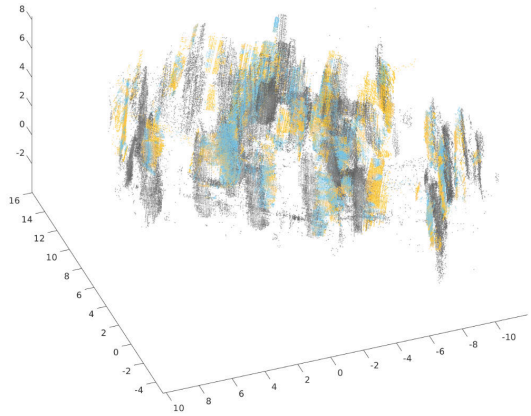
3.7 Conclusion

To create a pose graph optimization robust to degeneracies in the sensor data, we present a degeneracy-aware ICP algorithm for loop closure registration and show how to create partial factors that only incorporate well-constrained dimensions. For evaluation we use an underwater volumetric submap SLAM framework by integrating the degeneracy-aware ICP and loop closure formulation. We demonstrate the ability and robustness of our approach to correct for navigational drift by showing improvements over standard point-to-plane ICP for loop closure registration.

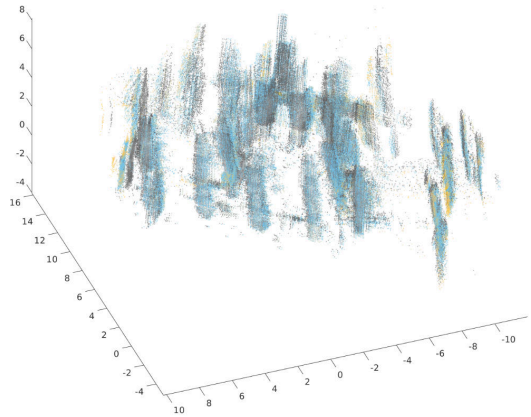
In future work, automatic selection of the threshold λ_t should be investigated. Our experiments have shown that different types of environments benefit from different threshold values, indicating the need to calculate the threshold from properties of the sensor data.



(a) Odometry solution



(b) Teixeira et al. solution



(c) Proposed solution

Figure 3.8: Results for pilings dataset

Chapter 4

Acoustic Localization and Communication Using a MEMS Microphone for Low-Cost and Low-Power Bio-Inspired Underwater Robots

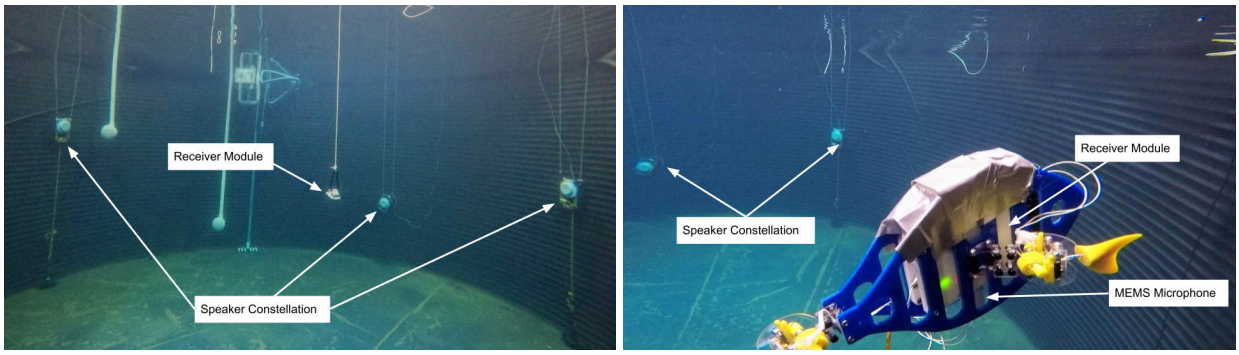
4.1 Problem Statement

Having accurate localization capabilities is one of the fundamental requirements of autonomous robots. For underwater vehicles, the choices for effective localization are limited due to limitations of GPS use in water and poor environmental visibility that makes camera-based methods ineffective. Popular inertial navigation methods for underwater localization using Doppler-velocity log sensors, sonar, high-end inertial navigation systems, or acoustic positioning systems require bulky expensive hardware which are incompatible with low-cost, bio-inspired underwater robots. To solve this problem, we introduce an approach for underwater robot localization inspired by GPS methods known as acoustic pseudoranging. Our method allows us to potentially localize multiple bio-inspired robots equipped with commonly available micro electro-mechanical systems microphones. This is achieved through estimating the time difference of arrival of acoustic signals sent simultaneously through four speakers with a known constellation geometry. We also leverage the same acoustic framework to perform one-way communication with the robot to execute some primitive motions.

4.2 Related Work

Underwater robots are increasingly being used to explore uncharted depths, inspect artificial undersea structures, as well as monitor aquatic life and the physical and chemical properties of their surrounding environment. While larger robots like autonomous underwater vehicles (AUVs) are more suited to tasks such as the inspection of ship hulls, harbors [83] and sub-sea pipelines, bio-inspired robots are better equipped to closely monitor underwater life given their smaller size and lack of moving parts likely to disturb the surrounding environment [68, 62]. Bio-inspired robots

have improved over recent years in aspects of locomotion, control, and communication. In 2014, Marchese et al. [54] presented their novel actuation method for a soft robotic fish capable of agile movement. Soon after in 2015, the group presented a compact acoustic communication module for in-water control of the robot [10]. More recently, an improvement over the previous work was presented by Katzschmann et al. [39] through their soft robotic fish, SoFi, which could be controlled using a handheld acoustic modem based controller presented in the research done by Marchese et al. [10]. The drawback here is that the controller is considerably expensive and can only communicate with one robot at a time. A more recent example of a smaller bio-inspired underwater robot is PATRICK from Patterson et al. [64], which demonstrated the capability of performing closed-loop locomotion planning using an external camera and Bluetooth communication. Here, the use of Bluetooth for passing instructions to the robot limits it to only operate at surface level or shallow water since it requires a non-submerged Bluetooth antenna to receive instructions. However, despite the limitations, these examples demonstrate the promising transition towards autonomy for bio-inspired underwater robots.



(a) Setup for pseudorange localization

(b) Bio-inspired fish robot

Figure 4.1: Water Tank Experiments: (a) Experimental verification of acoustic pseudoranging based localization. Estimated positions are compared to known ground truth positions. (b) A proof-of-concept bio-inspired robot with three actuated fins connected to the receiver module is verified to execute selective motions on receiving specific acoustic signals.

A fundamental necessity for an autonomous robot to navigate and perform specific tasks is the ability to localize itself accurately. Traditionally in robotics, there are systems that can achieve accurate location estimates by utilizing one or more sensors like cameras, lidars, radars, inertial measurement units (IMUs) and the global positioning system (GPS). Underwater robots face unique challenges when it comes to localization. In real-world scenarios, sensors like cameras and lidar are ineffective due to turbidity, and signals from absolute positioning systems like GPS cannot be received underwater. Most commercial AUVs rely on sensors like Doppler velocity logs (DVLs), depth sensors and IMUs for inertial navigation methods to help localize themselves. Sonars and cameras can be used for aiding inertial navigation by leveraging the geographical features of the surrounding environment. As mentioned earlier, cameras are effective only in clear water and sonars often make the cost of underwater vehicles prohibitively expensive. Costs and practicality aside, their weights and bulky form factors limit them to be used only on larger

robots. In this work, our focus is on localization for small, low-cost and low-power, untethered robots.

For localization of smaller and cheaper underwater robots, such as bio-inspired robot fish, acoustic positioning methods have more potential as a viable solution. The oldest and most popular methods of acoustic positioning are long baseline or short baseline (LBL/SBL) and ultra short baseline (USBL) systems [65, 40]. These methods are two-way travel-time (TWTT) methods, which means the beacons and the AUVs need to be equipped with active acoustic systems to communicate with one another. The use of atomic clocks to synchronize both beacons and AUVs can allow for localization using one-way travel-time (OWTT) methods. Earlier approaches achieved range-only OWTT localization and navigation by using filtering or fusion with other onboard sensors [14, 32]. More recently, Rypkema et al. [74] presented their novel method of one-way travel-time inverted ultra-short baseline (OWTT-iUSBL), which allowed for real-time on-board navigation using a single speaker and an array of four passively listening hydrophones on the AUV. Matched filtering and phased-array beamforming on the four received signals from each hydrophone gives a range and bearing estimate of the robot with respect to the speaker position.

While the OWTT-iUSBL can perform effective localization with a relatively cost effective setup compared to commercial AUVs, the components used are still expensive and too heavy to be used on much smaller bio-inspired robots. Atomic clocks, which make OWTT methods possible, are still expensive relative to other options. For a swarm of robots, these costs can add up significantly. Accurate time-of-arrival information is paramount for OWTT methods, and cheaper embedded real time clocks (RTCs) are prone to significant drift and lack the superior precision of atomic clocks. Time-difference of arrival (TDOA) techniques, on the other hand, allow us to avoid the need to know the exact time a signal was sent. This is achieved by instead measuring the differences between the arrival time of multiple signals which were known to be transmitted simultaneously from different beacons. This technique is commonly referred to as *pseudoranging*, and is the foundation of GPS localization [41].

In this paper we present a method for localization of a small robotic fish that is based on acoustic pseudoranging. This is accomplished using cheap, miniaturized, low power sensors and computation. Referring to Fig. 4.2, pseudorange localization and acoustic communication is performed on a fish-inspired robot that swims in a water tank that is instrumented with acoustic speakers and receivers. The robotic fish is embedded with a small and inexpensive micro-electro-mechanical systems (MEMS) microphone that is used for communication and localization through the estimation of TDOA of signals sent simultaneously from the four speakers in the water tank.

As an acoustically passive method on the receiver side, pseudoranging allows for multiple agents to be localized simultaneously without the need for individual signals specifying time of flight (TOF) information or cross communication between transmitter and receiver to achieve time of arrival (TOA). Variations of acoustic pseudoranging have been utilized for the localization of larger underwater vehicles before. Jorgensen et al. [35] presented an observer to estimate several parameters like position, velocity and IMU biases. Leveraging pseudorange measurement differences along with attitude and accelerometer gave better position estimates. A long range underwater navigation algorithm based on acoustic pseudoranging was tested in an area spanning ~275,000km^2 by Mikhalevsky et al. [55] by using GPS assisted beacons. Recent works

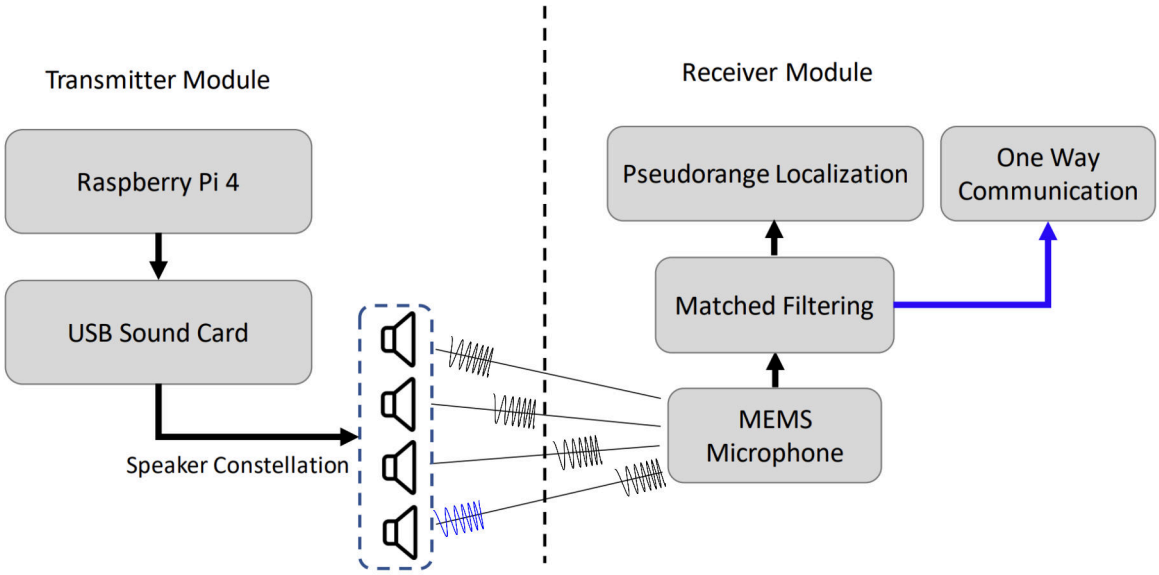


Figure 4.2: System Overview: The transmitter module periodically plays a sequence of identical chirps (black) followed by another, different, chirp (blue) through a constellation of speakers. The receiver module uses the information to estimate position as well as execute basic locomotion tasks.

by Berlinger et al. [3] and Novak et al. [60] have also explored the localization of a swarm of fish, both robotic and natural, through optical and acoustic methods respectively. However both methods perform localization from an external observer and not on-board the agent, which makes acoustic pseudoranging more suitable to use cases where geo-referenced data collection, and on-board navigation are important.

4.3 System Architecture

This section will now describe the hardware and framework for acoustic pseudoranging based localization and one-way acoustic communication. A diagrammatic illustration is shown in Fig. 4.2. There are a minimum of four speakers connected to a single computer playing a sequence of chirps followed by a single, different chirp, periodically. The receiver is time synchronized with the transmitter computer before deployment. The receiver, equipped with a MEMS microphone, records the acoustic signal from the transmitter's speakers, and processes the recorded sequence to obtain the pseudoranges to each speaker. The pseudorange observations are then used to solve for the receiver position and time bias between the receiver and transmitter clock. While an initial time sync is necessary to coordinate when the receiver should expect the signal, having a low-latency or highly accurate synchronization is not required. The same setup can also be used to listen to different types of signals to receive commands for certain tasks. Details on the system hardware and framework are discussed in detail in the subsequent subsections.

4.3.1 System Hardware

4.3.1.1 Acoustic Transmitters

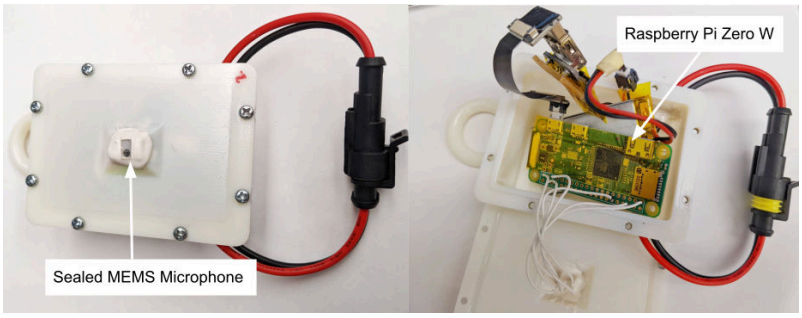
The transmitter module consists of four Lubell UW30 underwater speakers connected to a single Raspberry Pi 4 through a StarTech USB multi-channel sound card. The Raspberry Pi 4 is connected to a wireless network and makes available its local system time through network time protocol (NTP). A four channel WAV file with the same chirp signal in each channel, staggered at known intervals, is played periodically every 10 seconds. The chirp pattern used is a 10 ms, 4.5-8.5kHz linear up-chirp. While using four diverse chirps with different frequency bandwidths, played simultaneously would intuitively be more convenient, having a wider bandwidth and moderately longer chirp length gives better ranging resolution when using matched filters [42]. Thus the linear up-chirp is repeated at intervals of 200 ms across the 4 channels, with the sequence order known beforehand. We also use three shorter chirps in different bandwidths to each other and the aforementioned sequence to communicate different motion behaviors for the robotic fish. One of these shorter chirps then follows the sequential chirps for the purpose of relaying motion instructions to the robot.

