Place Recognition and Localization for Multi-Modal Underwater Navigation with Vision and Acoustic Sensors

by

Jie Li

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering: Systems)
in the University of Michigan
2017

Doctoral Committee:

Assistant Professor Matthew Johnson-Roberson, Co-Chair
Associate Professor Ryan M. Eustice, Co-Chair
Associate Professor Jason J. Corso
Professor Jeffrey A. Fessler
Assistant Professor Michael Kaess

ljlijie@umich.edu
ORCID: 0000-0003-0812-780X

ACKNOWLEDGMENTS

I would like to express my special appreciation and thanks to my advisors Dr. Matthew Johnson-Roberson and Dr. Ryan Eustice. You two have been a tremendous combination of mentors for me. I would like to thank you all for encouraging my research and for allowing me to grow as a research scientist. Your advice on both research as well as on my career have been priceless.

I would also like to thank Professor Jeffrey Fessler, Professor Jason Corso and Professor Michael Kaess for serving as my committee members. I appreciate your patience and suggestions for improvement to my dissertation.

I would especially like to thank all the folks in both PeRL and DROP (in the order I met them): Ayoung, Paul, Steve, Gaurav, Nick, Jeff, Ryan, Vittorio, GT, Enric, Alex, Arash, Katie, Steven, Josh, Edurado, Derrick, Wonhui, Gideon, Corina. Thank you all for making my whole PhD journey such an enjoyable one. I will always remember our shared happiness in the lab and places of the field trials, e.g. a hotel near nowhere in Jamaica with no chicken, no beef, no drink or bottled water.

I would like to thank Prof. Savarese, everyone in the Vision Lab and all my friends in EE:System. You helped me go through the hardness of my first year in Michigan and led me into the door of Computer Vision.

Special thanks to my family. Words cannot express how grateful I am to my husband, my parents and Batman. You are always my support in the hard moments. To dear Zhiyuan: It is never easier for a couple to pursue Ph.D. degrees in the same period. All the tough times double. But it is so lucky that we can go through this together. Now we are bound not only by love's blind chance, but also our shared research spirit, our shared taste on fast food and our similar Dota2 rank. To my Batman: You are the best companion when I need to work late. Maybe try not to delete or mess up all my codes when jumping on my laptop in the future.

TABLE OF CONTENTS

Chapter

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

LIST OF ACRONYMS

ASFM acoustic structure from motion

AUV autonomous underwater vehicle

A-KAZE Accelerated-KAZE

ASFM Acoustic Structure From Motion

BA bundle adjustment

BoW bag-of-words

CNN convolutional neural network

DOF degree-of-freedom

DR dead-reckoning

DVL Doppler velocity log

EKF extended Kalman filter

ESEIF Exactly Sparse Extended Information Filter

FAB-MAP fast appearance-based matching

FOV field-of-view

FLS Forward-looking Sonar

GPS global positioning system

GPU graphical processing unit

HAUV hovering autonomous underwater vehicle

HOG histogram of oriented gradients

IMU inertial measurement unit

KF Kalman Filtering

LED light-emitting diode

MAP maximum *a posteriori*

IMU inertial measurement unit

NDT Normal Distribution Transformation

NLP natural language processing

NCC Normalized Cross Correlation

PF Particle Filtering

PR Precision-Recall

RANSAC random sample consensus

ReLu rectified linear unit

SIFT scale-invariant feature transform

SLAM simultaneous localization and mapping

SURF speeded up robust features

SNR Signal-Noise-Ratio

SMC Sequential Monte Carlo

UAV unmanned aerial vehicle

UKF unscented Kalman filter

ABSTRACT

Place recognition and localization are important topics in both robotic navigation and computer vision. They are a key prerequisite for simultaneous localization and mapping (SLAM) systems, and also important for long-term robot operation when registering maps generated at different times. The place recognition and relocalization problem is more challenging in the underwater environment because of four main factors: 1) changes in illumination; 2) long-term changes in the physical appearance of features in the aqueous environment attributable to biofouling and the natural growth, death, and movement of living organisms; 3) low density of reliable visual features; and 4) low visibility in a turbid environment. There is no one perceptual modality for underwater vehicles that can single-handedly address all the challenges of underwater place recognition and localization.

This thesis proposes novel research in place recognition methods for underwater robotic navigation using both acoustic and optical imaging modalities. We develop robust place recognition algorithms using both optical cameras and a Forward-looking Sonar (FLS) for an active visual SLAM system that addresses the challenges mentioned above.

We first design an optical image matching algorithm using high-level features to evaluate image similarity against dramatic appearance changes and low image feature density. A localization algorithm is then built upon this method combining both image similarity and measurements from other navigation sensors, which enables a vehicle to localize itself to maps temporally separated over the span of years.

Next, we explore the potential of FLS in the place recognition task. The weak feature texture and high noise level in sonar images increase the difficulty in making correspondences among them. We learn descriptive image-level features using a convolutional neural network (CNN) with the data collected for our ship hull inspection mission. These features present outstanding performance in sonar image matching, which can be used for effective loop-closure proposal for SLAM as well as multi-session SLAM registration. Building upon this, we propose a pre-linearization approach to leverage this type of general high-dimensional abstracted feature in a real-time recursive Bayesian filtering framework, which results in the first real-time recursive localization framework using this modality.

Finally, we propose a novel pose-graph SLAM algorithm leveraging FLS as the perceptual sensors providing constraints for drift correction. In this algorithm, we address practical problems that arise when using an FLS for SLAM, including feature sparsity, low reliability in data association and geometry estimation. More specifically, we propose a novel approach to pruning out less-informative sonar frames that improves system efficiency and reliability. We also employ local bundle adjustment to optimize the geometric constraints between sonar frames and use the mechanism to avoid degenerate motion patterns.

All the proposed contributions are evaluated with real-data collected for ship hull inspection. The experimental results outperform existent benchmarks. The culmination of these contributions is a system capable of performing underwater SLAM with both optical and acoustic imagery gathered across years under challenging imaging conditions.

CHAPTER 1

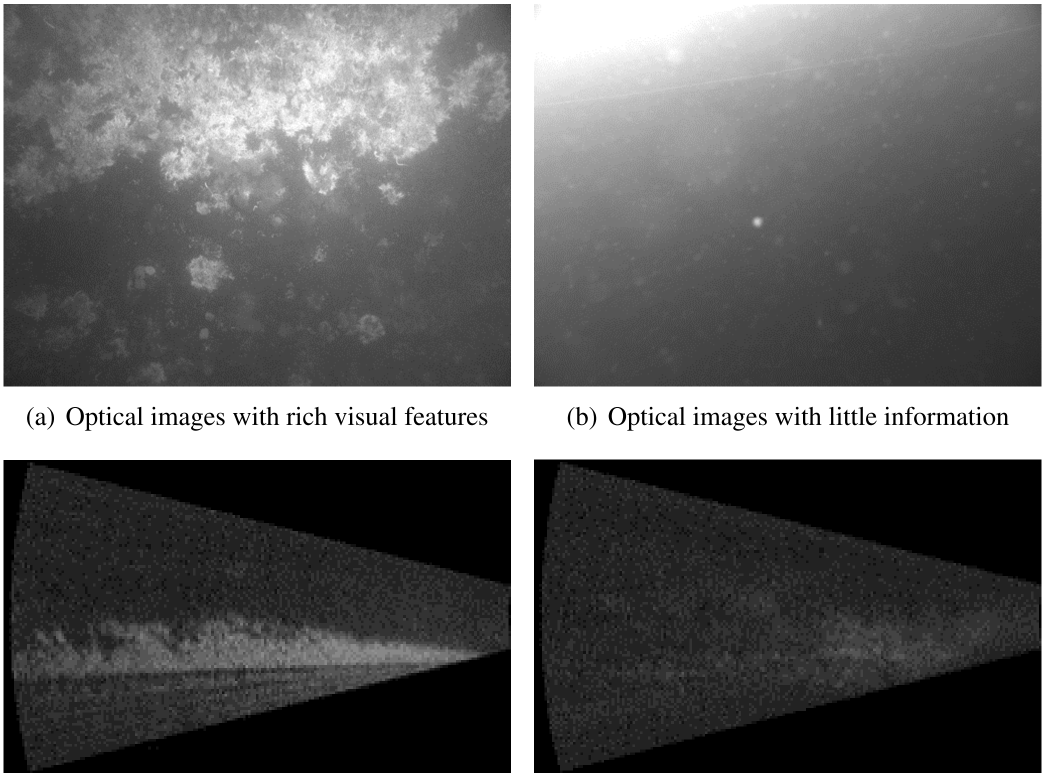
Introduction

1.1 Motivation

1.1.1 Importance and Challenges of underwater robot localization

Mobile robots have found great success in a variety of tasks in human inaccessible environments. Autonomous mobile robots operating in real-world environments must fulfill three core tasks: localization, mapping and planning. Popular techniques for addressing the first task, including localization frameworks based on recursive Bayesian filtering [64, 93] and simultaneous localization and mapping (SLAM) [12, 29], have been proposed and have served as the core component in a wide range of autonomous navigation systems. Representative examples include small-sized unmanned aerial vehicles (UAVs) [10, 20], advanced self-driving cars [100], and autonomous underwater vehicles (AUVs) [8, 67, 82]. Despite the great success of these frameworks, certain applications remain challenging due to the limitations posed by the environment. Among the different research areas and directions in mobile robotics, underwater autonomous navigation is of great importance as the oceans cover 71 percent of the Earth's surface and contain 97 percent of the Earth's water. AUVs provide great potential to revolutionize our access to the oceans, addressing a variety of critical problems such as underwater search and rescue [6, 103], climate change assessment [76], marine habitat monitoring [99] and underwater structure inspection [8, 66]. However, AUV navigation systems face challenges not found in terrestrial or indoor robotic environments. Limited underwater communication prevents the usage of the global positioning system (GPS), which is one of the most popular and powerful sensors for terrestrial robot localization. In addition, other available modalities are also limited due to the difficulties in signal communication underwater.