4.3.1.2 Receiver and Bio-inspired Robot

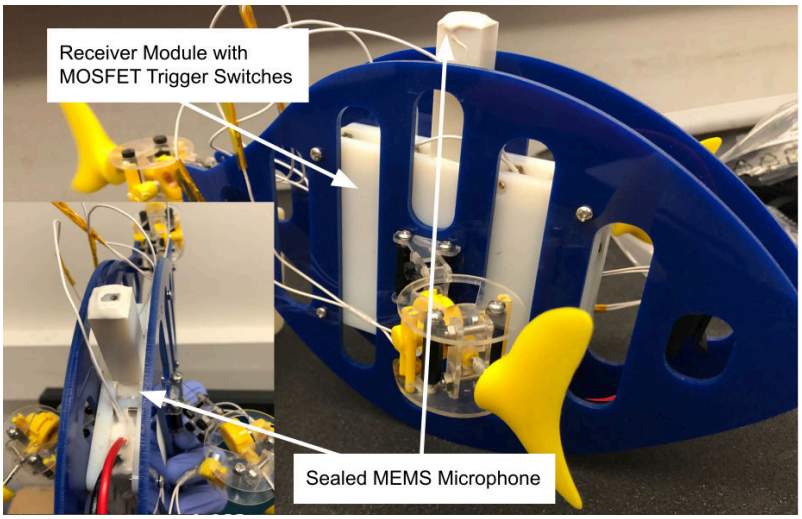
The receiver module runs on a Raspberry Pi Zero W powered by a 5V Li-Poly battery. The only sensor attached is the ICS43434 I2S MEMS microphone. The choice of using a MEMS microphone over commercially available hydrophones was made considering the significantly lower cost, and smaller form factor of the microphones. The receiver module is synchronized with the transmitter clock through NTP over wireless networks before submersion in water. After this point, the inbuilt RTC clock is sufficient to keep track of the periodic signal being sent by the transmitter. The receiver begins recording $2.5s$ long WAV files at each time step. The recording starts before the transmitter module plays the signals to keep ample buffer space to ensure all the chirps are recorded. Accumulated clock drift is corrected by using the time bias calculated in the pseudorange solution. The 1 GHz ARM11 32-bit processor of the Raspberry Pi Zero W takes approximately $6.05s$ for each final location estimate, including the recording duration. Fig. 4.3 (a) shows the static receiver module used for localization experiments. The bio-inspired fish robot used for experiments is shown in Fig. 4.3 (b). It functions on the same components as its static counterpart with the addition of MOSFET trigger switches for the fins. To achieve locomotion, each fin was individually activated by the commands from the controller to generate thrusts that result in overall forward or turning motions. All the electronic components and battery were waterproofed inside the central box while the cables carrying alternating current were connected to the end actuators at the fins.

4.3.2 Localization

The localization framework first performs the matched filtering of the recorded acoustic signal and uses the filter results for the pseudoranging based localization. As a pre-step, we also analyze speaker constellation geometry. The steps are explained in detail in the following sub-sections.



(a) Static receiver module



(b) Receiver module on fish-inspired robot

Figure 4.3: Acoustic Receiver Setup: (a) Receiver module for localization tests: While the MEMS microphone is sealed for water tightness, its bottom port is only covered by a very thin membrane to reduce the dampening of incoming signals. (b) The proof-of-concept bio-inspired fish prepared for experiments is internally identical to the box used for localization evaluation with added MOSFET trigger switches to control the actuators for its fins.

4.3.2.1 Data processing and matched filtering

We perform matched filtering on the recorded chirp sequence from the four speakers to obtain our pseudorange estimates. The matched filter is a linear filter suited for maximizing the signal-to-noise ratio of the recorded signal. The filter conducts a convolution between the recorded sequence, $x[n]$ and a replica of the original chirp played by the speakers $h[n]$:

$$y[n] = \sum_{k=0}^{N-1} h[n-k]x[k] \tag{4.1}$$

The sample number that resembles the replica $h[n]$ most gives a maximal or near maximal amplitude for output $y[n]$. Typically, the maximum output would indicate the sample number to be chosen. However, due to non-line-of-sight reflections in smaller enclosed environments, we

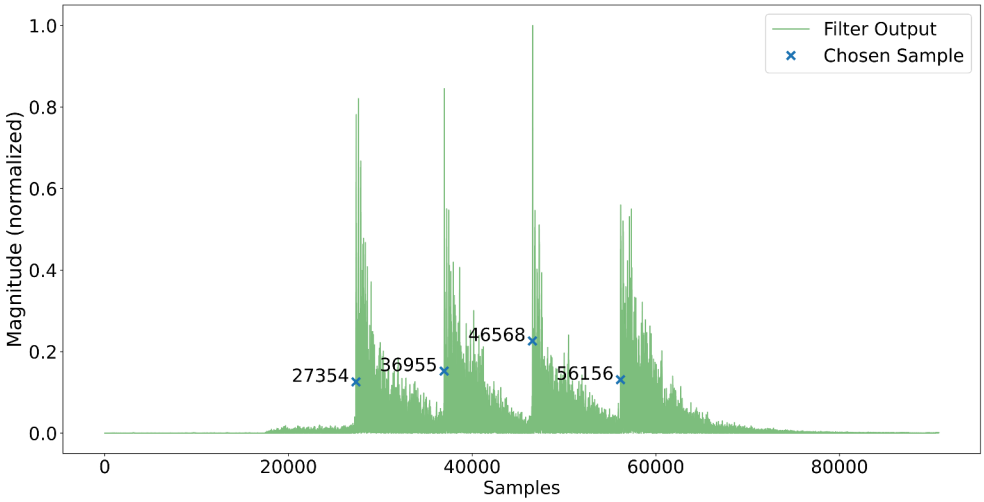


Figure 4.4: Matched Filter Example Output: The normalized filter output is compared against a threshold to pick the first significant peak for each chirp. The sample numbers, after adjusting for the known 9600 sample delay, provide a pseudorange of 843.415, 843.445, 843.847, and 843.477 meters to each speaker as in Eq. 4.2.

have observed that choosing the first major peak gives the best results. The four pseudoranges are obtained by adjusting the known signal delay between each chirp, giving us the sample number for each chirp s_i and multiplying by the speed of sound in fresh water ($1480\frac{m}{s}$) and dividing by the sampling rate (48kHz) gives us the pseudorange observation:

$$P^i = \frac{c}{F_s} s^i = \frac{1480}{48000} s^i \tag{4.2}$$

An example of the matched filter output is shown in Fig. 4.4. The filter output is normalized to keep a constant threshold to select the first significant peak for each chirp. As the expected gap between each chirp is known (9600 samples), the adjusted pseudorange observation for each speaker can be found and used to localize the receiver.

4.3.2.2 Acoustic Pseudoranging

Acoustic pseudoranging is a TDOA technique inspired from GPS trilateration. Our implementation of acoustic pseudoranging differs from its GPS counterpart in a few ways. Unlike in the case of GPS satellites, the speaker positions are constant and known, as well as connected to the same central transmitting module. As such, it does not necessitate the need to account for transmitter side clock biases and errors. We refer to literature from GPS point positioning [23, 4] estimation to formulate our acoustic pseudoranging equations which are described as follows:

Given speaker positions $[x^i, y^i, z^i]$ for speakers $i \in [1, 4]$, receiver position $[x_r, y_r, z_r]$, time step T and receiver clock bias dt_r, each pseudorange for the four or more speakers can be modeled as:

$$P^i(T) = R_r^i(T) + c \times dt_r(T) \tag{4.3}$$

where the distance between each speaker i and receiver r is given by the Euclidean distance

$$R_r^i(T) = \sqrt{(x^i - x_r(T))^2 + (y^i - y_r(T))^2 + (z^i - z_r(T))^2} \tag{4.4}$$

To find the location of the receiver with respect to the speakers, we need to solve for the unknown parameters which are the receiver position and receiver time bias $[x_r, y_r, z_r, dt_r]$ at each time step T. Assuming initial estimates of receiver position and time bias, $[x_0, y_0, z_{0,} dt_0]$, are known, the relationship of the initial estimates of receiver r, and the update parameters can be written as:

$$\begin{aligned} x_r &= x_0 + \Delta x, \quad y_r = y_0 + \Delta y \\ z_r &= z_0 + \Delta z, \quad dt_r = dt_0 + \Delta dt \end{aligned} \tag{4.5}$$

$\Delta x, \Delta y, \Delta z, \Delta dt$ are now the parameters we use a least squares formulation to solve for. Using a first order Taylor expansion, Eq. 4.3 can be expressed as

$$P_r^i(T) = R_0^i(T) + \left.\frac{\partial P}{\partial x}\right|_{x_0} \Delta x + \left.\frac{\partial P}{\partial y}\right|_{y_0} \Delta y + \left.\frac{\partial P}{\partial z}\right|_{z_0} \Delta z + \left.\frac{\partial P}{\partial dt}\right|_{t_0} \Delta dt \tag{4.6}$$

where R_0^i is found from Eq. 4.4 by using the initial position estimate. The partial derivatives from Eq. 4.6 are:

$$\begin{aligned} \left.\frac{\partial P}{\partial x}\right|_{x_0} &= -\frac{x^i - x_0(T)}{R_0^i(T)}, \quad \left.\frac{\partial P}{\partial y}\right|_{y_0} = -\frac{y^i - y_0(T)}{R_0^i(T)} \\ \left.\frac{\partial P}{\partial z}\right|_{z_0} &= -\frac{z^i - z_0(T)}{R_0^i(T)}, \quad \left.\frac{\partial P}{\partial dt}\right|_{t_0} = c \end{aligned} \tag{4.7}$$

Using our partial derivatives from Eq. 4.7, and rearranging the known terms together, we can rewrite Eq. 4.6 as:

$$\begin{aligned} P_r^i(T) - R_0^i(T) = -\frac{x^i - x_0(T)}{R_0^i(T)}\Delta x - \frac{y^i - y_0(T)}{R_0^i(T)}\Delta y \\ - \frac{z^i - z_0(T)}{R_0^i(T)}\Delta z + c \times \Delta dt \end{aligned} \tag{4.8}$$

For ease of reference, let:

$$\begin{aligned} a_{x_r}^i &= -\frac{x^i - x_o(T)}{R_o^i(T)} \\ a_{y_r}^i &= -\frac{y^i - y_o(T)}{R_o^i(T)} \\ a_{z_r}^i &= -\frac{z^i - z_o(T)}{R_o^i(T)} \\ b^i &= P_r^i(T) - R_o^i(T) \end{aligned} \tag{4.9}$$

Assuming we have the minimum of four required speakers, we have the system of linearized simultaneous equations:

$$\begin{aligned} b^1 &= a^1_{x_r}\Delta x + a^1_{y_r}\Delta y + a^1_{z_r}\Delta z + c\Delta dt \\ b^2 &= a^2_{x_r}\Delta x + a^2_{y_r}\Delta y + a^2_{z_r}\Delta z + c\Delta dt \\ b^3 &= a^3_{x_r}\Delta x + a^3_{y_r}\Delta y + a^3_{z_r}\Delta z + c\Delta dt \\ b^4 &= a^4_{x_r}\Delta x + a^4_{y_r}\Delta y + a^4_{z_r}\Delta z + c\Delta dt \end{aligned} \tag{4.10}$$

which can be simplified in their matrix form as:

$$\begin{aligned} A &= \begin{bmatrix} a^i_{x_r} & a^i_{x_r} & a^i_{x_r} & c \\ a^i_{x_r} & a^i_{x_r} & a^i_{x_r} & c \\ a^i_{x_r} & a^i_{x_r} & a^i_{x_r} & c \\ a^i_{x_r} & a^i_{x_r} & a^i_{x_r} & c \end{bmatrix} \\ \boldsymbol{x} &= \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \Delta dt \end{bmatrix}, \boldsymbol{b} = \begin{bmatrix} b^1 \\ b^2 \\ b^3 \\ b^4 \end{bmatrix} \end{aligned} \tag{4.11}$$

where:

$\boldsymbol{b}$ = prediction error

$\boldsymbol{x}$ = vector of unknowns which are the receiver position and clock bias update parameters.

A = design matrix constructed from the linear functions of the unknown variables.

Eq. 4.10 can now be expressed in matrix-vector form as:

$$A\boldsymbol{x} = \boldsymbol{b} \tag{4.12}$$

To better represent the observed measurement errors, which in acoustic pseudoranging arises from selecting the right peak from our matched filter result, we would need to adjust the prediction error and design matrices. This can be done through a form of whitening as explained in [9]. Let $\Sigma = diag(\sigma_1^2, \sigma_2^2, \sigma_3^2, \sigma_4^2)$ where $\sigma_1, .., \sigma_4$ represent the average error observed in pseudorange measurement for each speaker. The design and prediction error matrix are weighted as:

$$A_w = \Sigma^{-1/2}A, B_w = \Sigma^{-1/2}\boldsymbol{b} \tag{4.13}$$

Finally, our least squares problem is solved as:

$$\boldsymbol{x} = (A_w^T A_w)^{-1} A_w^T B_w \tag{4.14}$$

The updated parameters from $\boldsymbol{x}$ are then used to re-initialize the estimated receiver position. The least squares optimization is repeated until the desired error threshold is reached.

4.3.2.3 Dilution of Precision

Akin to GPS, the ability of acoustic pseudoranging to provide an accurate position estimate depends on the geometry composed by the speaker positions. One of the most popular metrics is

the geometric dilution of precision (GDOP). It can be obtained from the design matrix A in Eq. 4.11 by getting the covariance matrix as:

$$\Sigma_{\boldsymbol{x}} = (A^T A)^{-1} \tag{4.15}$$

The GDOP is given as the root of the trace of the covariance matrix, $\Sigma_{\boldsymbol{x}}$ [100]:

$$GDOP = \sqrt{tr(\Sigma_x)} \tag{4.16}$$

A lower GDOP value indicates a better constellation geometry. Based on an analysis by Isik et al. [31], values between 0–5 are considered to give excellent to great estimates, 5–10 give moderate and values 10–20 give a fair estimate. Values above are unlikely to give a reliable solution. As such, the use of acoustic pseudoranging in small spaces requires speaker geometry configurations be analyzed beforehand to ensure good estimation accuracy is obtained.

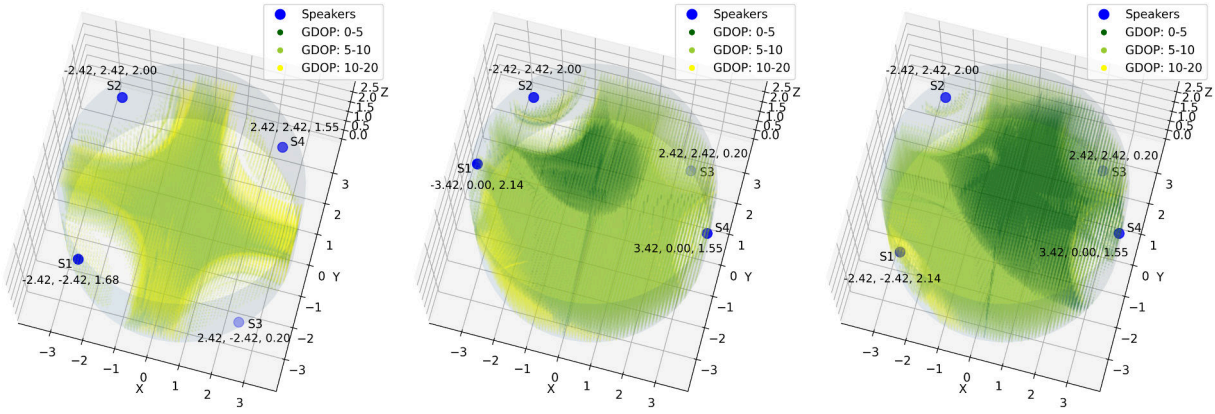
4.3.3 Communication

Apart from acoustic localization, the same hardware and framework can be leveraged to perform one way communication to perform some pre-determined behaviors. As seen earlier in Section 4.2, the controller used by Katzschmann et al. [39, 10] demonstrated the acoustic control of bio-inspired robots while in water. While this method is highly suitable for precise control of a single robot, its ability to simultaneously pass commands to several robots is to be seen. Fischell et al. [15] present a more viable solution for multiple robots to be given acoustic commands through a single speaker. While, the current framework with a single microphone will be unable to perform maneuvers needing bearing estimates to the beacon, it is still possible to give general commands relating to diving or re-surfacing, and directional commands when equipped with other supplementary sensors like IMUs or depth sensors. In our work, as a proof-of-concept, we execute directional commands to move a robotic fish with three actuated fins. We once again rely on matched filtering to differentiate between the type of communication signal received as well as the sequence used for localization. The maximal filter output determines the motion to be executed. The chirps used for communication can have shorter frequency bandwidths since timing resolution is not important to perform simple detection.

4.4 Evaluation

To test the localization framework, experiments were performed in a cylindrical water tank with diameter 7.3m and depth 2.7m. The tests were performed using a watertight box containing the receiver module described in Section 4.3 as seen in Fig 4.3. Ground truth positions were measured by using a graduated rope across the tank with the receiver hanging down from marked positions at pre-measured heights from the bottom of the tank as in Fig. 4.1 (a).

The experiments were repeated in three different speaker geometry configurations for a comparative study. First, the total available volume for each configuration was analyzed to obtain their average GDP values and "solvable volume". We divide our analyzed space into four groups, with GDOP values ranging from 0–5, 5–10, 10–20 and lastly, unsolvable space, where we are

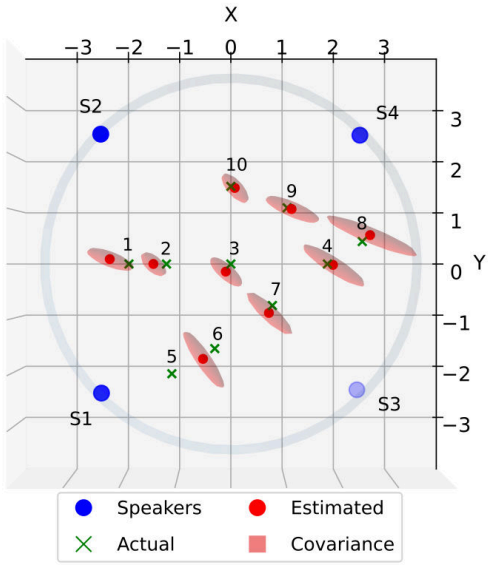


(a) Configuration 1 (b) Configuration 2 (c) Configuration 3

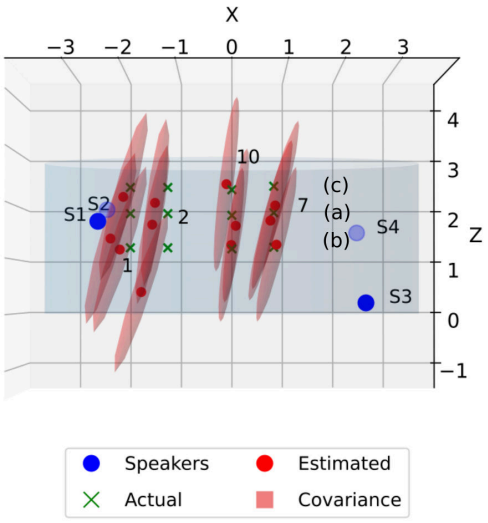
Figure 4.5: Analysis Of Available Volume: The three configurations show the solvable volume ranging from yellow to dark green, depicting a decrease in GDOP value. All points were provided an initial position estimate of (0,0,1). Configuration 1 shows the lowest solvable volume despite a better spread of the speakers in the X-Y plane. However due to the smaller variation in the speaker heights, a good spread in the X-Y plane is not enough to give reliable estimates. The average GDOP value for the solvable space for each configuration was calculated to be 9.39, 6.81 and 5.54 respectively, as evidenced with the increase in the dark green points as seen in (b) and (c).

unlikely to receive a good estimate of the receiver position. Configuration 1 is indicative of a moderate speaker geometry where only 59% of the total available volume of the tank is able to give a GDOP value under 20. Configurations 2 and 3 on the other hand prove to be better constellation geometries with 88% of the total available volume having GDOP value under 20. Details on speaker positions and further information are presented in Fig. 4.5.

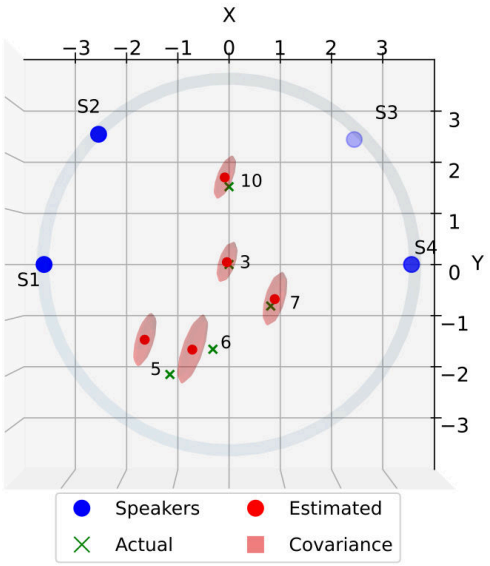
A diverse number of positions were chosen to obtain localization estimates. Mean error across all trials was found to be **0.345m** with a standard deviation of **0.228m**. We break down our results on the basis of the speaker geometry configuration. Configuration 1 had tests conducted for several X-Y coordinates in the tank as well as static X-Y position with different depth placements. Fig. 4.6 shows the estimated positions based on pseudorange observations along with covariance ellipses drawn to the confidence level of 2σ. As seen in Fig. 4.6 (b) the covariance in Z or depth exceeds the covariance in X-Y. This is also seen in Tables 4.1, 4.2, 4.3 and 4.4, where root mean square error (RMSE) in the X-Y plane is considerably lower than the total RMSE. This is explained by the smaller variation (~2m) of the depths in the speaker constellation geometry compared to the variation in the X-Y positions (~7m).



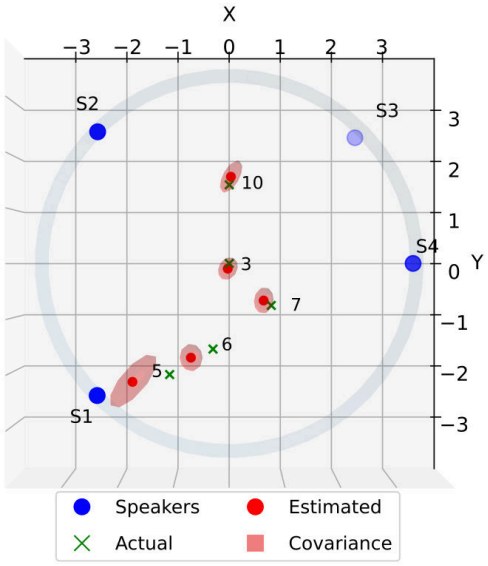
(a) Configuration 1



(b) Configuration 1: Depth Variation



(c) Configuration 2



(d) Configuration 3

Figure 4.6: Localization Results: Localization estimates for the three configurations are presented. All positions were provided with a starting estimate of (0,0,1). (a) Configuration 1 achieves good estimates despite the weaker constellation geometry. However, being in an unsolvable space, like position 5, would give a very poor estimate. (b) The largest variation in speaker depth is between speaker 2 and 3 at ~2m. This causes narrower elevation angles to the receiver making it difficult to achieve lower variance in depth estimates. (c) and (d) compare the change in position accuracy and confidence for the same positions with only speaker 1 shifting position. We see that having more variation is beneficial to achieve better localization accuracy.

Table 4.1: Configuration 1 Results

Pos No.	Ground Truth	Estimate	MSE	XY RMSE
1	-1.90, 0.00, 1.87	-2.27, 0.09, 1.40	0.605	0.387
2	-1.20, 0.00, 1.87	-1.44, 0.00, 2.07	0.313	0.239
3	0.00, 0.00, 1.87	-0.09,-0.14, 1.73	0.225	0.171
4	1.80, 0.00, 1.87	1.91,-0.02, 1.61	0.288	0.114
5	-1.10,-2.05, 1.87	Out of Bounds	NA	NA
6	-0.30,-1.58, 1.87	-0.52,-1.77, 1.86	0.292	0.292
7	0.77,-0.77, 1.87	0.71,-0.92, 1.72	0.217	0.157
8	2.45, 0.42, 1.87	2.60, 0.54, 1.74	0.234	0.195
9	1.05, 1.05, 1.87	1.13, 1.03, 1.92	0.100	0.084
10	0.00, 1.45, 1.87	0.07, 1.43, 1.67	0.214	0.077
Mean			**0.276**	**0.190**

Table 4.2: Configuration 1 Depth Variation Results

Pos No.	Ground Truth	Estimate	RMSE	XY RMSE
1 a	-1.90, 0.00, 1.87	-2.27, 0.09, 1.40	0.605	0.387
1 b	-1.90, 0.00, 1.23	-2.10, 0.04, 1.19	0.207	0.205
1 c	-1.90, 0.00, 2.36	-2.03, 0.03, 2.18	0.224	0.139
2 a	-1.20, 0.00, 1.87	-1.44, 0.00, 2.07	0.313	0.239
2 b	-1.20, 0.00, 1.23	-1.70, 0.04, 0.40	0.967	0.499
2 c	-1.20, 0.00, 2.36	-1.49, 0.11, 1.67	0.761	0.314
7 a	0.77,-0.77, 1.87	0.71,-0.92, 1.72	0.217	0.157
7 b	0.77,-0.77, 1.23	0.82,-0.83, 1.28	0.09	0.077
7 c	0.77,-0.77, 2.36	0.80,-0.79, 2.00	0.358	0.032
10 a	0.00, 1.45, 1.87	0.07, 1.42, 1.67	0.214	0.077
10 b	0.00, 1.45, 1.23	-0.01, 1.60, 1.30	0.170	0.153
10 c	0.00, 1.45, 2.36	-0.10, 1.59, 2.47	0.205	0.172
Mean			**0.360**	**0.204**

The localization results shown in Fig 4.6 (a) and (b) for configuration 1 are tabulated in Table 4.1 and Table 4.2. To be noted, position 5 is in an area which is considered to be unsolvable as per our GDOP analysis from Fig. 4.5 (a). As predicted by the analysis, experimental results for position 5 provide an out of bounds estimate. Comparatively, configurations 2 and 3 have a larger solvable volume, and are able to give reasonable localization estimate for position 5, although with relatively higher error than other points. Tables 4.3 and 4.4 for configurations 2 and 3 show results for tests conducted at the same positions which can be visualized in Fig. 4.6 (c) and (d). Comparing Fig. 4.6 (c) and (d) shows that larger variance in speaker positions is likely to give smaller localization error as evidenced by the smaller covariance ellipses seen for configuration 3. We ignore errors through multi-path reflections and speaker washout (being close enough to any speaker may produce artifacts in recorded signal sequences). We still find the localization

results to be satisfactory considering the low cost of implementing such a system.

Acoustic communication is successfully performed to cycle through three different motions using matched filtering to differentiate between the acoustic signals. The motions we implemented include moving the tail fin alone to move forward, and move left or right by simultaneously moving the tail and appropriate pectoral fin.

Table 4.3: Configuration 2 Results

Pos No.	Ground Truth	Estimate	RMSE	XY RMSE
3	0.00, 0.00, 1.87	-0.04, 0.04, 1.98	0.128	0.059
5	-1.10,-2.05, 1.87	-1.56,-1.39, 2.18	0.745	0.677
6	-0.30,-1.58, 1.87	-0.69,-1.60, 1.46	0.569	0.391
7	0.77,-0.77, 1.87	0.84,-0.64, 2.14	0.312	0.149
10	0.00, 1.45, 1.87	-0.08, 1.62, 1.79	0.211	0.194
Mean			**0.393**	**0.294**

Table 4.4: Configuration 3 Results

Pos No.	Ground Truth	Estimate	RMSE	XY RMSE
3	0.00, 0.00, 1.87	-0.03,-0.10, 1.71	0.186	0.101
5	-1.10,-2.05, 1.87	-1.81,-2.22, 1.26	0.754	0.448
6	-0.30,-1.58, 1.87	-0.71,-1.75, 1.65	0.495	0.445
7	0.77,-0.77, 1.87	0.63,-0.69, 1.81	0.168	0.159
10	0.00, 1.45, 1.87	0.03, 1.59, 1.86	0.153	0.153
Mean			**0.351**	**0.261**

4.5 Conclusions

This chapter presented an acoustic localization and communication method suitable for small underwater robots such as bio-inspired robotic fish. Our approach uses a cheap MEMS microphone on the robot in combination with a constellation of four speakers in a known geometry that makes it possible for multiple robots to localize themselves simultaneously, which is useful for geo-referencing collected data. We present an implementation using a cost and power efficient Raspberry Pi Zero W on a bio-inspired robotic fish.

Enclosed spaces, like the water tank are prone to artifacts like multi-path reflections, speaker washout, and difficulty in finding ideal speaker constellation geometry. As such, acoustic pseudoranging is likely to perform better in larger, open waters. We show that despite being deployed in a small area, it is still effective in providing a reasonable position estimate. Future directions for the work on acoustic localization would involve the implementation of filtering and smoothing for trajectory estimation and optimization. Information from auxiliary sensors like IMUs and MEMS pressure sensors when fused with pseudoranging-based position estimates provide a

more accurate state estimate. We aim to prove the efficacy of the improved method in open water compared to traditional inertial navigational methods used for underwater robots.

The paper also demonstrates the use of the same architecture for controlling a robot through acoustic signals. While we limit our experiments to basic signal processing and primitive motions, more robust communication protocols like frequency shift keying can be used to perform a vast array of motions and tasks. Future iterations would aim to have a more sophisticated communication framework to achieve closed loop navigation and control, pushing bio-inspired underwater robots further towards autonomy.

Chapter 5

SONIC: Sonar Image Correspondence using Pose Supervision for Imaging Sonars

5.1 Problem Statement

Feature-based methods excel at localization and mapping by tracking distinct features over time. With camera frameworks, these methods utilize photometric consistency and invariance to common transformations such as scale and rotation, to produce precise feature correspondences [56]. Camera-based localization and mapping face hurdles underwater due to reduced color depth and visibility, limiting usable data. In contrast, *forward looking* or *imaging* sonars excel underwater, offering long-range visibility impervious to water particulates. While sonars are optimal for such scenarios, they currently lack robust feature descriptors. In the camera imaging domain, feature descriptors and detection are well-researched, with notable examples being SIFT [59], ORB [72] and AKAZE [1].

The robustness of these feature descriptors stems mainly from two principles, photometric consistency and geometric invariance. Photometric consistency maintains pixel value stability against viewing angle variations and noise, while geometric resilience ensures feature recognition across varying orientations and scales. On the other hand, imaging sonars exhibit variations in intensity returns and observable shapes when viewing the same object from different angles, influenced by the object's material and geometry. These aforementioned descriptors struggle with the prevalent speckle noise and intensity variations characteristic of sonar images, especially in the polar space. An example of such a failure shown in Fig. 5.1, in row 3 with AKAZE and the brute force (BF) matcher. The downstream effects of this failure can result in poor, or error-prone loop closure detection and state estimation.