Both optical cameras and acoustic imaging sonar are important and efficient perceptual modalities for underwater navigation. But neither modality can consistently provide the level of observation needed to support navigation systems. Optical signals can suffer from



(a) Optical images with rich visual features

(b) Optical images with little information

(c) Acoustic images with rich visual features

(d) Acoustic images with little information

Figure 1.1: Examples of optical images and sonar images taken from ship hull inspection applications. The images are taken with a vehicle stand-off of $1.5\ \mathrm{m}$ from the ship hull. The sensor footprints with respect to the ship hull are given in Fig. 1.3. The image in (a) captures an area with biofouling with clear visibility. However, in (b) the camera captures an area with little recognizable texture or features, while the image quality is also decreased by back scattering. Sonar image in (c) contains relatively rich visual features that can be extracted as low level image information, while in (d) the image contains little texture and is mostly covered by noise. With imagery like (b) or (d), it is unlikely that useful information can be extracted for a SLAM system.

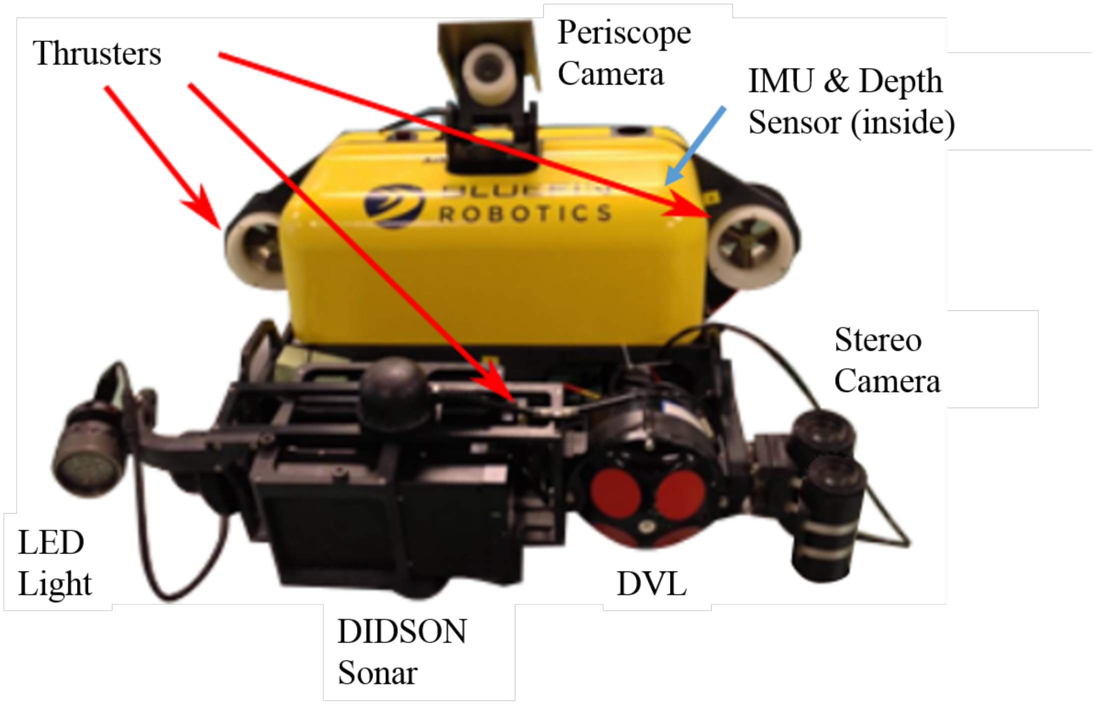


Figure 1.2: Sensor configuration of the HAUV. Figure courtesy of Ozog, Johnson-Roberson, and Eustice [85]

strong back scattering and low visibility due to the turbidity of the underwater environment. Acoustic signals, on the other hand, suffer from low Signal-Noise-Ratio (SNR) due to speckle noise. The low texture detail and ambiguities of the sensor also make it challenging to extract useful information from sonar images. On top of that, the sparse distribution of useful visual features in the environment can limit the information provided by both of these modalities. Thus, real-world underwater robotic systems demand a comprehensive navigation system that could jointly employ different sensor modalities to increase system reliability and robustness.

In this thesis, we propose place recognition and localization approaches for underwater navigation systems and address the challenges present in extreme underwater environments when using both optical and acoustic modalities. In particular, we report on a place recognition approach that leverages optical cameras in the presence of dramatic appearance changes, a real-time localization technique using sonar imagery, and a novel approach to explicitly modeling Forward-looking Sonar (FLS) image features in a SLAM systems with other sensor modalities.

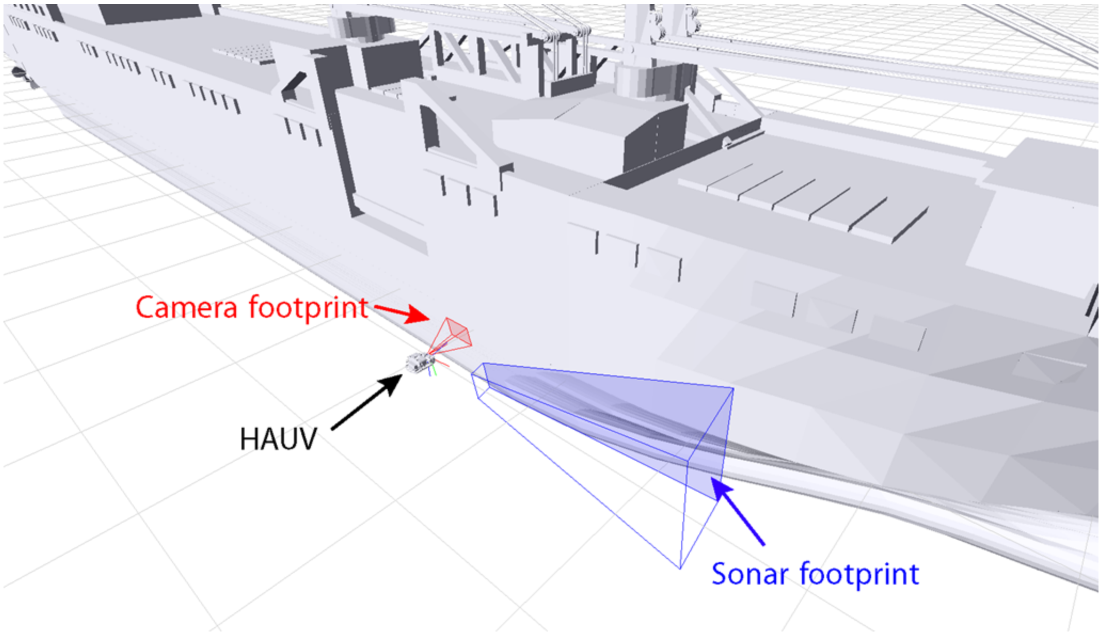


Figure 1.3: Perceptual sensor footprint with respect to a ship hull.

1.1.2 Autonomous Ship Hull Inspection with the HAUV

The main motivation for our work is the application of autonomous ship hull inspections using a hovering autonomous underwater vehicle (HAUV). This is a joint research project between the University of Michigan, Massachusetts Institute of Technology, Carnegie Mellon University, and Bluefin Robotics. The goal of the inspection is to map and virtually reconstruct the below-water portion of a ship hull, in order to detect structural damage or suspicious objects. These tasks are traditionally performed by human divers in a time-consuming and dangerous manner.