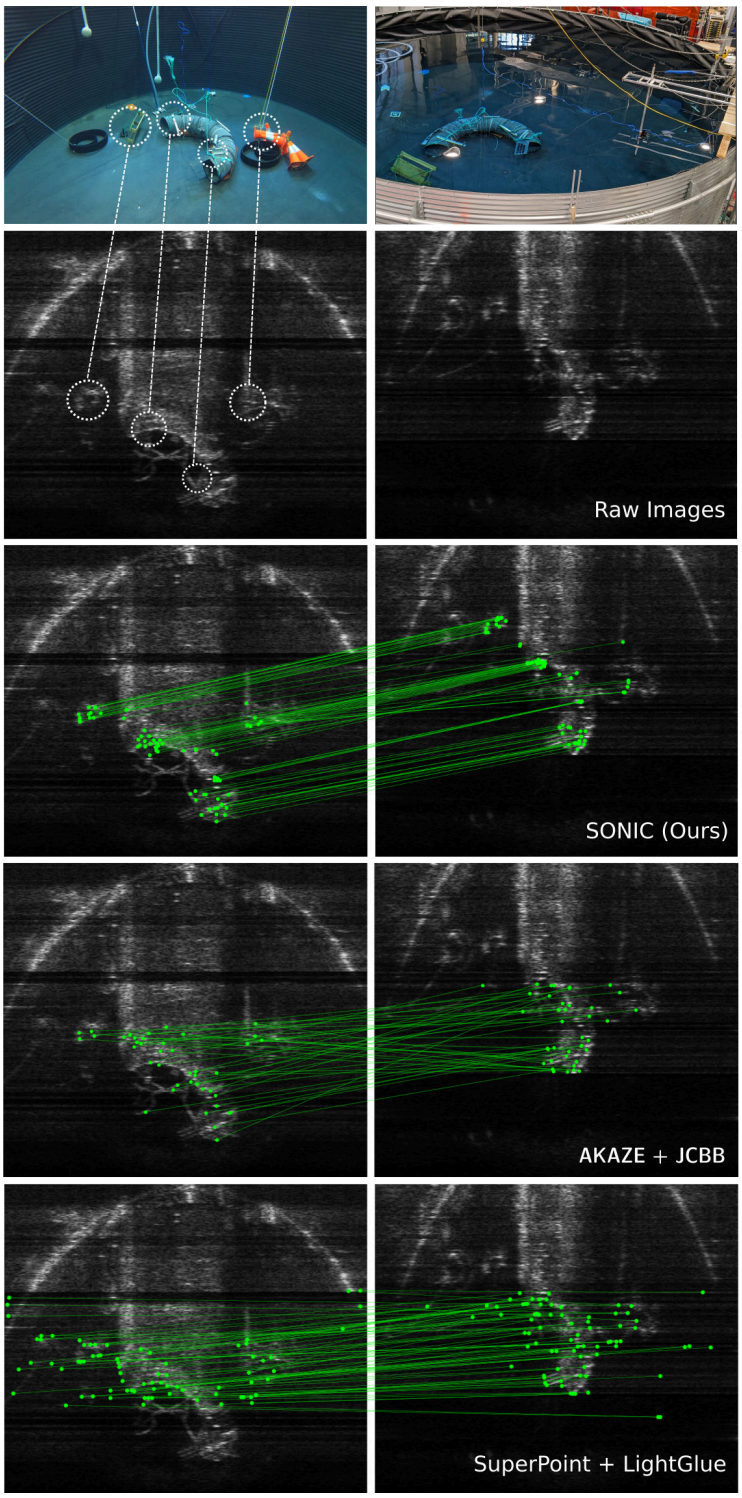


Figure 5.1: Real-world matching performance: Sonar images taken from different planar positions in a test tank show our method providing significantly better matches than AKAZE with the brute force matcher, and the SuperPoint keypoints matched using LightGlue. Given the keypoints in the first frame, SONIC uses expectation matching to determine the correspondences and presents only those correspondences with high confidence.

Our work addresses this issue by using pose supervision grounded in sonar geometry. This offers a feature correspondence method for sonar images, robust against photometric and geometric variations. Recent research has improved upon the matching performance of camera images with deep neural network based approaches using ground truth correspondences [75, 11, 81][75, 11, 81]. There has also been recent semi-supervised work techniques leveraging pose-supervised learning [88]. These models, though powerful, are trained on camera data and suffer from the same problems explained above when used on sonar data.

Motivated by [88] and the availability of pose-supplemented sonar data using simulation [67], we propose a novel pose-supervised method to learn sonar image matching using a sonar epipolar-contour (see 5.3.1) based loss function. In addition, we also enforce cyclic consistency as done in [88]. Our network learns directly from sonar images in the polar space and the loss is calculated in this space ensuring the model learns unique representations in sonar images. Specifically, our main contributions are:

1. A novel semi-supervised method for sonar image correspondence, utilizing sonar-specific epipolar geometry negating the need for ground truth correspondences. This technique offers correspondences tailored for imaging sonar, producing features resilient to viewpoint changes.
2. An extensive simulated dataset that comprises over 300K pairs of sonar images and their corresponding ground truth poses, collected across 10 distinct scenes. Each scene is made from randomized positioning of a specific set of objects, with repeatable sensor motion from different pose offsets.

5.2 Related Work

The descriptors mentioned in Section 5.1 have been utilized for sonar images in the past for applications towards acoustic structure from motion (ASFM) and feature-based SLAM [29, 91, 76, 44, 92]. These descriptors worked as long as the change in viewing angle was minimal. A couple of recent works [61, 85] give a summary of different keypoint detectors and feature extractors on sonar images and they along with [91] suggest the need for an invariant sonar specific feature descriptor.

Hand-designing a new feature descriptor for a specific type of sensor is a viable approach [20, 99], and there has been recent development on a SIFT-like descriptor made for multi-beam sonar[103]. The drawback to this process is that the descriptor parameters would need to be manually tuned for different imaging sonar models, as each sonar make has a unique elevation, bearing and range specification, and signal-to-noise profile. On the other hand, recent research on learned feature descriptors for cameras [11, 81, 17] and 3D lidars [12] have shown encouraging results when used for correspondence estimation. Similarly, methods like [69] leveraged deep neural networks for place recognition for sonar images but have not learned feature correspondences. These methods utilized deep networks to solve the problem of large variations across scenes, and multiple sensor makes. Looking specifically towards research for camera images, there have been several supervised methods such as SuperPoint [11] and LOFTR [81]. SuperPoint's network works on producing the learned feature descriptors using homographic

adaptations for supervision. Follow-up work in SuperGlue, and more recently, LightGlue [75, 49] utilize the SuperPoint feature descriptors to form a robust feature matching framework. Approaches similar to SuperPoint rely heavily on abundant training data, for example from the MegaDepth dataset[46]. Other approaches, like the current state of the art, SiLK[17], depend on sub-pixel ground truth correspondences as a linear mapping during training.

These techniques, while likely to yield similar great results for sonar, are a challenging endeavor to replicate for imaging sonar. This is primarily owing to the scarcity of readily available open-access sonar data and a dearth of ground truth feature point and correspondence information. Additionally, for sonar images, the motion model for planar scenes do not reduce to a homography [57]. To address these issues, we look towards ongoing research dedicated to semi-supervised approaches that harness sensor pose data. This enables us to circumvent the necessity for precise ground truth correspondences among feature points. These methods have demonstrated success in achieving equivalent, if not better accuracy compared to their fully supervised counterparts, all while demanding a smaller volume of training data. A noteworthy example of a pose supervised method is CAPS[88]. With access to pose information relating to two images capturing the same scene, CAPS effectively employs a pair of loss functions: epipolar loss and cyclic loss. The foundation of their system rests upon the fundamental notion that a point of interest in the initial image should invariably align with the epipolar line corresponding to its counterpart in the second image. Our approach builds on the backbone of CAPS, utilizing an analog to epipolar geometry for imaging sonars which is detailed in Section~\ref{sec:System}. This gives a data and time-efficient way of obtaining trained feature descriptors for sonar images which can outperform the currently popular methods in use.

5.3 Pose Supervision for Imaging Sonar

In this section we first introduce the concept of sonar epipolar geometry, and then discuss the approach to be taken for training descriptors for imaging sonar.

5.3.1 Sonar Epipolar Geometry

Negahdaripour introduced the concept of the epipolar contour [58]. In cameras, the *epipolar line* is defined as the intersection of the epipolar plane of a point in space with the image plane. As a result, it can also be thought of as the projection of the line joining the camera center and the point in 3D on to the image plane of the other camera . This line represents the depth and direction from the first view. In sonar stereo, elevation arcs serve as the analog to the epipolar lines found in cameras, and their projection is an *epipolar contour*. We will describe the epipolar geometry in brief here, but refer the reader to [58] for a detailed explanation.

We refer to Section 2.3where we again use Equation 2.16 for notation relating to the conversion of a point in polar space to the Cartesian space. However, since we are focused on training sonar images, we prefer to use pixels in the range bearing space instead. Given a point in one sonar image at some range, R and bearing θ. We know that the point in 3D will lie along the elevation arc at the same range and bearing as defined and at some arbitrary elevation angle, φ. Since we do not know the elevation angle, the ambiguity is along the elevation arc for

$\varphi <= \varphi_{max}$ and $\varphi >= \varphi_{min}$. Now, the conversion of points in 3D Cartesian space to the range bearing space is as seen in Equation 5.1.

$$\boldsymbol{P} = \begin{bmatrix} R \\ \theta \\ \phi \end{bmatrix} = r \begin{bmatrix} \sqrt{x^2 + y^2 + z^2} \\ arctan2(y, x) \\ arctan2(x, \sqrt{x^2 + y^2}) \end{bmatrix} \tag{5.1}$$

Assuming we have a feature point $\boldsymbol{p}$ in the first image, and $\boldsymbol{R_{1,2}}$and $\boldsymbol{t_{1,2}}$are the known rotation and translation between the coordinate frame of image 1 and image 2. The locus of the same feature point in image 2, $\boldsymbol{p}^{'}$ is calculated as seen in Equation 5.3. This 3D feature locus can now be projected into the polar sonar image plane as in Equation 5.4. Fig. 5.2 visually describes the process of projecting the elevation arc from the first frame onto the second image plane. The red line is the *epipolar contour* we use for the loss function described in a later section.

$$\boldsymbol{R_{1,2}} = \begin{bmatrix} r_1 \\ r_2 \\ r_3 \end{bmatrix}, \boldsymbol{t_{1,2}} = \begin{bmatrix} t_x & t_y & t_z \end{bmatrix}^T \tag{5.2}$$

$$\boldsymbol{p}'(\phi) = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} r_1 \cdot P + t_x \\ r_2 \cdot P + t_y \\ r_3 \cdot P + t_z \end{bmatrix} \tag{5.3}$$

$$\boldsymbol{p}^{'}_{\boldsymbol{proj}}(\phi) = \begin{bmatrix} r' \\ \theta' \end{bmatrix} = \begin{bmatrix} \left\| P'_p \right\|_2 \\ arctan2(y', x') \end{bmatrix} \tag{5.4}$$

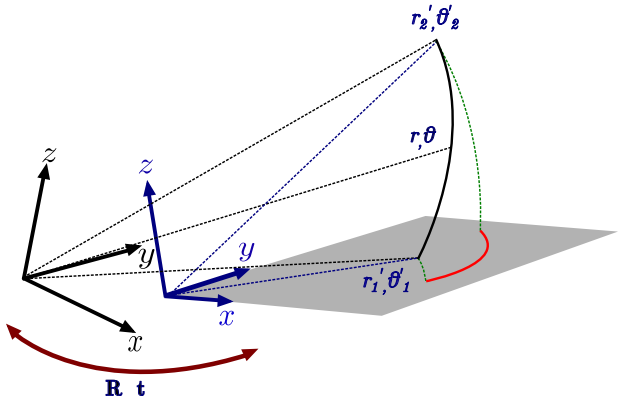


Figure 5.2: Sonar Epipolar Geometry: The elevation arc of a point in the first image is transformed into the frame of the second image and then projected, which creates an epipolar contour.

5.3.2 Keypoint Detection

SONIC, like CAPS, trains feature representations and requires keypoints as an input. We use a combination of AKAZE and SuperPoint keypoints for training to ensure we have enough keypoints per frame while training. Keypoints are found in polar space as converting sonar images to euclidean spaces leads to loss of information, especially when near the sonar.

5.3.3 Network highlights

We use a CNN based encoder-decoder architecture with a differentiable matching layer and coarse-to-fine technique similar to [88]. Unlike CAPS, SONIC uses a ResNet-34 [21] base to prevent overfitting our small dataset. From here, similar to CAPS, the encoder creates a coarse representation; given the input keypoint we find the corresponding point in this representation using the differentiable matching layer. A pictorial representation of the differentiable matching layer and coarse to fine architecture is shown in Fig. 5.3 (a) and (b) respectively. Given two images, the network with shared weights creates the representation M_1 and M_2. For each query point x_1, the feature descriptor $M_1(x_1)$ is correlated to each point in M_2. This is used to find the probability of each point being the correspondence of x_1 in M_1 as shown in Eq. 5.5. A correspondence is calculated by finding the expectation of this probability in Eq. 5.6.

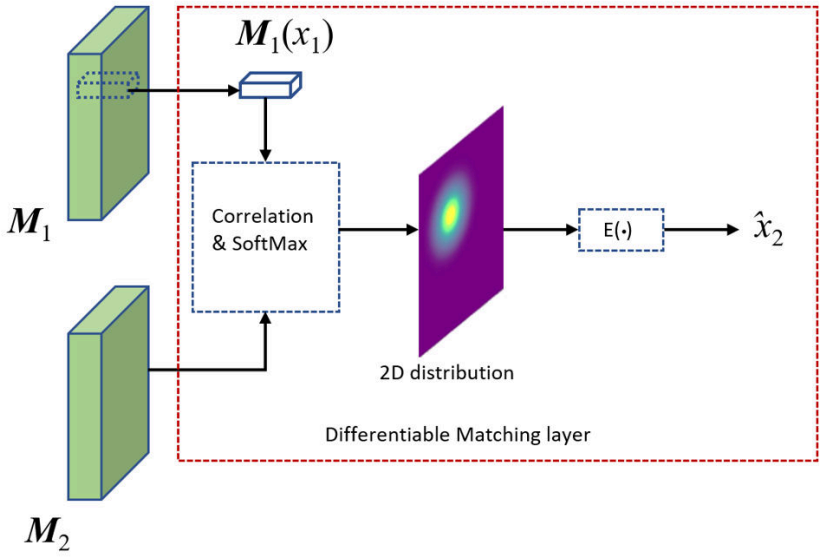
Searching correspondences for all points over the image is very computationally costly, hence it makes most sense to sparsely sample the query points for supervision. This coarse to fine architecture improves on efficiency. At the coarse level, a correspondence distribution is computed over all the locations. However, at the finer level, the distribution is only computed at the highest probability location observed from the coarse map. The loss functions are imposed on both levels. Our other modifications to the original CAPS architecture include changes to work with single channel sonar images, and condense final coarse and fine layer outputs to 64 dimensions instead of the original 128.

$$p(x|x_1, M_1, M_2) = \frac{\exp(M_1(x_1)^T M_2(x))}{\sum_{y \in I_2} \exp(M_1(x_1)^T M_2(y))} \tag{5.5}$$

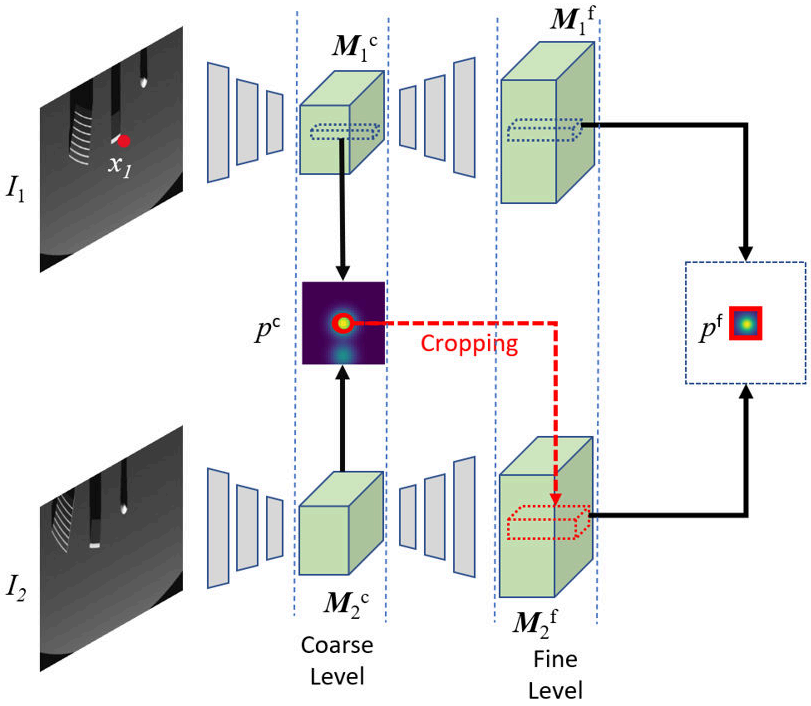
$$\hat{x}_2 = h_{1\to 2}(x1) = \sum_{x \in I_2} x \cdot p(x|x_1, M_1, M_2) \tag{5.6}$$

5.3.4 Supervision

Like CAPS, we propose two loss functions, modifying it for sonar images. We introduce the sonar-epipolar and sonar-cyclic loss. Given the relative pose between two image frames I_1 and I_2, we use use Equations 5.3 and 5.4 to determine the epipolar contour. The predicted point, if predicted correctly, should lie on this epipolar contour. Thus, we use this as a distance metric that we can optimize over. The epipolar loss term, $L_{epipolar}$ in Equation 5.7 is defined as the shortest distance between the predicted correspondence of a feature point $x_1$in I_1 and the epipolar contour of x_1 in the second image I_2. In our implementation, we sample points along the elevation arc of the first point in the first image and then transform and project them on to the second image to create a discrete epipolar contour of the sampled points. In the polar frame, the loss is thus the minimum of the distance between the predicted point and each point on the arc as seen in Equation 5.11. The epipolar loss only checks for the predicted match to lie on the estimated contour, and does not necessarily look in the vicinity of the ground truth correspondence location. To further constrain the system, a cyclic consistency loss is utilized which aims to keep the forward-backward mapping of the point to be close to itself. The weighted sum of the losses $L_{epipolar}$ and L_{cyclic} is our final loss function as seen in Equation5.11. A point to note is that the



(a) Differentiable Matching Layer



(b) Coarse to Fine module

Figure 5.3: Network architecture highlights: a) For each query point x_1, its corresponding location $\hat{x}_2$(Eq. 5.6) is represented as the expectation of a distribution computed from the correlation between the feature descriptors. The associated uncertainty also helps in reweighting training loss. During training, keypoints serve as queries. (b) Searching correspondence across the entire image is costly. The location of the correspondence p^c at the coarse level is used to ascertain a local window at the fine level, p^f is found in this window using differentiable matching.

distances found for both the losses are in the range-bearing space. Due to nature of sonar images, it is important for the network to learn this distinction, and the combined loss terms in the range bearing space help with this. A graphical representation of the losses is seen in Fig. 5.4.

$$L_{epipolar}(x_1) = dist(h_{1\to2}(x_1), ep\_contour) \tag{5.7}$$

$$L_{cyclic}(x_1) = \|h_{2\to1}(h_{1\to2}(x_1) - x_1\|_2 \tag{5.8}$$

$$L_{(I_1,I_2)} = \sum_{i=1}^{n}[L_{epipolar}(x_1^i) + \lambda L_{cyclic}(x_1^i)] \tag{5.9}$$

$$L_{cyclic}((r_1,\theta_1),(r_2,\theta_2)) = r_1^2 + r_2^2 - 2r_1r_2(cos(\theta_1 - \theta_2)) \tag{5.10}$$

$$L_{epipolar}(p\prime_{proj}(\varphi), r_2, \theta_2) = \min_{\phi}(r(\varphi)^2 + r_2^2 - 2r(\varphi)r_2(cos(\theta(\varphi) - \theta_2))) \tag{5.11}$$

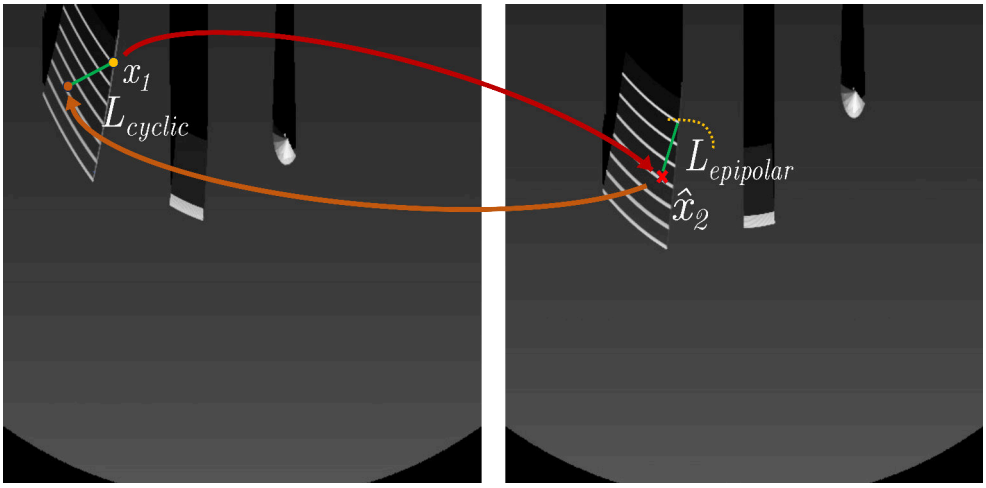


Figure 5.4: Loss functions: The yellow point x_1 represents a queried keypoint in the first image. The red cross $\hat{x_2}$ is the predicted point. The yellow dotted line represents the sampled points on the epipolar contour of point x_1. $L_{epipolar}$ is the shortest distance to the epipolar contour, or simply the epipolar loss. L_{cyclic} is the cyclic loss to assert that the mapping of the feature point is close to its original position.

5.3.5 Data and Training

Acquiring numerous sonar images with ground truth pose is challenging, and real-world data often lacks feature-rich frames for network training. While there are sonar datasets like those by Singh et al. [78] and Malios et al. [53] for object classification or SLAM, they either lack pose information or do not use wide elevation imaging sonars. Thus, simulated environments, particularly the HoloOcean simulator [66, 67], supplemented by custom worlds, offer an enhanced solution for data generation.

For our primary application, seafloor mapping, sonar images were gathered from 1-4m above the floor at a downward pitch of 10-20^0. Multiple trajectories with varying rotation and translation offsets produced image pairs. Our dataset comprises approximately 300K training and 30K validation pairs. We configure our simulated images on the Blueprint Subsea M1200d's [5] low frequency mode. In this mode, the images have a maximum azimuthal field of view of 130^0 and elevation of 20^0. The maximum range is set to be 10m. The image is comprised of 512x512 range and bearing bins. Image noise parameters are marginally varied around the values provided in HoloOcean's example scenarios.

Epipolar and cyclic losses are given 0.7 and 0.3 weights, respectively. We train the network for $\approx$ 7 epochs with a batch size of 14 pairs, using a Nvidia GeForce 4090 RTX with 24GB memory.

5.4 Evaluation

We assessed matching performance by comparing SONIC, AKAZE, and the camera-trained LightGlue model. As highlighted in Section 5.1, AKAZE is the preferred keypoint and descriptor for imaging sonar applications. LightGlue [49] is a recent improvement on SuperGlue, with a performance equivalent to LoFTR [81] and MatchFormer [89] which are considered the current state of the art for camera image correspondence. We compare SONIC with LightGlue since both use SuperPoint keypoints and require a keypoint detector, unlike other detector-free matchers.

Assuming a planar scene we evaluate the number of matched keypoints that are considered to be inliers for each of AKAZE, LightGlue and SONIC. Inliers are classified by projecting the keypoints from the reference image using the ground truth relative pose information as described in Section 5.3.1 onto the query image. The threshold selected for simulation images is 12 pixels, which corresponds to 0.23m in range or 3° in bearing. We increase this threshold for real images to be 20 pixels, which corresponds to 0.2m and 5° in range and bearing respectively. To show the benefit of our method towards downstream tasks such as feature-based SLAM, we evaluate performance in relative sensor pose recovery using the two-view acoustic bundle adjustment framework to recover sensor pose information as presented by Westman et al. [92].

It is typical for hovering underwater vehicles to be equipped with pressure sensors for depth estimation, and to be limited to planar motion [36]. Most graph-based solutions for underwater SLAM would thus model *Z*, *pitch* and *roll* as a separate unary factor, independent to planar motions in *X*, *Y*, and *Yaw* [83]. Thus we focus mainly on evaluating the average absolute error in planar translation (*xy*) and rotation (*yaw*).

Prior pose estimates are derived from corrupted ground truth data. These poses are used to filter matches. We perform a Z-test on the distance between the corresponding projected and matched query keypoints. We prune matches not falling under a 2σ threshold for all three methods. The same noisy pose prior is provided to the acoustic bundle adjustment optimizer along with matched keypoints from the reference and query images, passed as their bearing and range values. In SLAM application, the data association could be improved by using RANSAC [77] or JCBB [92].

We present and discuss results for our simulated datasets first, and then for the real world

Method	Simulation - Small Variation	Simulation - Large Variation	Test Tank
AKAZE	24.23%	10.22%	40.08%
LightGlue	39.13%	11.18%	51.92%
SONIC	**49.43%**	**23.56%**	**74.53%**

Table 5.1: Percentage of Inliers

Method	Simulated Data			
	Small Variation		Large Variation	
	Translation (m)	Rotation (rad)	Translation (m)	Rotation (rad)
AKAZE	μ: 4.24, σ: 2.40	μ: 1.56, σ: 1.22	μ: 5.06, σ: 2.35	μ: 1.44, σ: 1.01
LightGlue	μ: 3.62, σ: 4.52	μ:0.97, σ: 1.12	μ: 3.49, σ: 3.53	μ: 0.97, σ: 1.00
SONIC	**μ: 0.88, σ: 2.01**	**μ: 0.25, σ: 0.61**	**μ: 2.23, σ: 3.35**	**μ: 0.60, σ: 0.82**

Table 5.2: Planar Translation and Rotational Accuracy

data from a test tank.

5.4.1 Simulation

We sample sonar image pairs, unused for SONIC's training, with varied pose differences including minor roll, pitch, and z variations, and broader x, y, and yaw differences. These pairs are categorized into two groups: small and large variation. The small variation group contains pairs with *yaw* $< \pm 5^\circ$ and *x,y* $< \pm 1.5$m. The large variation group contains pairs with up to $\pm 40^\circ$ variation in rotation and ± 7m in translation. Evaluation images had the same parameters for range, bearing and elevation as used in training.

We first look at the inlier ratio in Table 5.1, where both small and large variation groups show significantly more ground truth pose agreeable matches than the other two methods. We then look at Table 5.2 to see the average absolute error and standard deviation for planar translation(meters) and rotation(radians). Our method outperforms the other two in all situations. SONIC is specifically trained to identify features which are more likely to be invariant to viewpoint changes, as well as learn the geometric variance of features across these changes. The inductive bias of convolutional neural nets also ensures that it looks at a area and structure around the keypoint and finds matches closest to that in the other image. AKAZE and other traditional descriptors try to describe features such as as corners, edges and blobs which do not stay consistent in sonar images. LightGlue, and other learned matchers also look at the image as a whole and thus perform much better than AKAZE. However, since LightGlue is trained on camera images it cannot predict accurate matches when the structures or keypoints change over significant translation or rotational variation due to the change in object geometries occurring in the polar image-space as seen in Fig. 5.5.

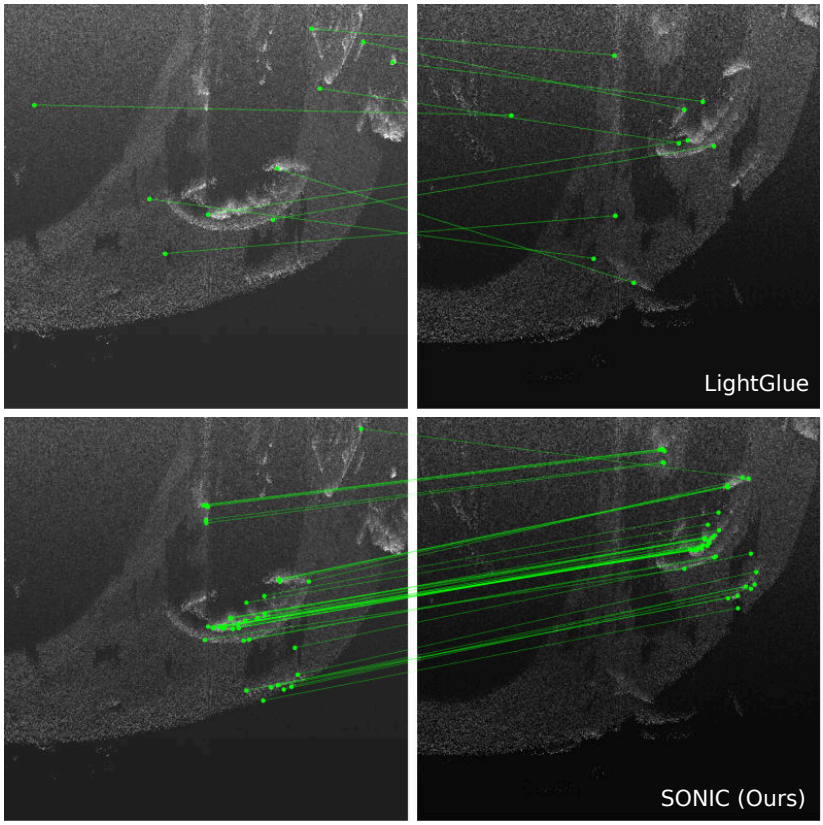


Figure 5.5: Simulation Matching Performance: LightGlue is unable to match features when the structures' shape warps under sensor motion, a common phenomenon in polar images.

5.4.2 Test Tank Evaluation

Transferring simulated performance to real-world scenarios is challenging. As SONIC was solely trained on simulated data, it's crucial to assess if it learned the geometry of view-invariant features in actual sonar data. Accurately modeling sonar noise is challenging, making real-image performance evaluation vital. We conducted experiments in a test tank with a sliding gantry system with a configurable 6 degrees of freedom sensor pose as seen in Fig. 5.1. We used a Leica Total Station 16 [43] with a tracking prism for ground truth estimation. Here, we show two sample frames, taken at differing positions with a change of 1.153m and 1.310m in $\mathbf{x}$ and $\mathbf{y}$ and a yaw rotation of $-15°$. The other degrees of freedom remained constant. We observe our method is able to provide the right matches, whereas previously used methods like AKAZE with symmetric nearest-neighbour brute force matching and SuperPoint descriptors with LightGlue are unable to do so. The matching results of the three methods in the tank can also been seen in the same figure. While we trained our images on a maximum range setting of 10 meter, the real world parameters was fixed to 6m due to the small size of the tank (7m diameter). We also had to limit the keypoint detectors for all three methods from detecting points beyond the general position of the objects due to significant acoustic reflections between the metallic surfaces of the tank wall and pipe placed inside the tank. Due to the limited change in pose, and concentrated features, the two-view acoustic bundle adjustment framework was unable to resolve the relative

pose for most image pairs. However, we can see from the inlier ratio in Table 5.1 that SONIC is once again more likely to provide better correspondences.

5.5 Conclusions

We propose SONIC, a pose-supervised model that solves the challenging problem of feature correspondence in sonar images. Our results demonstrate that SONIC excels over existing techniques in sonar image feature extraction, addressing the need for sonar-centric feature correspondence. Our method achieves this through a novel sonar epipolar loss and a cyclic consistency loss. The epipolar loss utlizes the epipolar contour projected by an arc onto the second image using relative pose information. This guides the predicted correspondence to align with this contour. Simultaneously, our consistency loss ensures cyclic congruence between the predicted and the initial keypoints. Our method, trained solely on simulated data, demonstrates promising real-world feature correspondence due to its understanding of underlying geometry. However, more thorough open water tests are needed, along with supplementing real data towards training to close the gap between simulation and real-world results. Future applications of this work include the extension of the method to incorporate different sonar frequency modes and parameters under one model. Recent work on image matching using attention [96] opens the door to enable cross-sonar feature matching and improving performance, which could potentially help cross-platform localization and mapping.