The HAUV is a free-floating vehicle developed for underwater, man-made structure inspection. The main feature that sets it apart from other AUVs is its hovering functionality with respect to the targeted structure, which is enabled by its main proprioceptive sensor, the Doppler velocity log (DVL). The DVL is servoed to point orthogonal to the ship hull, which allows the robot to navigate in a hull-relative reference frame. The HAUV also features a bottom-referenced mode with the DVL downward-looking to the sea floor. Fig. 1.2 shows the sensor configuration of the HAUV platform used in the project. A DVL, an inertial measurement unit (IMU), and an on-board depth sensor serve as the main navigation sensors for dead-reckoning (DR). A stereo camera rig for underwater images, a periscope camera for above-water images, and a DIDSON [90] forward-looking imaging sonar are perceptual modalities on the vehicle used to examine the ship hull appearance and structure. The

Table 1.1: Payload characteristics of the HAUV

Prosilica GC1380	12-bit digital stills, fixed-focus, monochrome, 1 Megapixel
Periscope Camera	Monocular Prosilica GC1380 in water-proof housing
Underwater Camera (monocular, pre-2013)	Monocular Prosilica GC1380 in water-proof housing
Underwater Camera (stereo, post-2013)	Two Prosilica GC1380s in separate water-proof bottles, linked via Fast Ethernet
Lighting	520 nm (green) light-emitting diode (LED)
IMU	Honeywell HG1700
Depth	Keller pressure, 1-σ noise at 10 cm
Imaging Sonar	Sound Metrics 1.8 MHz DIDSON
DVL	RDI 1200 kHz Workhorse; also provides four range beams
Thrusting	Five rotor-wound thrusters
Battery	1.5 kWh lithium-ion
Dry Weight	79 kg
Dimensions	1 m × 1 m × 0.45 m

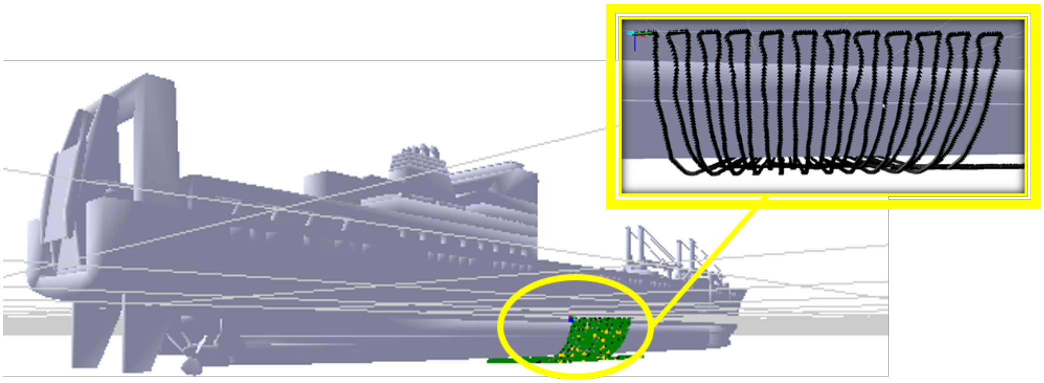


Figure 1.4: Lawn-mowing trajectory pattern during inspection.

specifics of the vehicle payload are given in Table 1.1 as reported by Hover et al. [45].

Although the project targets the inspection and virtual reconstruction of the whole ship hull, in this thesis we primarily focus on the planar portion of the ship hull. In the survey of non-complex areas of the ship-hull, the HAUV follows a default trajectory pattern, as depicted in Fig. 1.4. A real-time visual-based SLAM framework [8] has been proposed in previous work to prevent the localization *drift* due to accumulated error in DR. Due to the limited operation time of the vehicle, which is constrained by its battery capacity, a single mission is not able to survey the whole non-complex area of a ship hull. Multiple mission sessions at different locations are merged together using perceptual observations to achieve full coverage of the area, as depicted in Fig. 1.5. The main goal of this thesis lies in enhancing the capability of the HAUV localization system against extreme environment conditions, using advanced recognition algorithms in both optical and acoustic sensors.

Given the full coverage of the area with accurate navigation and localization during the

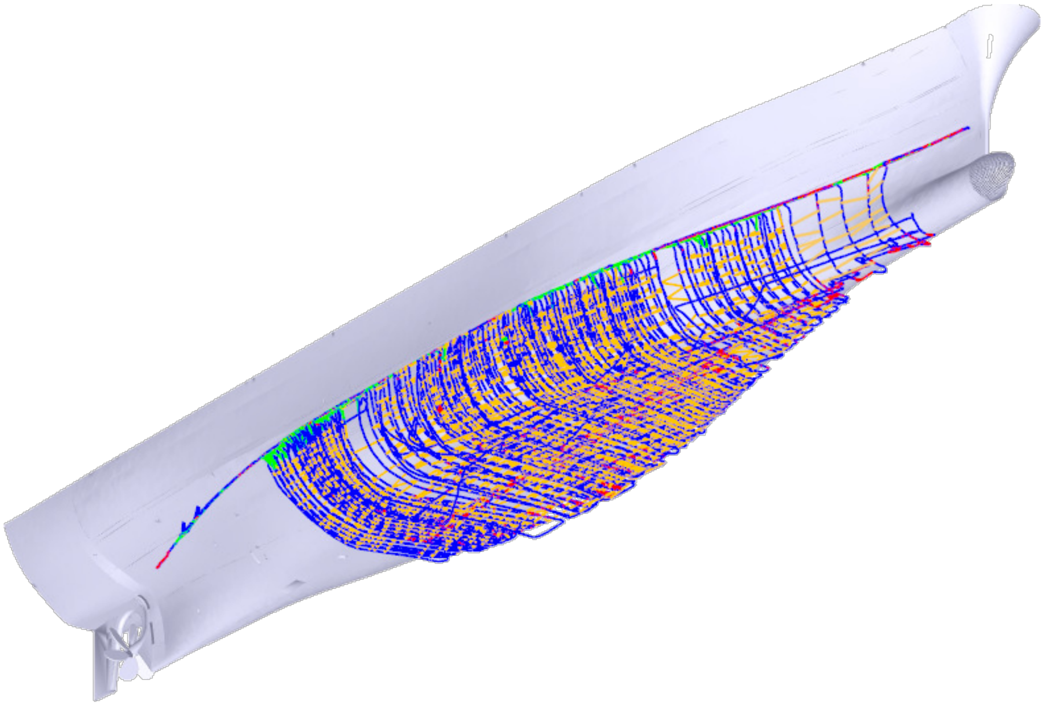


Figure 1.5: Dense full coverage of the planar area.

mission, the observation data can be used to provide high-fidelity models of the ship hull for periodic ship hull condition evaluation and foreign object detection. Fig. 1.6 gives an example of the offline reconstruction.

As depicted in Fig. 1.1, the challenges of the underwater environment still limit the robustness and application scenarios of the navigation system. The primary purpose of the proposed work is to improve robustness of the navigation system under extreme environment conditions by leveraging multiple modalities.

The method proposed in this thesis is evaluated using data collected during the multi-inspection on a Wright-class Aviation Logistics Support Container Ship, as depicted in Fig. 1.7. The total length of this vessel is $183\ \mathrm{m}$.

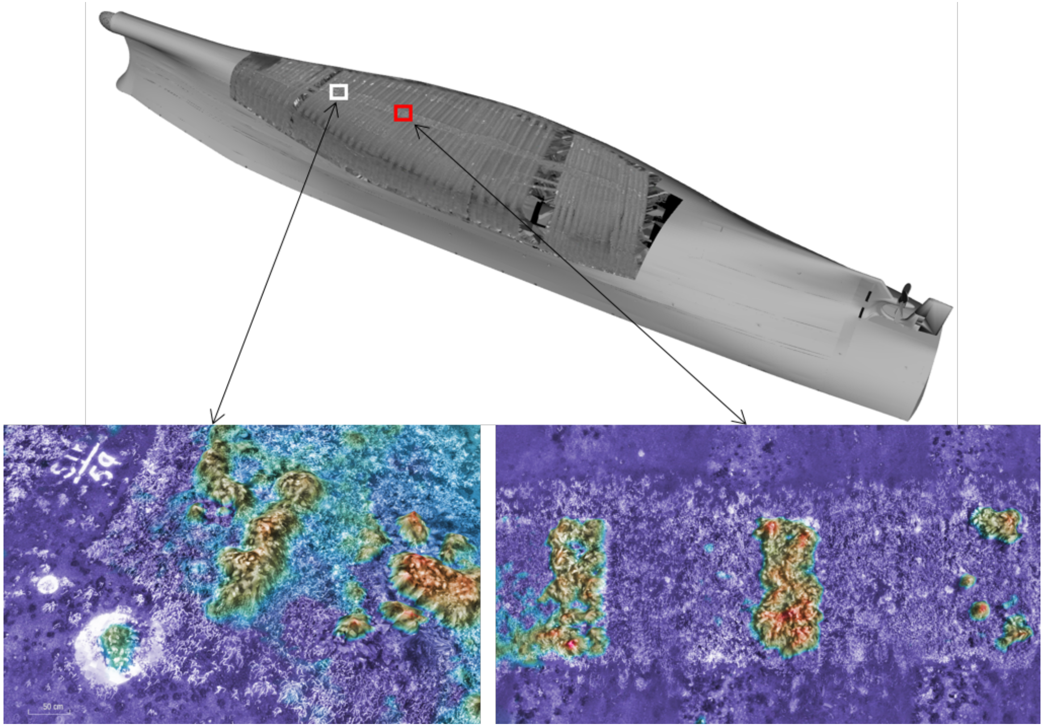


Figure 1.6: Reconstructed high-fidelity model using inspection data of the HAUV. Fine scale structure stands out and the original CAD model of the ship hull can be detected from a model-assisted bundle adjustment reconstruction framework. Figure courtesy of Ozog, Johnson-Roberson, and Eustice [85].