Chapter 6

C-SONIC: Cross-Sonar Image Correspondence for Generalized Feature Matching in Imaging Sonars

6.1 Problem Statement

Imaging sonars are often used in deep water applications where visibility and turbidity make the use of cameras suboptimal. They offer long-range visibility that is largely unaffected by particulates. However, there are some notable downsides to their use, most notably the variability in image formation. This variability arises due to user-defined parameters like frequency mode, maximum and minimum range, and manufacturer design specifications like range and azimuthal binning. This compounds the issue of geometric invariance and the lack of photometric consistency in sonar images. Thus, when it comes to fleet applications, it is desirable to have the same sonar on all robots for collaborative mapping, which can lead to a significant expense if higher-end sonars are required. In our previous work, SONIC [18], we demonstrated the use of pose supervision to train a network to learn feature correspondences for imaging sonars in a fixed parameter setting. SONIC provided consistent and coherent feature correspondences between images taken from significant viewpoint variations. In this chapter, we present a natural progression of SONIC called Cross-SONIC or C-SONIC in short. As the name suggests, C-SONIC enables cross-sonar image feature matching. C-SONIC provides a general purpose model capable of finding reliable correspondence between different frequency and range parameters. This opens the possibility of localizing a cheaper AUV with a more cost-effective sonar through a map generated by a more expensive AUV with a higher frequency sonar. The contributions of this work can be summarized as follows:

1. A new cross-sonar feature matching model capable of working for most of the Blueprint Subsea Oculus family of imaging sonars. This also includes a framework to train a new model using custom data from other imaging sonars.
2. A 550K image pair dataset containing imaging sonar data with differing frequency and range parameters for the Blueprint Subsea Oculus family of imaging sonars, as well as a small subset of data modeled around the SoundMetrics DIDSON imaging sonar. All

the image pairs, as with SONIC, contain pose information. In addition, the dataset also contains metadata information for each frame.

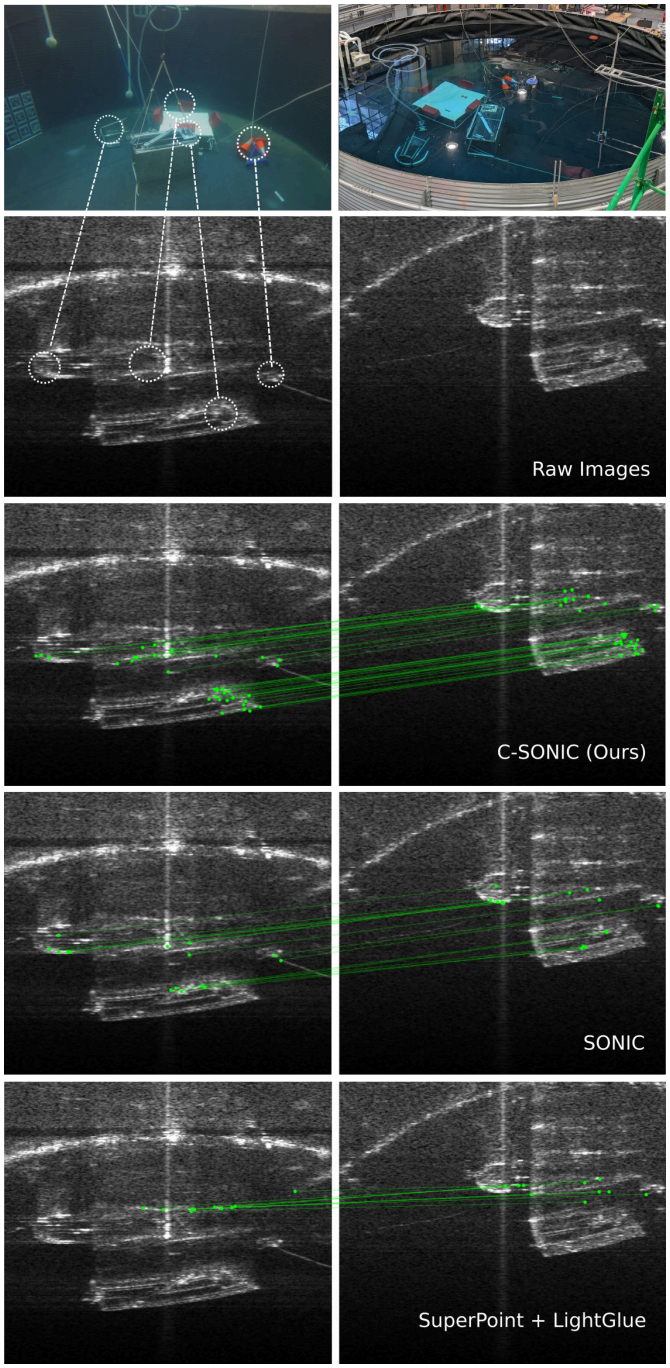


Figure 6.1: Real-world matching performance: Sonar images taken from different planar positions in a test tank, with the left being in the low frequency mode, and the right in the high frequency mode. We see our new method, C-SONIC, providing significantly better matches than LightGlue, and the retrained SONIC network. The cross-attention layer improves matching performance significantly, even with different frequency mode image pairs.

6.2 Related Work

A significant challenge posed to determining correspondences between two images when considering the same scene or object is the variability in viewpoint, texture or illumination. In sonar imagery, this problem is further pronounced due to the underlying acoustics-based mechanics of image formation. As acoustic reflections are highly dependent on incidence angles and object material textures, images captured from different viewpoints can vary drastically. The previous chapter highlighted the use of pose-supervision as a potential solution to this problem. However, when it comes to camera images, different makes and models of cameras still produce generally similar looking images. On the other hand, sonar images produced by different sonars can look very different. Thus, this in effect becomes a semi-multi-modal perception problem.

Attention mechanisms in deep neural networks have evolved from their early applications in sequence-to-sequence models [2, 51] to becoming the core component in the groundbreaking "Attention is All You Need" paper [86], which introduced the Transformer architecture and revolutionized natural language processing. Since then, attention mechanisms have made their way into visual networks in both CNN architectures [34, 97], as well as the vision transformer networks [13]. While self-attention allowed classification tasks to perform better, it is cross-attention (also referred to as co-attention) which made the performance distinction for feature and object correspondence and detection in single [81, 89, 96] and multi-modal perception models [45]. While transformers can provide greater flexibility and improved performance for computer vision tasks, it often comes at the expense of computational costs and larger datasets [13]. Similarly, inference times also increase multifold while using transformer architectures. Hence we look towards using cross-attention modules in CNN architectures like the work done by Wiles et al. [96] in their CAPS variant, which forms the backbone of SONIC. They introduce a modular cross-attention layer which can be applied at different stages of the architecture. In this work, we focus on a single such module, embedded into the final encoder layers instead.

6.3 Method

In this section we describe the key improvements to the architecture of SONIC which lead to better correspondence matching performance in cross-sonar images with variable parameter settings through C-SONIC.

6.3.1 Cross-Attention Module

The aim of the cross-attention module is to determine the location of a set of features in the first image which should attend to a set of features in another image. Referencing from Wiles et al. [96], consider a set of features g and h in images $I_1$and I_2 respectively. For each feature i in g, the module attains a feature $\hat{g}_i$ as seen in Equation 6.1:

$$\hat{g}_i = \sum_j A_{ij} h_j \quad A_{ij} = \frac{\exp(g_i^T h_j)}{\sum_k \exp(g_i^T h_k)} \tag{6.1}$$

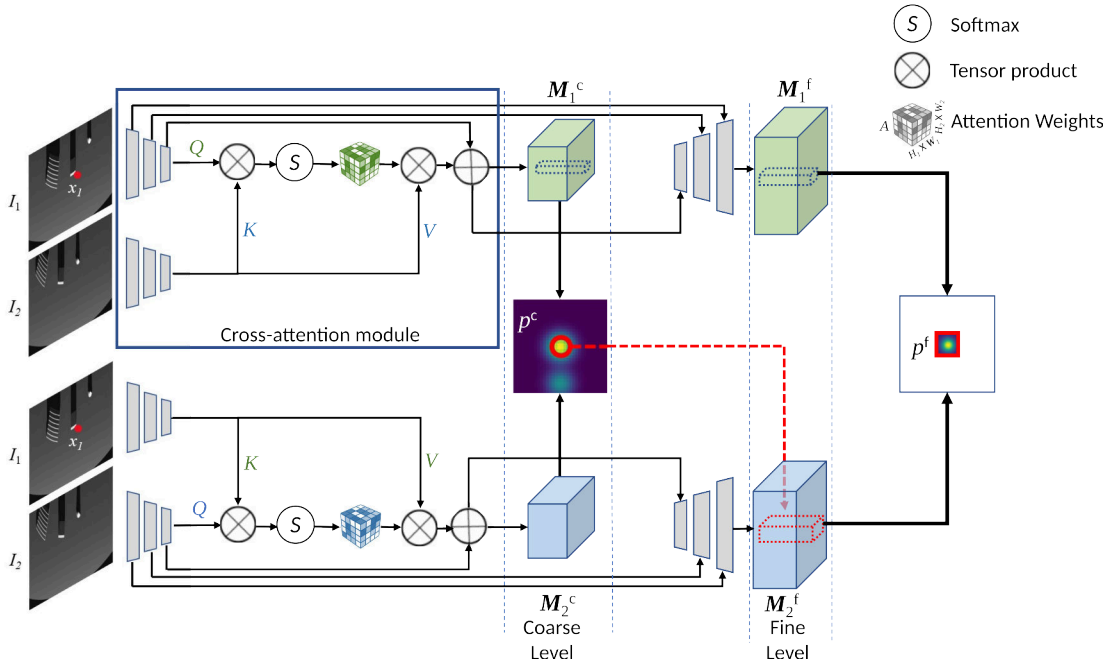


Figure 6.2: Cross-attention Module applied at the encoder level in SONIC: Feature maps from the ResNet layer 3 are cross-attended to each other. Using residual connections, we obtain a conditioned layer 3 feature map, which is used to derive the coarse (M^c) and fine (M^f) layer maps. This process is repeated for the second image, by reversing the key, query and value orders.

where $\hat{g}_i$ is the weighted sum over all spatial features from I_2 . A is the similarity matrix comparing g and h using the inner product followed by the softmax normalization step.

This cross-attention is applied on the ResNet layers in the encoder. This is done for both images, and the attended features are passed through a skip connection, concatenating the attended feature map to the original layer, and is then passed on to the coarse and fine layers.

We utilize multi-headed attention as described in Vaswani et al. [86] which enables the model to attend to different aspects of the input simultaneously, and enhances expressiveness. For embeddings, we look towards rotary positional embeddings [80, 22] to enhance spatial relationships between correspondences and introduce rotational invariance while retaining relative positions. We also implement attention dropout for regularizing the attention weights [87, 48] which discourages excessive interdependence among the diverse contextualized feature vectors.

6.3.2 Architecture

In this section we describe how the cross-attention module is applied to SONIC. The feature maps from the ResNet layer 3 are cross-attended to each other through the attention module. The attended features are added back to the layer 3 map to provide the refined layer 3 map. The refined layer 3 feature map along with layer 1 and layer 2 ResNet feature maps are then passed to the decoder. The layer 3 output undergoes a convolution to form the coarse layer, and the 3 layers

together finally provide the fine layer feature map after a series of further convolutions. This provides an efficient way to apply cross attention which affects both, coarse and fine layer maps, without needing a specific fine layer cross-attention module. Further cross-attention modules could be added to the fine layer outputs, but at a significantly higher computational expense which is avoided by instead applying it before the coarse layer.

The cross-attention modules are used to condition the descriptors generated by SONIC for both images. A visual description of how cross-attention is applied is shown in Figure 6.2. As seen in the figure, the output of the cross-attention module leads to produce coarse and fine feature maps for image 1, attended by image 2. The order is then reversed to provide the coarse and fine maps for image 2 attended by image 1. The rest of the network remains largely unchanged from SONIC, which includes the sonar specific epipolar and cyclic loss functions introduced in Chapter 5. To diminish the network learning to find correspondence in shadowed or occluded regions, a mask is introduced which checks the ground truth correspondence pixels lying in the second image. If these pixels fall under a certain intensity threshold, they are considered to be lying in shadows, and are reweighed to reflect a bad correspondence.

6.3.3 Data and Training

Continuing with the application focus on seafloor mapping, we expand the dataset prepared for SONIC. We continue to configure our simulated images on the Blueprint Subsea M1200d's [5]. We diversify the dataset with multiple scenes captured in both high and low frequency modes, with varying noise and maximum range values. Range values now vary from 10 to 25 meters. As before, sonar images were gathered from 1-4m above the floor at a downward pitch of 10-20^0. Multiple trajectories with varying rotation and translation offsets and image parameters are used to generate randomized pairs. Each image has an associated pose, and metadata information. This metadata information contains all image parameters like azimuth, elevation and range bins, as well as image width and height. The dataset used in training amounted to 550K image pairs, with 60K validation pairs. The network was trained for ≈ 6 epochs with a batch size of 14 pairs, using a Nvidia GeForce 4090 RTX with 24GB memory.

6.4 Evaluation

Similar to how we assessed SONIC, we compare matching performance between C-SONIC, SONIC, and the camera-image trained LightGlue model [49]. All three methods use SuperPoint keypoints and require a keypoint detector, unlike other detector-free matchers [81, 89].

Assuming a planar scene we evaluate the number of matched keypoints that are considered to be inliers for each LightGlue, SONIC and C-SONIC. Inliers are classified by projecting the keypoints from the reference image using the ground truth relative pose information as described in Section 5.3.1 onto the query image. The threshold selected for simulation images is 12 pixels, which corresponds to roughly 0.23m in range or 3° in bearing for all images. We increase this threshold for real images to be 20 pixels, which corresponds to 0.2m and 5° in range and bearing respectively. To show the benefit of our method towards downstream tasks such as feature-based SLAM, we evaluate performance in relative sensor pose recovery using the two-view acoustic

Method	Simulation - Low Variation	Simulation - High Variation	Test Tank
LightGlue	33.39%	12.39%	36.25%
SONIC	37.28%	28.77%	47.09%
C-SONIC	**40.93%**	**31 .76%**	**64.62%**

Table 6.1: Percentage of Inliers

bundle adjustment framework to recover sensor pose information as presented by Westman et al. [92].

It is typical for hovering underwater vehicles to be equipped with pressure sensors for depth estimation, and to be limited to planar motion [36]. Most graph-based solutions for underwater SLAM would thus model *Z*, *pitch* and *roll* as a separate unary factor, independent to planar motions in *X*, *Y*, and *Yaw* [83]. Thus we focus mainly on evaluating the average absolute error in planar translation (*xy*) and rotation (*yaw*).

We perform a Z-test on the distance between the corresponding projected and matched query keypoints. We prune matches not falling under a 2σ threshold for all three methods. Prior pose estimates are derived from corrupted ground truth data. These corrupted poses are used to further filter matches. We utilize the sonar-epipolar loss function as a cost function for outlier rejection as seen in 6.1. The noisy pose prior and the filtered matched keypoints from the reference and query images are passed to the acoustic bundle adjustment optimizer to obtain relative pose difference between the frames.

We present and discuss results for our simulated datasets and real world data from a test tank. We divide our sampled simulated dataset for evaluation into two groups, low and high variation. Both subsets contain randomized pairs containing images from both frequency modes, multiple maximum range parameters and varying sensor origin poses. The low variation group contains pairs with *yaw* $< \pm 10^{\circ}$ and *x,y* $< \pm 1.5$m. The large variation group contains pairs with *yaw* $< \pm 40^{\circ}$ and *x,y* $< \pm 10$m. The test tank data was collected as described before in SONIC, with different objects in the test tank.

In Table 6.1 for the inlier test, we see that C-SONIC benefits from the cross-attention layer added into the network. This improvement is more pronounced in real world scenarios, which differ significantly from the simulated training data. The cross-attended features offer more robustness and accuracy. In Table 6.2 we see the results from the two-view acoustic bundle adjustment. As evident by the lower mean error for both small and large variation test groups, C-SONIC is able to provide more accurate matches, reducing the dependency on computationally expensive robust data association algorithms like JCBB [92]. An example of the matches produced by the methods for an image pair with different frequency modes can be seen in Fig. 6.1. We see that while SONIC does find good matches, C-SONIC, is able to find more accurate and diverse matches which are not concentrated to regions. This is especially beneficial for the two-view bundle adjustment optimization, where well spreadout matches provide a better constrained solution space. As before, LightGlue struggles to perform when the pose variation increases, as well as the difference introduced by the variations in the frequency mode.