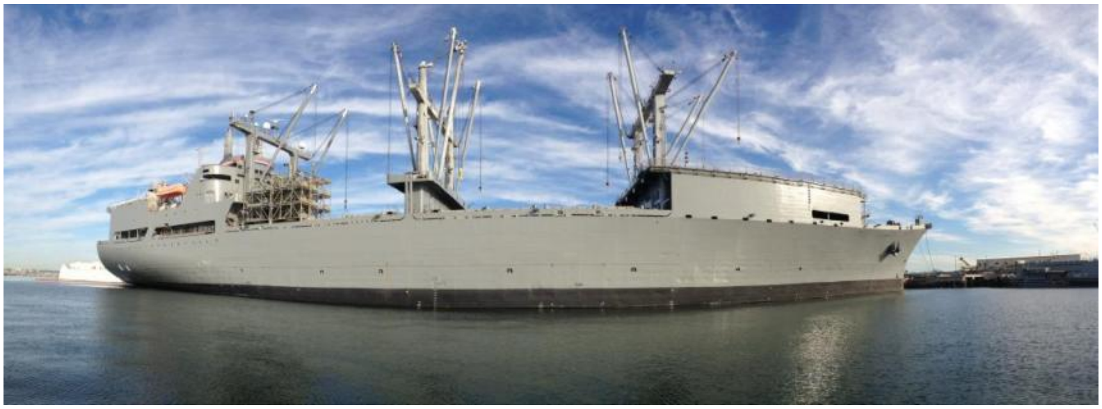


Figure 1.7: SS Curtiss.

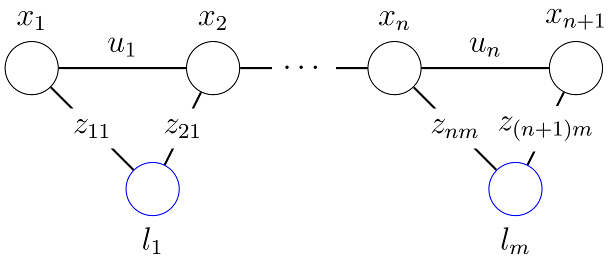


Figure 1.8: Graphical model of Full-SLAM formulation

1.2 A Review of Visual-SLAM

A simultaneous localization and mapping (SLAM) system addresses a fundamental problem in mobile robotics where a robot travels through an unknown environment and uses its sensors to collect observations of its surroundings to simultaneously estimate a map of its environment and localize itself within the map. Researchers in robotics have been studying SLAM for decades. This work has established the theory that models the noisy sensor measurements and robot state estimate jointly in a probabilistic framework [12, 29]. A SLAM system normally consists of two main components: (i) a front-end component that parses sensor measurements and formalizes them as random variables with a well defined noise model, and (ii) a back-end solver that optimizes over robot poses and map features given the measurements from the front-end.

For a Full-SLAM problem, the solver estimates the maximum *a posteriori* (MAP) robot states and landmarks in the environment:

$$\mathbf{X}^*, \mathbf{L}^* = \underset{\mathbf{X}, \mathbf{L}}{\operatorname{argmax}}\, \mathbf{p}(\mathbf{X}, \mathbf{L} | \mathbf{U}, \mathbf{Z}), \tag{1.1}$$

where $\mathbf{X}$ represents the robot states along the trajectory and $\mathbf{L}$ are landmark states observed in the environment. $\mathbf{U}$ represents control information or odometry measurements between poses depending on the actual robot platform and $\mathbf{Z}$ are other direct measurements of the environment, such as laser scans or feature point measurements from camera images. The graph model of the problem formulation is depicted in Fig. 1.8.

In some application scenarios in which real-time maps are not a target product, Pose-SLAM is often employed to decrease the optimization complexity by marginalizing the landmarks. The corresponding graph model is given in Fig. 1.9. This formulation is also very useful when sensor measurements cannot be easily modeled as a landmark, such as laser scans. In this document, we employ Pose-SLAM formulation in most of the applications.

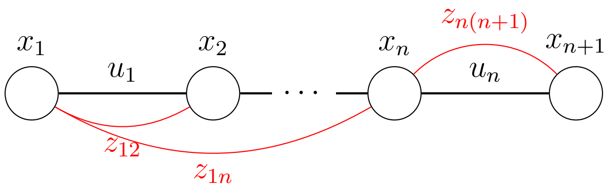


Figure 1.9: Graphical model of Pose-SLAM formulation

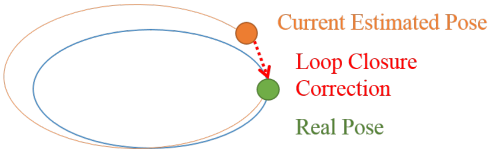


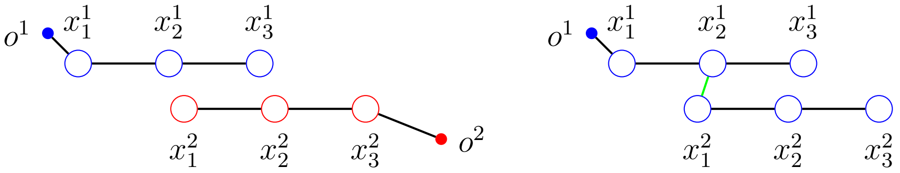
Figure 1.10: Loop-closure in SLAM. Blue line is the ground truth trajectory. Orange line is the estimated trajectory. When the robot is able to recognize a previously visited position, a loop-closure constraint can be generated and used to correct the drifting.

1.2.1 Appearance-based Place Recognition

The problem of recognizing locations based on scene appearance using perceptual sensors is of great importance for mobile robotics in the context of loop-closure detection within SLAM and global localization across multiple SLAM sessions.

Loop closing is the task of deciding whether or not a vehicle has, after an excursion of arbitrary length, returned to a previously visited area, as depicted in Fig. 1.10. A successful detection of loop closure could help correct the drift error in robot pose estimation accumulated along the trajectory and decrease the estimation uncertainty. However, loop closure detection is very difficult because the uncertainty in the robot trajectory increases overtime and results in a large range of possible locations to be tested.

Place recognition in a global localization context poses a more challenging problem



(a) Two SLAM sessions are independent without any prior information or constraint on their relative positions.

(b) Two SLAM sessions can be merged into one with an alignment constraint induced by place recognition (green line).

Figure 1.11: Place recognition for multi-session SLAM.

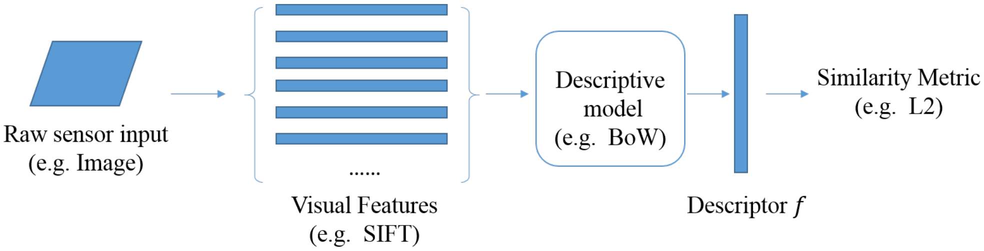


Figure 1.12: General flowchart of appearance-based place recognition algorithms.

where no prior information is available. Typical application scenarios include multi-session SLAM, when multiple robots cooperatively explore a common area or a single robot surveys an area with multiple deployments. An illustration of global place recognition is given in Fig. 1.11.

Appearance-based information is one of the most commonly used cues for place recognition of mobile robot platforms. As the appearance of a scene is independent of the pose and error estimate, these methods can potentially provide a solution which allows loop closures to be quickly and robustly detected. Sensors commonly used in this work include monocular and stereo camera systems and imaging sonar in some underwater vehicles. In general, approaches addressing the place recognition problem describe the scene using descriptive models computed over visual features and judge the scene similarity based on metrics that suit the scene descriptor. We give a general flowchart of key steps in place recognition in Fig. 1.12. One of the most representative algorithms is the Fast Appearance-based Mapping algorithm (FAB-MAP) [24], which describes the environment through a bag-of-words (BoW) model built upon scale-invariant feature transform (SIFT) features. Other approaches can vary in the type of visual features extracted or in the descriptive models used to represent the features in a concise manner. Commonly-used visual features include SIFT, speeded up robust features (SURF), edges, and histogram of oriented gradients (HOG). More recent work has explored the potential of using mid-level features from a pre-trained convolutional neural network (CNN) as visual features in the system [22].

On top of the scene modeling, systematic methods that combine multiple images or pieces of scenes for place recognition are also conducted to provide more robust detection. SeqSLAM [75], proposed by Milford and Wyeth, estimates the topological location of the vehicle by matching two segments of the robot's trajectory instead of matching two individual images. This approach is based on the fact that ground vehicles or self-driving cars will repeat the trajectory at the same location while following the route. Naseer et al. [77] proposed a framework that builds upon SeqSLAM, which used the Flow Network technique that relaxes the shared-route assumption to a partial shared-route assumption.

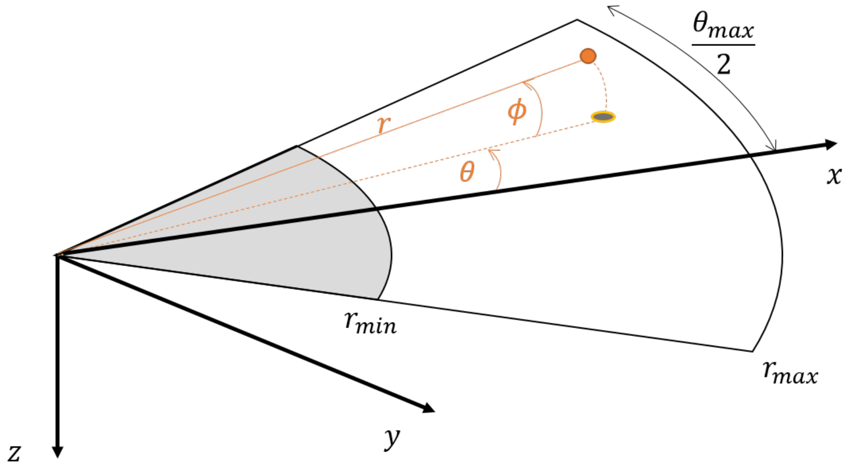


Figure 1.13: FLS Sensor Measurement.

1.3 A Review of Forward-looking Sonar (FLS) in Underwater Robotics

In this section, we review the use of FLS in underwater robotics. We first describe the basic operation principles and image formation of FLS devices. Then we provide a discussion on the challenges of using FLS as a perceptual modality. Finally, we review the applications of FLS for underwater navigation and scene reconstruction.

1.3.1 FLS

FLS, also referred to as forward-scan sonar or an acoustic camera, is a new generation of sonar that provides high-definition acoustic imagery at a fast rate. FLS sends out acoustic waves spanning the field-of-view (FOV), as depicted in Fig. 1.14. Then an array of transducers observes the reflected signals from the scene as returns in the form of range and bearing (r, θ). A FLS is not able to disambiguate the elevation angle of the acoustic return originating at a particular range and bearing. Thus, a measurement point (r, θ) can originate anywhere from an arc defined by (r, θ) in the Spherical coordinate system centered at the projection center of the sonar head, as depicted in Fig. 1.13. This representation is then converted to the final 2D image in Cartesian coordinates for an easier interpretation. Resulting image examples are shown in Fig. 1.1.

Based on this operation principle, different types of FLS vary in operating characteristics. A review of FLS system design and feature comparison is given by Loggins [70].

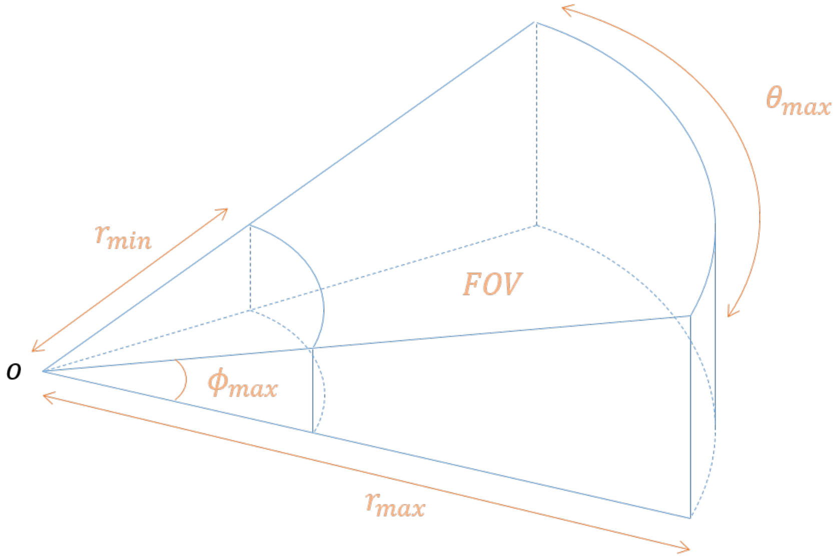


Figure 1.14: An illustration of FLS Field of View.

1.3.2 Challenges in FLS processing

As discussed in Section 1.1, FLS provides more stable signals compared to optical sensors in turbid water. However, challenges exist for this type of imagery. We provide a discussion of the significant challenges that impact most FLS applications:

1. Low Signal-Noise-Ratio (SNR): The FLS images suffer from a very low SNR. The main noise source for the FLS images is the speckle noise induced by the mutual interference of return acoustic signals [41]. The same phenomenon also degrades imagery from other coherent imaging systems, such as synthetic aperture radar or medical ultrasound [87].

2. Sensitive to View Point Changes: Appearance variations in imagery due to a change in the sonar's viewpoint are inherent in the image formation process and more significant compared to optical sensors. Imaging the scene or object from two different view points can dramatically change the visual appearance. Unlike optical images, where corner points and edges can be preserved if the scene appearance is consistent, low level features in sonar images are jointly affected by scene structure and relative position to the sonar head. Thus, powerful hand-crafted features developed for optical images will lose their feature invariant properties and may perform poorly on sonar frames.

3. Low Resolution: Although FLS is considered high resolution among sonars, as a perceptual sensor, the image resolution is still low compared to optical camera images.

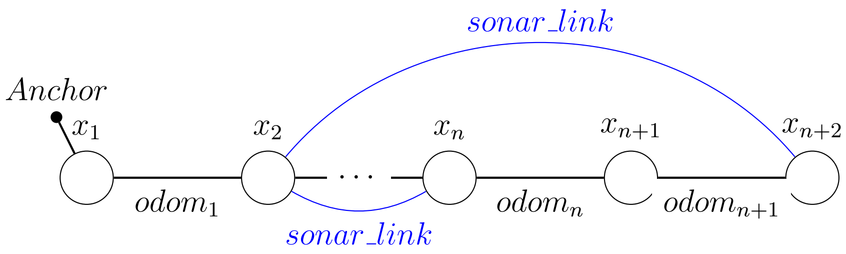


Figure 1.15: Depiction of FLS-aided pose-graph SLAM system

The angular resolution is limited by the number of transducers physically held by the device. In addition, the resolution with respect to the physical space will further decrease as the range increases.

1.3.3 Imaging Sonar in Navigation Systems

The geometry system of FLS imaging described in the last section defines the relationship between the 6 degree-of-freedom (DOF) position of the sonar and the 3D structure. Given sufficient data association on image features, constraints on sonar motions across different frames can be recovered. Pose-graph SLAM is a popular way to include these measurements in the vehicle state estimation for underwater navigation systems. As depicted in Fig. 1.15, relative-motion constraints derived from sonar data association are added to the SLAM graph that only estimates vehicle poses without explicitly modeling landmarks in the environment [45, 53, 96].

However, 6 DOF relative motion estimation from feature point correspondents in sonar frames formulates an under-constraint estimation problem to deal with the loss of information in the dimension of elevation in sonar measurements. Additional assumptions are often made, adding extra constraints to the problem. A representative assumption is the planar assumption of the observation scene, which is feasible to some extent in the survey of large open areas of underwater structures or sea floor. Johannsson et al. [53] make an orthographic projection approximation on the sonar geometry based on the planar scene assumption with the fact of narrow elevation in a ship hull inspection application. The actual approximation made within the image frame is depicted in Fig. 1.16. Under this assumption, the ambiguity in elevation is transferred to the ambiguity in axis z. A 2D motion of (x, y, θ) can be recovered from a fully constrained estimation problem following the equation below:

$$\hat{p}_j = H_{ij} \cdot \hat{p}_i = \begin{bmatrix} \cos\theta_{ij} & -\sin\theta_{ij} & x_{ij} \\ \sin\theta_{ij} & \cos\theta_{ij} & y_{ij} \end{bmatrix} \cdot \hat{p}_i, \tag{1.2}$$

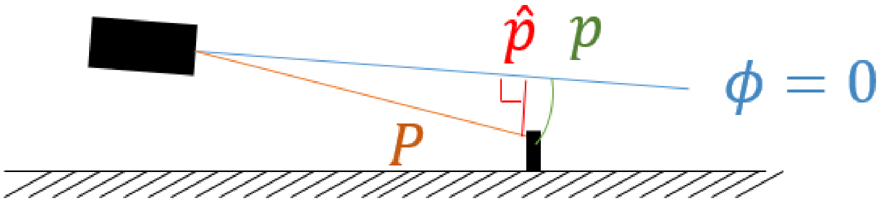


Figure 1.16: Illustration of the approximation in orthogonal projection.