Method	Simulated Data			
	Small Variation		Large Variation	
	Translation (m)	Rotation (rad)	Translation (m)	Rotation (rad)
LightGlue	μ: 4.34, σ: 6.17	μ: 0.88, σ: 1.12	μ: 8.27, σ: 9.51	μ: 0.49, σ: 0.82
SONIC	μ: 2.08, σ: 1.78	μ: 0.86, σ: 0.93	μ: 3.24, σ: 1.93	μ: 0.39, σ: 0.49
C-SONIC	μ: **0.79,** σ**: 1.87**	μ: **0.41,** σ: **0.82**	μ**: 0.90,** σ**: 1.44**	μ**: 0.18,** σ**: 0.31**

Table 6.2: Planar Translation and Rotational Accuracy

6.5 Discussions

The attention mechanisms are very versatile and can be placed at multiple steps. However there are several considerations needed, especially with respect to computational costs during training and inference. A self-attention layer prior to cross-attention was explored. While it performed better than SONIC, it did not do better than using a single cross-attention layer as deployed in C-SONIC. This also leads to slower inference speeds due to the additional complexity.

An additional cross-attention layer could also be applied to the fine layer feature map. However due to the feature map depth at the fine layer, it requires more memory to train which significantly increases costs. The cross-attention implementation in C-SONIC allows the encoder level attended features to directly influence both the coarse and fine layer feature maps in a far more efficient manner.

Lastly, loss re-weighting experiments determined that biasing more towards the epipolar loss is counter-productive, and the weights used for SONIC remain as the optimal choice.

Algorithm 6.1 Outlier Rejection using Epipolar Cost Function

procedure EPIPOLAR_OUTLIER_REJECTION($mkpts1$, $mkpts2$, $pose1$, $pose2$, $meta1$, $meta2$, num_iters, th)
  $best\_inliers \leftarrow$ empty list
  for $i = 1$ to num_iters **do**
    $subset\_indices \leftarrow$ randomly select $len(mkpts1)/4$ indices from $mkpts1$ with replacement
    $s\_mkpts1, s\_mkpts2 \leftarrow mkpts1[subset\_indices], mkpts2[subset\_indices]$
    $epipolar\_cost \leftarrow$ sonar_epipolar_cost($s\_mkpts1, s\_mkpts2, meta1, meta2, pose1, pose2$)
    $epipolar\_cost \leftarrow epipolar\_cost / \max(epipolar\_cost)$
    $inlier\_indices \leftarrow$ indices where $epipolar\_cost < th$
    $inlier\_mkpts1, inlier\_mkpts2 \leftarrow mkpts1[inlier\_indices], mkpts2[inlier\_indices]$
    if $length(inlier\_indices) > length(best\_inliers)$ **then**
      $best\_inliers \leftarrow inlier\_indices$
    end if
  end for
  return $best\_inliers$
end procedure

6.6 Conclusions

In this chapter, we introduced C-SONIC, an evolution of the SONIC architecture designed and trained to find correspondences between sonar image pairs which can differ not only in sensor pose, but also image characteristics. Thus, we present a general purpose model capable of finding correspondences between images in both frequency modes, and larger variability in range values for the Blueprint Subsea Oculus. This is achieved through cross-attention mechanisms inspired from visual transformer architectures. The cross-attention layer implemented in C-SONIC allows an efficient way of generating attended feature maps which directly influence the coarse and fine layers with marginal impact on the inference speeds. Along with the pre-trained model, and associated training data for the Blueprint Subsea family of imaging sonars, this work also gives a general framework for other imaging sonars. By providing new data in the same format, the network can be re-trained to learn correspondences between different makes of imaging sonars.

The model is occasionally prone to corresponding to occluded or shadowed regions. The architecture can benefit from utilizing a regressor as seen in CD-UNet [96]. This adds an additional MLP to discriminate matches against pseudo-groundtruth correspondences to further tune the confidence levels.

Chapter 7

Discussion

7.1 Summary

This thesis explores underwater localization and mapping, particularly keeping cost-effective AUVs in mind. To achieve this goal, I have worked on implementing several different algorithms and frameworks for specific problems which impact low-cost AUVs.

For AUVs relying on cheaper inertial navigation sensors, I proposed the degeneracy aware factor, which allows the factor graph to optimize loop closures only in directions that are well conditioned. This selective optimization allows for greater robustness to sensor noise.

For the localization of low cost and low power bio-inspired robotic fish, I proposed an acoustic localization framework which uses pseudoranging to help localize multiple small robots within a volume surrounded by a known constellation of beacons. This framework provides on-chip localization, at a fractional cost using off the shelf components.

For improving feature based SLAM using imaging sonars, I identified the lack of reliability from traditional feature keypoints and descriptors used for correspondence matching. To remedy this, I proposed a novel pose supervised network to learn feature correspondences between two sonar images with the same image parameters. The network, SONIC, achieved this through two novel loss functions, the sonar-epipolar, and the sonar-cyclic loss function and provided robust matches under significant viewpoint variations.

The framework provided by SONIC could be expanded to assist multi-robot SLAM, with different sensors available to each. I proposed C-SONIC, an extension to SONIC which uses cross-attention to learn correspondences between two images having different frequency modes as well as parameters such as maximum range, azimuth and elevation. This allows a robot with a cheaper, low frequency sonar to localize itself in a map generated by a more expensive, high frequency sonar.

7.2 Future Directions

Underwater robotics is an exciting, but niche field. The barriers to scalable research and development are getting lowered every passing year. This thesis lays the ground work for several key problems underwater robots face, and tries to generalize the frameworks and solutions to cater to

low-cost AUVs. The provided frameworks, models and training datasets aim to help accelerate further development in the field.

For the works presented in Chapter 5 and Chapter 6, a key improvement could be made in keypoint detection. The current system relies on camera image based detectors or models to identify keypoints in the reference image. I suggest the use of detector free methods, as seen in the work by Sun et al. [81] where a transformer architecture is used to find correspondences between images without the need for explicit keypoints. The downside here is the added computational costs, not only for training but also inference. However, given the progress made on board Neural Processing Units (NPU), these models could very soon be run in real-time on smaller System on Chips (SOCs), which could mitigate the inference penalty commonly observed on large models.

The availability of simulators which are able to generate customizable data, with pose information, opens the possibility of a number of future projects related to underwater SLAM and perception. Future research could explore Birds-Eye-View (BEV) solutions to localization using forward facing cameras and satellite imagery [47, 98]. A similar architecture can be used to localize a hovering AUV with imaging sonar in a map created by long-range AUVs using side-scan sonars. Further investigations into cross-modal fusion techniques using cross-attention [45] to find correspondences between camera and imaging sonar, or monocular depth and imaging sonar could yield valuable insights. Lastly, increased collaborations and greater focus towards the collection of real-world data to help supplement simulated datasets used for training, as well as to create well crafted datasets used for benchmarking purposes as the field sees a boost in research would be highly beneficial.

Bibliography

[1] P. F. Alcantarilla, J. Nuevo, and A. Bartoli. "Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces". In: *British Machine Vision Conf. (BMVC)*. Bristol, UK, 2013.

[2] D. Bahdanau, K. Cho, and Y. Bengio. "Neural Machine Translation by Jointly Learning to Align and Translate". In: *CoRR* abs/1409.0473 (2014). URL: https://api.semanticscholar.org/CorpusID:11212020.

[3] F. Berlinger, M. Gauci, and R. Nagpal. "Implicit coordination for 3D underwater collective behaviors in a fish-inspired robot swarm". In: *Science Robotics* 6 (Jan. 2021), eabd8668. DOI: 10.1126/scirobotics.abd8668.

[4] G. Blewitt. *Basics of the GPS Technique: Observation Equations*. 2000.

[5] Blueprint Subsea. *Blueprint Subsea Oculus M1200d*. https://blueprintsubsea.com/oculus/oculus-m-series.

[6] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. Leonard. "Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age". In: *IEEE Trans. Robotics*. Dec. 2016, pp. 1309–1332.

[7] Y. Chen and G. Medioni. "Object Modeling by Registration of Multiple Range Images". In: *Image Vision Comput.* 10 (Jan. 1992), pp. 145–155. DOI: 10.1109/ROBOT.1991.132043.

[8] H. Cho, S. Yeon, H. Choi, and N. Doh. "Detection and Compensation of Degeneracy Cases for IMU-Kinect Integrated Continuous SLAM with Plane Features". In: *Sensors* 18.4 (2018). ISSN: 1424-8220. DOI: 10.3390/s18040935. URL: http://www.mdpi.com/1424-8220/18/4/935.

[9] F. Dellaert and M. Kaess. "Factor Graphs for Robot Perception". In: *Foundations and Trends in Robotics* 6.1-2 (Aug. 2017). http://dx.doi.org/10.1561/2300000043, pp. 1–139.

[10] J. DelPreto, R. Katzschmann, R. MacCurdy, and D. Rus. “A Compact Acoustic Communication Module for Remote Control Underwater”. In: *WUWNet 2015: Proceedings of the 10th International Conference on Underwater Networks and Systems*. Oct. 2015, pp. 1–7. DOI: 10.1145/2831296.2831337.

[11] D. DeTone, T. Malisiewicz, and A. Rabinovich. “Superpoint: Self-supervised interest point detection and description”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2018, pp. 224–236.

[12] A. Dewan, T. Caselitz, and W. Burgard. “Learning a Local Feature Descriptor for 3D LiDAR Scans”. In: Oct. 2018, pp. 4774–4780. DOI: 10.1109/IROS.2018.8594420.

[13] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=YicbFdNTTy.

[14] R. M. Eustice, L. L. Whitcomb, H. Singh, and M. Grund. “Experimental Results in Synchronous-Clock One-Way-Travel-Time Acoustic Navigation for Autonomous Underwater Vehicles”. In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2007, pp. 4257–4264. DOI: 10.1109/ROBOT.2007.364134.

[15] E. M. Fischell, N. R. Rypkema, and H. Schmidt. “Relative Autonomy and Navigation for Command and Control of Low-Cost Autonomous Underwater Vehicles”. In: *IEEE Robotics and Automation Letters* 4.2 (2019), pp. 1800–1806. DOI: 10.1109/LRA.2019.2896964.

[16] General Dynamics Mission Systems. *Bluefin HAUV*. https://gdmissionsystems.com/products/underwater-vehicles/bluefin-hauv.

[17] P. Gleize, W. Wang, and M. Feiszli. “SiLK – Simple Learned Keypoints”. In: *ICCV*. 2023.

[18] S. Gode*, A. Hinduja*, and M. Kaess. “SONIC: Sonar Image Correspondence using Pose Supervised Learning for Imaging Sonars”. In: *Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA)*. Yokohama, Japan, 2024.

[19] J. Grabek and B. Cyganek. “Speckle Noise Filtering in Side-Scan Sonar Images Based on the Tucker Tensor Decomposition”. In: *Sensors* 19 (June 2019), p. 2903. DOI: 10.3390/s19132903.

[20] P. Hansen, P. Corke, W. Boles, and K. Daniilidis. “Scale invariant feature matching with wide angle images”. In: *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2007, pp. 1689–1694.

[21] K. He, X. Zhang, S. Ren, and J. Sun. *Deep Residual Learning for Image Recognition*. 2015. arXiv: 1512.03385 [cs.CV].

[22] B. Heo, S. Park, D. Han, and S. Yun. “Rotary Position Embedding for Vision Transformer”. In: *arXiv preprint arXiv:2403.13298* (2024).

[23] T. Herring. “Geodetic applications of GPS”. In: *Proceedings of the IEEE* 87.1 (1999), pp. 92–110. DOI: 10.1109/5.736344.

[24] A. Hinduja, B.-J. Ho, and M. Kaess. “Degeneracy-aware factors with applications to underwater SLAM”. In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. Macau, 2019.

[25] A. Hinduja, Y. Ohm, J. Liao, C. Majidi, and M. Kaess. “Acoustic Localization and Communication Using a MEMS Microphone for Low-cost and Low-power Bio-inspired Underwater Robots”. In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2022, pp. 5470–5477. DOI: 10.1109/IROS47612.2022.9981355.

[26] B.-J. Ho, P. Sodhi, P. V. Teixeira, M. Hsiao, T. Kusnur, and M. Kaess. “Virtual Occupancy Grid Map for Submap-based Pose Graph SLAM and Planning in 3D Environments”. In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. Oct. 2018.

[27] M. Hsiao, E. Westman, and M. Kaess. “Dense Planar-Inertial SLAM with Structural Constraints”. In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. May 2018.

[28] G. Hu, S. Huang, L. Zhao, A. Alempijevic, and G. Dissanayake. “A robust RGB-D SLAM algorithm”. In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. Oct. 2012, pp. 1714–1719. DOI: 10.1109/IROS.2012.6386103.

[29] T. A. Huang and M. Kaess. “Towards Acoustic Structure from Motion for Imaging Sonar”. In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. Hamburg, Germany, Oct. 2015, pp. 758–765.

[30] N. Hurtós, D. Ribas, X. Cufí, Y. Petillot, and J. Salvi. “Fourier-based Registration for Robust Forward-looking Sonar Mosaicing in Low-visibility Underwater Environments”. In: *J. of Field Robotics* 32.1 (2014), pp. 123–151.

[31] O. K. Isik, J. Hong, I. Petrunin, and A. Tsourdos. “Integrity Analysis for GPS-Based Navigation of UAVs in Urban Environment”. In: *Robotics* 9.3 (2020). ISSN: 2218-6581.

DOI: 10.3390/robotics9030066. URL: https://www.mdpi.com/2218-6581/9/3/66.

[32] M. V. Jakuba, J. C. Kinsey, J. W. Partan, and S. E. Webster. "Feasibility of low-power one-way travel-time inverted ultra-short baseline navigation". In: *OCEANS 2015 - MTS/IEEE Washington*. 2015, pp. 1–10. DOI: 10.23919/OCEANS.2015.7401992.

[33] J. Jaybhay and R. Shastri. "A Study of Speckle Noise Reduction Filters". In: *Signal & Image Processing : An International Journal* 6 (June 2015), pp. 71–80. DOI: 10.5121/sipij.2015.6306.

[34] S. Jetley, N. A. Lord, N. Lee, and P. Torr. "Learn to Pay Attention". In: *International Conference on Learning Representations*. 2018. URL: https://openreview.net/forum?id=HyzbhfWRW.

[35] E. K. Jorgensen, T. I. Fossen, T. H. Bryne, and I. Schjolberg. "Underwater Position and Attitude Estimation Using Acoustic, Inertial, and Depth Measurements". In: *IEEE Journal of Oceanic Engineering* 45.4 (2020), pp. 1450–1465. DOI: 10.1109/JOE.2019.2933883.

[36] M. Kaess, H. Johannsson, B. Englot, F. Hover, and J. Leonard. "Towards Autonomous Ship Hull Inspection using the Bluefin HAUV". In: *Ninth International Symposium on Technology and the Mine Problem*. May 2010.

[37] M. Kaess, A. Ranganathan, and F. Dellaert. "iSAM: Incremental Smoothing and Mapping". In: *IEEE Trans. Robotics* 24.6 (Dec. 2008), pp. 1365–1378.

[38] S. Karabchevsky, D. Kahana, O. Ben-Harush, and H. Guterman. "FPGA-Based Adaptive Speckle Suppression Filter for Underwater Imaging Sonar". In: *IEEE Journal of Oceanic Engineering* 36.4 (2011), pp. 646–657. DOI: 10.1109/JOE.2011.2157729.

[39] R. K. Katzschmann, J. DelPreto, R. MacCurdy, and D. Rus. "Exploration of underwater life with an acoustically controlled soft robotic fish". In: *Science Robotics* 3.16 (2018). DOI: 10.1126/scirobotics.aar3449. eprint: https://www.science.org/doi/pdf/10.1126/scirobotics.aar3449. URL: https://www.science.org/doi/abs/10.1126/scirobotics.aar3449.