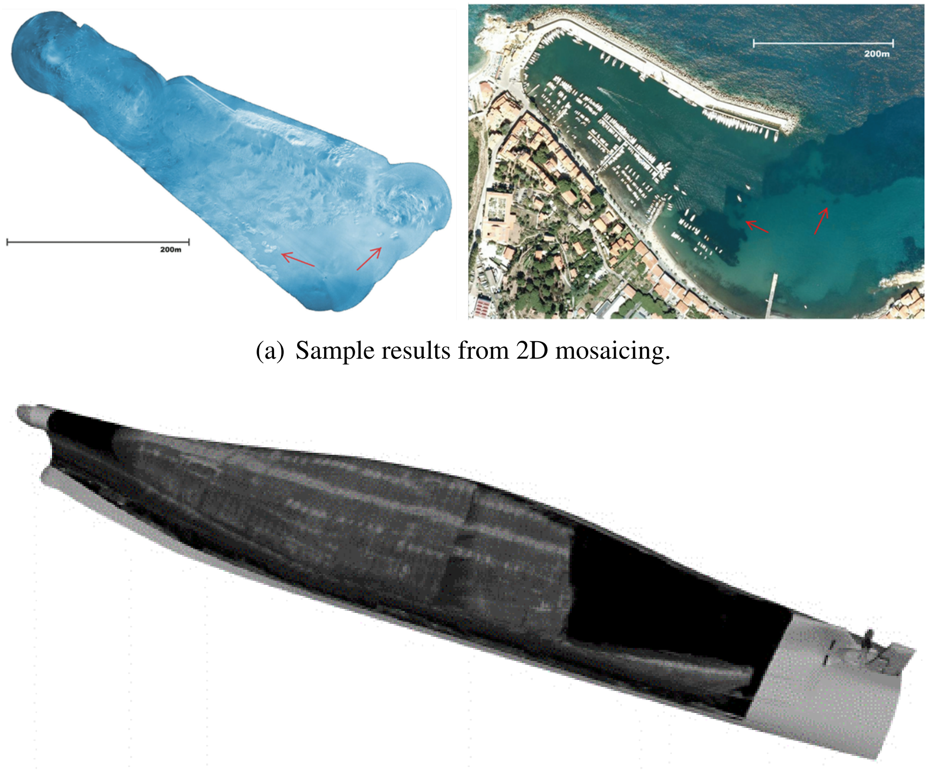
where $\hat{p}_j$ and $\hat{p}_i$ are corresponding feature points in sonar frame j and i under orthogonal projection approximation, and $(x_{ij}, y_{ij}, \theta_{ij})$ is motion transformation from frame i to frame j in 2D.

Other literature shares similar ideas of assumptions with different formulations of the model. Aykin and Negahdaripour relax the whole scene planar assumption in their work in two-view scan matching [9], but make the local planar surface assumption in image patches to enable the use of shadow information for feature detection and registration. Shin et al. [89] relax the assumption of narrow elevation by modeling the normal vector of the scene surface in a bundle adjustment framework.

Huang and Kaess [46] introduced Acoustic Structure From Motion (ASFM), which used multiple sonar frames from different viewpoints to reconstruct 3D structure as well as sonar motions jointly with no specific assumption to the scene. ASFM shows promise in providing sonar-based geometry estimation without further scene assumptions. However, the existing work was mainly evaluated on simulation data or ideal experimental setting. There are still great challenges towards a deployable system, which include but are not limited to robust data association and efficient key frame selection for real-time performance as well as singular motion pattern avoidance.

1.3.4 Scene Reconstruction Using FLS Imagery

In addition to recovering vehicle movement, reconstruction of the scene is another important direction of FLS application in underwater robotics. Given relatively reliable motion prior between sonar frames, image registration is conducted in a global optimization manner across the whole image. The majority of literature on FLS scene reconstruction addresses the problem by solving 2D mosaicing, where the planar assumption and orthogonal projection approximation are used, as discussed in Section 1.3.3. A typical example of resulting 2D mosaicing is given in Fig. 1.17(a). More recent work starts to look at the possibility of 3D mosaicing based on dense 3D structure prior of the scene [84], as shown in Fig. 1.17(b).



(a) Sample results from 2D mosaicing.

(b) Sample results from 3D mosaicing.

Figure 1.17: Example results from state-of-art sonar mosaicing literature. Fig. 1.17(a) depicts a large scale 2D mosacing result of seafoor area at the new Marciana Marina Harbor. Orthophotomap of the Marciana Marina environment is also given for reference. Figure courtesy Hurts et al. [49]. Fig. 1.17(b) depicts a 3D mosaicing result aligned to a ship hull CAD model. Figure courtesy Ozog et al. [84].

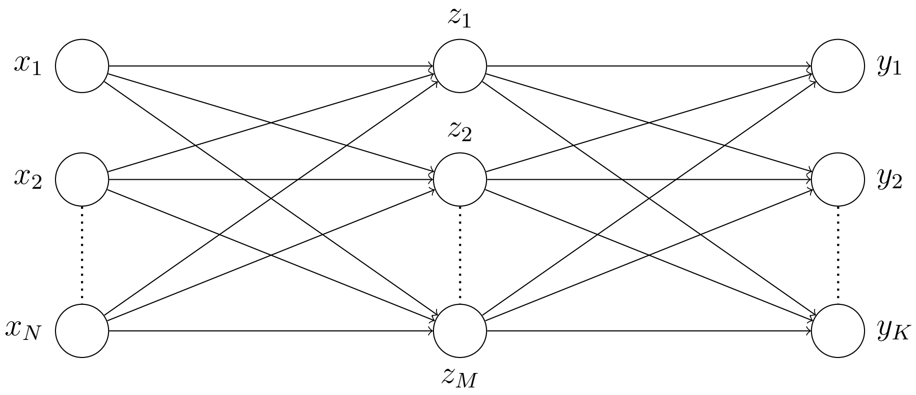


Figure 1.18: Example of basic neural network model.

1.4 A Review of Convolutional Neural Networks

In this section, we give a brief review of CNNs. We start with an introduction of the basic idea of a neural network as a regression method, then focus on convolutional neural networks which are widely used in the context of computer vision. Finally, we discuss literature leveraging CNN techniques in mobile robotics applications.

Neural networks, also known as artificial neural networks, comprise a variety of regression models where layers of parametric basis functions are stacked with parameters adapted during training. The term ‘neural network‘ has its origins in attempts to find mathematical representation of information processing in biological systems. In this document, we only consider neural networks as efficient models for statistical pattern recognition. Although neural network models in practice can be complex and large, it is easy to capture the general idea following a simple and basic network example. The system shown in Fig. 1.18 constructs M linear combination of the input variables $x_1, x_2, ..., x_N$:

$$a_j = \Sigma_{i=1}^{N} w_{ij}^{(1)} x_i + w_{j0}^{(1)},\ j \in \{1, ..., M\}. \tag{1.3}$$

The superscript of w indicates the index of 'layers' of the network. Each of these linear combinations is then transformed using a differentiable, nonlinear activation function $h_1(\,\cdot\,)$ to generate the output variables z_j of the first layers:

$$z_j = h_1(a_j). \tag{1.4}$$

These variables are often referred to as hidden units or hidden variables as they are not directly targeted outputs of the system. Hidden variables $z_1, ..., z_M$ are then passed into the second layer in a similar way.:

$$b_k = \Sigma_{j=1}^{K} w_{ik}^{(2)} z_j + w_{k0}^{(2)},\ k \in \{1, ..., K\}. \tag{1.5}$$

Similarly, each output variable is transformed using an activation function $h_2(\,\cdot\,)$.

$$y_k = h_2(b_k). \tag{1.6}$$

The activation functions used in the network vary based on the expected properties of both input and output variables. In the case of multiple binary classification problems, logistic sigmoid functions are often used to provide a differentiable activation function while keeping the output value between $[0, 1]$.

$$\sigma(b) = \frac{1}{1 + e^{-a}} \tag{1.7}$$

By substituting z_j with $\{x_i\}$, it is easy to combine the systems to see what model is actually used to model $\{y_k\}$ from $\{x_i\}$:

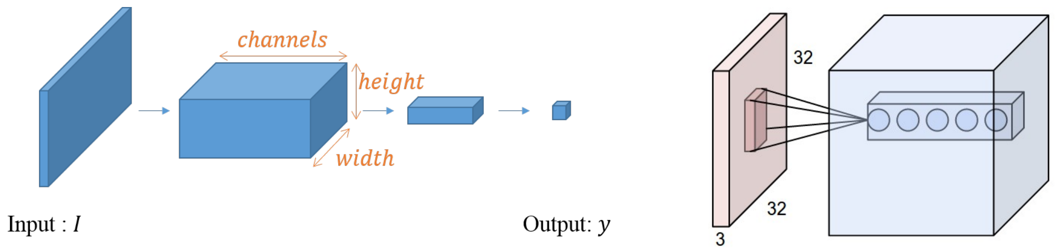
$$y_k(\mathbf{x}, \mathbf{w}) = \sigma(\Sigma_{j=1}^{M} w_{kj}^{(2)} \cdot h_1(\Sigma_{i=1}^{N} w_{ij}^{(1)} x_i + w_{j0}^{(1)}) + w_{k0}^{(2)}), \tag{1.8}$$

where all the parameters in the basic functions are grouped into a vector $\mathbf{w}$. From Eg. 1.8 we can tell this network model is a simple nonlinear function from a set of input variables $\mathbf{x}$, controlled by the parameter $\mathbf{w}$. This simple architecture given in Fig. 1.18 is the most commonly used basic component in larger networks, where more layers with variations are stacked to model the system with high complexity and nonlinearity.

1.4.1 Convolutional Neural Networks

In this document, we mostly consider the most commonly used type of neural network models in the context of computer vision and robotics: convolutional feed-forward neural networks, also referred to as CNNs. Convolutional neural networks share the basic idea of ordinary neural networks as they consist of a hierarchy structure of base functions with adaptable parameters. The main difference lies in the fact that CNNs make the explicit assumption that the inputs are images, which allows us to encode certain properties into the architecture. These then make the forward function more efficient to implement and vastly reduce the amount of parameters in the network.

As depicted in Fig. 1.19(a), variables in CNN models are often arranged in 3 dimensions: width, height and depth. This arrangement follows the assumption of an image input, which can also be generalized to other similar data types such as RGB-D images or point clouds.



(a) Simple example of data arrangement in CNN.

(b) Illustration of convolutional layer.

Figure 1.19: Illustration of basic CNN architecture. Fig. 1.19(a) depicts the data arrangement in a typical CNN architecture. Fig. 1.19(b) depicts how the convolutional layer is constructed and the effect on the output volume. Figure courtesy Stanford online class CS231n [1].

Based on this volume type of data arrangement, a CNN architecture is formed by a stack of distinct layers that transform the input volume into an output volume. The convolutional layer is the core building block of a CNN. The layer's parameters consist of a set of filters (or kernels), which have a small receptive field, but extend through the full depth of the input volume, as depicted in Fig. 1.19(b). The output volume are given by:

$$\mathbf{z}^{(i)}(u, v) = \hat{\mathbf{x}} * \mathbf{w}^{i}, \tag{1.9}$$

where superscript i denotes the index of layer, $\mathbf{w}^{i}$ refers to the kernel weight shared by the whole layer i, and $\hat{\mathbf{x}}$ refers to the corresponding receptive field of $\mathbf{z}^{(i)}(u, v)$ in the input volume. By parameter sharing within the whole layer, the network architecture is able to keep a reasonable parameter size despite the high dimensionality of variables in each layer.

CNNs have been applied extensively to computer vision problems for quite a long time. Prior work on the CNN architecture can be traced back to [63] by Lecun et al. on handwritten digit recognition. More recently, with the performance increase in graphical processing units (GPUs), more sophisticated network structures have been proposed and demonstrated with great success in many traditional computer vision tasks, such as object recognition and image classification. Best performance on benchmark datasets has been continually improved by CNN-based methods, including ImageNet [27] and MNIST [62]. Representative networks include AlexNet [61] and GoogLeNet [92]. More recently, well-developed open source libraries, such as Caffe [51], Theano [15] and TensorFlow [2], also help to generalize CNNs into powerful and adaptable tools in many different applications.

1.4.2 Application in Mobile Robotics

The interest in CNNs is not limited to the field of computer vision, but encompasses almost all AI-related areas, including natural language processing (NLP) [23] and information retrieval [98].

Mobile robotics is definitely one of the areas in which CNNs can achieve great success. The utilization of CNNs in mobile robotic applications can be roughly categorized into three main directions: i) Semantic understanding of the scene that provides information for motion planners; ii) Non-semantic information extraction for robot navigation systems; iii) End-to-end model learning that provides direct control output from input image sequences.

The majority of current CNN utilization in mobile robotics lies in the application of real-time semantic understanding of the scene from perceptual modalities including camera and depth sensors. Extracting semantic information from the scene is intrinsically very similar to some of the classic computer vision tasks such as object detection and recognition and scene segmentation. A lot of architectures have been adapted to solve these types of problems specifically for mobile robotics. Hadsell et al. [37] leverages a CNN model to predict traversable areas from videos using unsupervised data. The KITTI dataset [35] provides sufficient sensor data collected by autonomous driving platforms in urban areas, which has enabled a set of state-of-the-art semantic understanding algorithms to be developed for self-driving vehicles [11, 44, 101, 105]. Johnson-Roberson et al. [54] propose to use photo-realistic computer images from a simulation engine to rapidly generate annotated data with variation in illumination and weather in the application of urban scene understanding for self-driving vehicles.

Non-semantic understanding of perceptual sensor measurements using CNNs focuses on metrical or topological understandings of the scene or vehicle motion that provides information for improving navigation accuracy. Kendall, Grimes, and Cipolla propose to train CNNs that directly estimate 6 DOF camera position from input images. Luo, Schwing, and Urtasun [73] developed a stereo image matching algorithm using CNNs by modeling the disparity estimation as a classification problem. Chen et al. [22] propose the first place recognition based on the mid-layer features of a pre-trained model.

In more recent literature, end-to-end model learning has attracted a lot of attention with a brave attempt to leverage the CNN architecture as a complete system that provides direct output to some higher level and more complex tasks such as control and planning. Bojarski et al. [17] propose to use CNNs to provide steering wheel angles in real-time to allow for lane keeping from input stereo image sequences. Although the algorithm has only been evaluated on simulation tasks and limited on-road tests, the promising result empirically demonstrates that CNNs are able to learn the entire task of lane and road following without manual decomposition into road or lane marking detection, semantic abstraction, or path planning and control.

1.5 Thesis Outline

1.5.1 Thesis Goals

The problems we consider in this thesis are as follows:

i Registration among optical observation of the scene across time is a key pre-requisite for underwater navigation systems. Changes in illumination, water condition and physical appearance in the underwater environment make this task extremely challenging. In this thesis, we explore registration and comparison between optical observations against dramatic changes happening in underwater environments across long periods, given periodic observation measurement of an underwater scene or structure.

ii The capability of optical sensors in underwater navigation systems is limited by their signal sensitivity to water conditions and limited FOV. In this thesis, we improve perception and navigation capabilities of an underwater autonomous vehicle by incorporating FLS imaging measurement at different scales. As an extra perceptual modality to the environment alongside optical cameras, imaging sonar will improve system robustness towards turbid environment and increase the perceptual footprint of the vehicle with larger FOV.

To address the problem above, we present the following contributions to the development of a multi-modality SLAM system for underwater autonomous vehicles:

1 A high-level feature detection and description approach that enables a robust registration between optical images despite dramatic appearance changes. We show that the proposed feature combined with sample-based localization frameworks could achieve promising performance against low feature density and appearance changes.

2 An image-level descriptive feature leveraging CNN techniques for forward-scan imaging sonar. The learned features provide a robust and efficient similarity measurement in high-noise-level and weak-feature-texture images. The place recognition performance using these features achieves promising accuracy compared to that achieved using traditional hand-crafted features.

3 A general pre-processing procedure to utilize high-dimensional features in real-time Bayesian filtering frameworks. We reduce the curse of dimensionality of observations in recursive Bayesian filtering and provide noise modeling to black-box features using data perturbation techniques. We show that the proposed approach can be used in a real-time localization framework using the proposed sonar image features.

4 An imaging sonar SLAM front-end for a pose-graph SLAM system which provides vehicle pose constraints from local bundle adjustment among a clique of neighboring sonar images. We provide robust solutions to practical problems including sparse feature distribution and unreliable data association to guarantee a deployable approach for real-world systems. We show that the proposed method improves the vehicle state estimation to the SLAM system by providing robust navigation information through acoustic signals.

The work introduced in this thesis has been published in the following venues:

J. Li, R. M. Eustice, and M. Johnson-Roberson. High-level visual features for underwater place recognition. In *Proc. IEEE Int. Conf. Robot. and Automation*, pages 3652–3659, May 2015. doi: 10.1109/ICRA.2015.7139706

J. Li, R. M. Eustice, and M. Johnson-Roberson. Underwater robot visual place recognition in the presence of dramatic appearance change. In *Proc. IEEE/MTS OCEANS Conf. Exhib.*, pages 1–6, Oct 2015. doi: 10.23919/OCEANS.2015.7404369

J. Li, P. Ozog, J. Abernethy, R. M. Eustice, and M. Johnson-Roberson. Utilizing high-dimensional features for real-time robotic applications: Reducing the curse of dimensionality for recursive bayesian estimation. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, pages 1230–1237, Oct 2016. doi: 10.1109/IROS.2016.7759205

J. Li, M. Kaess, R. M. Eustice, and M. Johnson-Roberson. Forward-looking sonar pose-graph SLAM. In *Proc. Int. Symp. Robot. Res.*, 2017. Submitted

1.5.2 Document Road-map

Each contribution discussed above is described in detail in the following chapters:

Chapter 2 describes a high-level feature detector and corresponding description approach of optical images, which enables image registration against strong appearance changes.