[40] J. Kinsey, R. Eustice, and L. Whitcomb. "A Survey of Underwater Vehicle Navigation: Recent Advances and New Challenges". In: *IFAC Conference of Manoeuvering and Control of Marine Craft* 88 (Jan. 2006), pp. 1–12.

[41] E. Lassiter. "Navstar Global Positioning System: A Satellite Based Microwave Navigation System". In: *IEEE-MTT-S International Microwave Symposium*. 1975, pp. 334–334. DOI: 10.1109/MWSYM.1975.1123385.

[42] P. Lazik and A. Rowe. “Indoor pseudo-ranging of mobile devices using ultrasonic chirps”. In: *ACM Conf. on Embedded Networked Sensor Systems (SenSys)*. Nov. 2012, pp. 99–112. DOI: 10.1145/2426656.2426667.

[43] *Leica TS16: Robotic Total Station*. URL: https://leica-geosystems.com/en-us/products/total-stations/robotic-total-stations/leica-ts16.

[44] J. Li, M. Kaess, R. Eustice, and M. Johnson-Roberson. “Pose-graph SLAM using forward-looking sonar”. In: *IEEE Robotics and Automation Letters* 3.3 (2018), pp. 2330–2337.

[45] Y. Li, A. W. Yu, T. Meng, B. Caine, J. Ngiam, D. Peng, J. Shen, Y. Lu, D. Zhou, Q. V. Le, A. Yuille, and M. Tan. “DeepFusion: Lidar-Camera Deep Fusion for Multi-Modal 3D Object Detection”. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 17161–17170. DOI: 10.1109/CVPR52688.2022.01667.

[46] Z. Li and N. Snavely. “MegaDepth: Learning Single-View Depth Prediction from Internet Photos”. In: *Computer Vision and Pattern Recognition (CVPR)*. 2018.

[47] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Y. Qiao, and J. Dai. “BEVFormer: Learning Birdâs-Eye-View Representation from Multi-Camera Images via Spatiotemporal Transformers”. In: ().

[48] Z. Lin, P. Liu, L. Huang, J. Chen, X. Qiu, and X. Huang. “DropAttention: A Regularization Method for Fully-Connected Self-Attention Networks”. In: *ArXiv* abs/1907.11065 (2019). URL: https://api.semanticscholar.org/CorpusID:198895166.

[49] P. Lindenberger, P.-E. Sarlin, and M. Pollefeys. “LightGlue: Local Feature Matching at Light Speed”. In: *ICCV*. 2023.

[50] K. Low. *Linear least-squares optimization for point-toplane ICP surface registration*. Tech. rep. 2004.

[51] T. Luong, H. Pham, and C. D. Manning. “Effective Approaches to Attention-based Neural Machine Translation”. In: *ArXiv* abs/1508.04025 (2015). URL: https://api.semanticscholar.org/CorpusID:1998416.

[52] K. Ma, J. Zhu, T. J. Dodd, R. Collins, and S. R. Anderson. “Robot Mapping and Localisation for Feature Sparse Water Pipes Using Voids as Landmarks”. In: *Towards Autonomous Robotic Systems*. Ed. by C. Dixon and K. Tuyls. Cham, 2015, pp. 161–166.

[53] A. Mallios, E. Vidal, R. Campos, and M. Carreras. “Underwater caves sonar data set”. In: *The International Journal of Robotics Research* 36.12 (2017), pp. 1247–1251. DOI:

10.1177/0278364917732838. eprint: https://doi.org/10.1177/0278364917732838. URL: https://doi.org/10.1177/0278364917732838.

[54] A. D. Marchese, C. D. Onal, and D. Rus. "Autonomous Soft Robotic Fish Capable of Escape Maneuvers Using Fluidic Elastomer Actuators." In: *Soft robotics* 1 1 (2014), pp. 75–87.

[55] P. Mikhalevsky, B. Sperry, K. Woolfe, M. Dzieciuch, and P. Worcester. "Deep ocean long range underwater navigation". In: *The Journal of the Acoustical Society of America* 147 (Apr. 2020), pp. 2365–2382. DOI: 10.1121/10.0001081.

[56] M. Mur-Artal Raúl, J. M. M., and J. D. Tardós. "ORB-SLAM: a Versatile and Accurate Monocular SLAM System". In: *IEEE Transactions on Robotics* 31.5 (2015), pp. 1147–1163. DOI: 10.1109/TRO.2015.2463671.

[57] S. Negahdaripour. "On 3-D Motion Estimation From Feature Tracks in 2-D FS Sonar Video". In: *IEEE Trans. Robotics* 29.4 (Aug. 2013), pp. 1016–1030.

[58] S. Negahdaripour. "Analyzing Epipolar Geometry of 2-D Forward-Scan Sonar Stereo for Matching and 3-D Reconstruction". In: *Proc. of the IEEE/MTS OCEANS Conf. and Exhibition*. 2018, pp. 1–10.

[59] P. C. Ng and S. Henikoff. "SIFT: Predicting amino acid changes that affect protein function". In: *Nucleic acids research* 31.13 (2003), pp. 3812–3814.

[60] A. Novak, P. Cisar, M. Bruneau, P. Lotton, and L. Simon. "Localization of sound-producing fish in a water-filled tank". In: *The Journal of the Acoustical Society of America* 146 (Dec. 2019), pp. 4842–4850. DOI: 10.1121/1.5138607.

[61] A. Oliveira, B. Ferreira, and N. Cruz. "A Performance Analysis of Feature Extraction Algorithms for Acoustic Image-Based Underwater Navigation". In: *J. Mar. Sci. Eng.* 9 (2021), p. 361. URL: https://doi.org/10.3390/jmse9040361.

[62] D. A. Paley and N. M. Wereley. *Bioinspired Sensing, Actuation, and Control in Underwater Soft Robotic Systems*. Springer, 2021.

[63] K. Pathak, A. Birk, N. Vaskevicius, and J. Poppinga. "Fast Registration Based on Noisy Planes With Unknown Correspondences for 3-D Mapping". In: *IEEE Trans. Robotics* 26.3 (2010), pp. 424–441.

[64] Z. J. Patterson, A. P. Sabelhaus, K. Chin, T. Hellebrekers, and C. Majidi. "An Untethered Brittle Star-Inspired Soft Robot for Closed-Loop Underwater Locomotion". In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. 2020, pp. 8758–8764. DOI: 10.1109/IROS45743.2020.9341008.

[65] L. Paull, S. Saeedi, M. Seto, and H. Li. "AUV Navigation and Localization: A Review". In: *IEEE Journal of Oceanic Engineering* 39.1 (2014), pp. 131–149. DOI: 10.1109/JOE.2013.2278891.

[66] E. Potokar, S. Ashford, M. Kaess, and J. G. Mangelson. "Holoocean: An underwater robotics simulator". In: *2022 International Conference on Robotics and Automation (ICRA)*. IEEE. 2022, pp. 3040–3046.

[67] E. Potokar, K. Lay, K. Norman, D. Benham, T. B. Neilsen, M. Kaess, and J. G. Mangelson. "HoloOcean: Realistic Sonar Simulation". In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2022, pp. 8450–8456. DOI: 10.1109/IROS47612.2022.9981119.

[68] A. Raj and A. Thakur. "Fish-inspired robots: design, sensing, actuation, and autonomy-a review of research". In: *Bioinspiration & biomimetics* 11.3 (2016), p. 031001.

[69] P. O. C. S. Ribeiro, M. M. dos Santos, P. L. J. Drews, S. S. C. Botelho, L. M. Longaray, G. G. Giacomo, and M. R. Pias. "Underwater Place Recognition in Unknown Environments with Triplet Based Acoustic Image Retrieval". In: *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. 2018, pp. 524–529. DOI: 10.1109/ICMLA.2018.00084.

[70] G. Robotics. *ULS-500 PRO Dynamic Underwater Laser Scanner*. http://www.tritech.co.uk/product/gemini-720is-1000m-or-4000m.

[71] Z. Rong and N. Michael. "Detection and prediction of near-term state estimation degradation via online nonlinear observability analysis". In: *IEEE International Symposium on Safety Security and Rescue Robotics (SSRR)*. Oct. 2016, pp. 28–33.

[72] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. "ORB: An efficient alternative to SIFT or SURF". In: *2011 International conference on computer vision*. Ieee. 2011, pp. 2564–2571.

[73] R. Rusu and S. Cousins. "3D is here: Point Cloud Library (PCL)". In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. Shanghai, China, May 2011.

[74] N. R. Rypkema, E. M. Fischell, and H. Schmidt. "One-way travel-time inverted ultra-short baseline localization for low-cost autonomous underwater vehicles". In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. 2017, pp. 4920–4926. DOI: 10.1109/ICRA.2017.7989570.

[75] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich. "Superglue: Learning feature matching with graph neural networks". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 4938–4947.

[76] Y. S. Shin, Y. Lee, H. T. Choi, and A. Kim. “Bundle adjustment from sonar images and SLAM application for seafloor mapping”. In: *Proc. of the IEEE/MTS OCEANS Conf. and Exhibition*. Oct. 2015, pp. 1–6.

[77] Y.-S. Shin, Y. Lee, H.-T. Choi, and A. Kim. “Bundle adjustment from sonar images and SLAM application for seafloor mapping”. In: *OCEANS 2015 - MTS/IEEE Washington*. 2015, pp. 1–6. DOI: 10.23919/OCEANS.2015.7401963.

[78] D. Singh and M. Valdenegro-Toro. “The Marine Debris Dataset for Forward-Looking Sonar Semantic Segmentation”. In: *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*. Los Alamitos, CA, USA: IEEE Computer Society, 2021, pp. 3734–3742. DOI: 10.1109/ICCVW54120.2021.00417. URL: https://doi.ieeecomputersociety.org/10.1109/ICCVW54120.2021.00417.

[79] Sound Metrics Corporation. *SoundMetrics Didson 300 Specifications*. http://www.soundmetrics.com/Products/DIDSON-Sonars/DIDSON-300-m/.

[80] J. Su, Y. Lu, S. Pan, B. Wen, and Y. Liu. “RoFormer: Enhanced Transformer with Rotary Position Embedding”. In: *CoRR* abs/2104.09864 (2021). arXiv: 2104.09864. URL: https://arxiv.org/abs/2104.09864.

[81] J. Sun, Z. Shen, Y. Wang, H. Bao, and X. Zhou. “LoFTR: Detector-free local feature matching with transformers”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 8922–8931.

[82] Y. Taguchi, Y.-D. Jian, S. Ramalingam, and C. Feng. “Point-plane SLAM for hand-held 3D sensors”. In: May 2013, pp. 5182–5189. ISBN: 978-1-4673-5641-1. DOI: 10.1109/ICRA.2013.6631318.

[83] P. Teixeira, M. Kaess, F. Hover, and J. Leonard. “Underwater inspection using sonar-based volumetric submaps”. In: *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*. Daejeon, Korea, Oct. 2016, pp. 4288–4295.

[84] M. J. Tribou, D. W. L. Wang, and S. L. Waslander. “Degenerate Motions in Multicamera Cluster SLAM with Non-overlapping Fields of View”. In: *CoRR* abs/1506.07597 (2015). arXiv: 1506.07597. URL: http://arxiv.org/abs/1506.07597.

[85] P. Tueller, R. Kastner, and R. Diamant. “A Comparison of Feature Detectors for Underwater Sonar Imagery”. In: *OCEANS 2018 MTS/IEEE Charleston*. 2018, pp. 1–6. DOI: 10.1109/OCEANS.2018.8604786.

[86] A. Vaswani, N. M. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. "Attention is All you Need". In: *Neural Information Processing Systems*. 2017. URL: https://api.semanticscholar.org/CorpusID:13756489.

[87] L. Wan, M. Zeiler, S. Zhang, Y. Le Cun, and R. Fergus. "Regularization of Neural Networks using DropConnect". In: *Proceedings of the 30th International Conference on Machine Learning*. Ed. by S. Dasgupta and D. McAllester. Vol. 28. Proceedings of Machine Learning Research 3. Atlanta, Georgia, USA: PMLR, 2013, pp. 1058–1066. URL: https://proceedings.mlr.press/v28/wan13.html.

[88] Q. Wang, X. Zhou, B. Hariharan, and N. Snavely. "Learning feature descriptors using camera pose supervision". In: *European Conference on Computer Vision*. Springer. 2020, pp. 757–774.

[89] Q. Wang, J. Zhang, K. Yang, K. Peng, and R. Stiefelhagen. *MatchFormer: Interleaving Attention in Transformers for Feature Matching*. 2022. arXiv: 2203.09645 [cs.CV].

[90] E. Westman. "Underwater Localization and Mapping with Imaging Sonar". PhD thesis. Pittsburgh, PA: Carnegie Mellon University, 2019.

[91] E. Westman, A. Hinduja, and M. Kaess. "Feature-Based SLAM for Imaging Sonar with Under-Constrained Landmarks". In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. Brisbane, Australia, May 2018, pp. 3629–3636.

[92] E. Westman and M. Kaess. "Degeneracy-Aware Imaging Sonar Simultaneous Localization and Mapping". In: *IEEE J. of Oceanic Engineering* (2019). DOI: 10.1109/JOE.2019.2937946.

[93] E. Westman and M. Kaess. *Underwater AprilTag SLAM and calibration for high precision robot localization*. Tech. rep. CMU-RI-TR-18-43. Pittsburgh, PA: Carnegie Mellon University, Oct. 2018.

[94] T. Whelan, S. Leutenegger, R. F. Salas-Moreno, B. Glocker, and A. J. Davison. "ElasticFusion: Dense SLAM Without A Pose Graph". In: *Robotics: Science and Systems (RSS)*. Rome, Italy, July 2015.

[95] T. Whelan, J. McDonald, M. Kaess, M. Fallon, H. Johannsson, and J. J. Leonard. "Kintinuous: Spatially Extended KinectFusion". In: *RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras*. 2012.

[96] O. Wiles, S. Ehrhardt, and A. Zisserman. "Co-attention for conditioned image matching". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 15920–15929.

[97] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon. “CBAM: Convolutional Block Attention Module”. In: *Computer Vision â ECCV 2018: 15th European Conference, Munich, Germany, September 8â14, 2018, Proceedings, Part VII*. Munich, Germany: Springer-Verlag, 2018, 3â19. ISBN: 978-3-030-01233-5. DOI: 10.1007/978-3-030-01234-2_1. URL: https://doi.org/10.1007/978-3-030-01234-2_1.

[98] C. Yang, Y. Chen, H. Tian, C. Tao, X. Zhu, Z. Zhang, G. Huang, H. Li, Y. Qiao, L. Lu, J. Zhou, and J. Dai. “BEVFormer v2: Adapting Modern Image Backbones to Bird’s-Eye-View Recognition via Perspective Supervision”. In: *ArXiv* (2022).

[99] J. Yang, Z. Cao, and Q. Zhang. “A fast and robust local descriptor for 3D point cloud registration”. In: *Information Sciences* 346-347 (2016), pp. 163–179. ISSN: 0020-0255. DOI: https://doi.org/10.1016/j.ins.2016.01.095. URL: https://www.sciencedirect.com/science/article/pii/S0020025516300378.

[100] R. Yarlagadda, N. Al-Dhahir, and J. Hershey. “GPS GDOP Metric”. In: *Radar, Sonar and Navigation, IEE Proceedings* 147 (Nov. 2000), pp. 259 –264. DOI: 10.1049/ip-rsn:20000554.

[101] J. Zhang, M. Kaess, and S. Singh. “On Degeneracy of Optimization-based State Estimation Problems”. In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)*. Stockholm, Sweden, May 2016, pp. 809–816.

[102] J. Zhang and S. Singh. “Visual-lidar Odometry and Mapping: Low-drift, Robust, and Fast”. In: *IEEE Intl. Conf. on Robotics and Automation (ICRA)* (May 2015).

[103] W. Zhang, T. Zhou, C. Xu, and M. Liu. *A sift-like feature detector and descriptor for Multibeam Sonar Imaging*. 2021. URL: https://www.hindawi.com/journals/js/2021/8845814/.