Chapter 3 describes an image-level descriptive feature of FLS images extracted by CNNs.

Chapter 4 describes a general approach to utilize abstracted high-dimensional features in Bayesian filtering frameworks. An evaluation of the efficiency of this method on the sonar image features proposed in Chapter 3 is provided.

Chapter 5 describes a geometry estimation algorithm from a neighboring clique of sonar frames, which can serve as the front-end of a pose-graph SLAM system.

Chapter 6 offers a summary of our contributions described in this document and provides a discussion on potential directions for future work.

CHAPTER 2

High-level Feature Place Recognition

2.1 Introduction

Recognizing a place that has been previously viewed is an important challenge in both robotic navigation and computer vision. It is a prerequisite for visual-based navigation systems, and also important for long-term robot operation using perception. Several robust approaches have been proposed in the last decade for terrestrial place recognition. However, place recognition in the underwater environment is a more challenging problem. Dramatic changes in scene appearance occur due to viewpoint dependent illumination (underwater robots often carry their own light source) or biofouling, both of which can cause poor performance when executing point-based feature matching. The low density of salient visual features is another factor that also increases the difficulty of underwater image matching.

In this chapter, we introduce our work addressing the problem of underwater visual place recognition across years by leveraging high-level information that persists longer in the underwater environment across time, such as structure or shape. Built upon this idea, we provide answers to three main questions embedded in the problem: feature detection (or feature proposal), feature description and feature-based image matching.

The proposed algorithm first conducts feature patch region proposal using image segmentation techniques. A classifier trained on-line is then used to provide similarity metrics for each feature region based on high-level information. Finally, a geometric model selected from all the feature patch matching results provides a reliable image-level similarity measurement. Additionally, since individual image matching can sometimes be unreliable due to insufficient features, neighboring images are also considered in the proposed method to provide more visual evidence to improve robustness.

The proposed solution has applications for temporally periodic underwater structure inspection and long term monitoring of benthic marine habitats. We evaluate the proposed method on a labeled dataset from a periodic ship hull inspection application. To provide a realistic example of application of our algorithm, we also embed it into an application scenario of multi-session graph merging for underwater long-term navigation. The experimental results indicate that the proposed method outperforms most of the classic features dealing with dramatic scenes changes and is able to provide robust place recognition results for graph merging across years.

This chapter represents the culmination of several of our publications [65, 66]. In the sections that follow, we discuss several contributions that arose from our work:

- A visually-salient patch-based feature proposal method that enables us to localize the salient regions in underwater images;
- A patch-feature description that is robust to dramatic changes in appearance;
- A robust outlier rejection for patch matching using a geometric constraint; and
- An online relocalization framework for multi-SLAM session registration against dramatic environment changes.

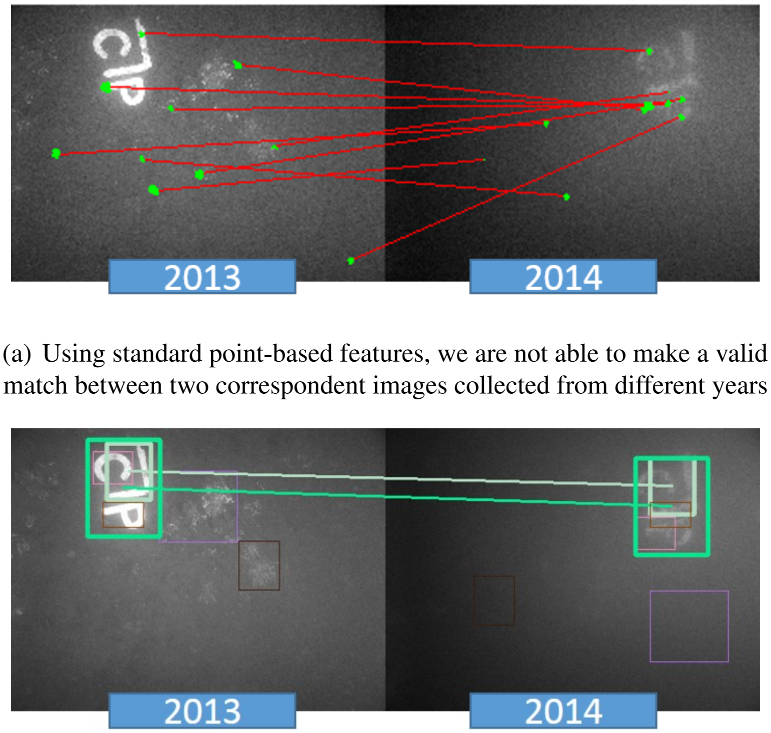
While the proposed approach can be generalized to many different applications, we evaluate the proposed approach in the relocalization task of underwater ship hull inspection across years.

2.1.1 Outline

The rest of this chapter is arranged as follows: (i) Section 2.2 will give a brief introduction about related works of visual-based navigation and visual feature matching; (ii) Section 2.3 presents the key steps in the algorithm of high-level feature matching; (iii) Section 2.4 introduces how the proposed high-level visual feature matching approach can be adapted into multi-session merging tasks for long-term SLAM; (iv) Section 2.5 evaluates the approach through the use of ship hull inspection data and compares the approach with some state-of-art algorithms; and (v) Section 2.6 concludes the work of this chapter.

2.2 Background

In this section, we cover two broad areas of related work: (i) visual features used in the context of place recognition; and (ii) visual-based navigation systems.



(a) Using standard point-based features, we are not able to make a valid match between two correspondent images collected from different years

(b) Using the proposed method, we are able to match two correspondent images collected from different years

Figure 2.1: Depiction of the proposed high-level feature matching method with real data. The two images are corresponding images collected from the years 2013 and 2014. Obvious texture-fading can be seen, which limits the performance of point-based features. The proposed method, on the other hand, is able to select a salient patch that contains high-level information that tends to last longer, resulting in higher robustness against appearance changes.

2.2.1 Visual feature matching

Visual feature matching is the key prerequisite of visual-based place recognition. For decades, fundamental work has been done to select signature features and describe them in a unique way for image matching or image classification. An important domain has been point-based features that provide feature correspondences invariant to view point changes, such as Scale-Invariant Feature Transform (SIFT) [71], Speeded Up Robust Features (SURF) [13] and Binary Robust Independent Elementary Features (BRIEF) [18]. SIFT is proposed by Lowe for object recognition [71]. He proposed to detect key points from maxima or minima of Difference of Gaussian (DoG) filters, and to describe the feature point with an oriented and normalized descriptor based on the surrounding image patch of the key point. Currently, SIFT is one of the most popular features used in object recognition and classification due to its robustness to scale and local affine distortions. However, these features are mainly detected and described based on the gradients of a small neighboring patch in the image. When illumination conditions change drastically, or any decay occurs to the scene, details of image appearance will change, resulting in large differences in small patches. Additionally, extracting a sufficient number of features is important to guarantee the matching performance, which is hard to achieve in the underwater environment.

Higher level features are also used elsewhere in the computer vision community. Such features capture more structural information and are more tolerant to changes or differences in appearance in small areas of the image. Representative works include GIST [81] designed for building recognition, and HoG [25] that is mainly used for specific object type recognition. GIST proposed to interpret an image by its principle components in spectrograms. Natural images can be easily distinguished from images of man-made objects since they often contain different components in the frequency domain. HoG describes a bounding box with a vector containing the histogram of gradients for different parts of the bounding box. It captures the shape information of a image patch. However, these features are seldom used in place recognition frameworks, since a large amount of labeled training data is needed for each single inquiry object or bounding box. Additionally, they are not distinct enough to provide registration between two areas or two images. The output of these approaches is often a set of candidate matches or a set of positive responses in object recognition.

Recent work, inspired by the success of Artificial Neural Networks, proposes to go beyond handcrafted features and explore new possibilities by treating feature extraction and representation as a machine learning problem. Hinton et al. [42] make a breakthrough improvement in the MNIST digit image classification problem using deep belief networks. A similar idea is used by Krizhevsky et al. [61], who broke the record on ImageNet, a natural image classification challenge dataset. Subsequently, Sermanet et al. [88] proposed to use the high level features trained from ImageNet to solve more pattern recognition problems in computer vision, such as object detection and object classification. Current competition records for many prominent classification datasets are held by learning-based methods. Feature learning frameworks are able to learn complex and high level representations of a patch or the whole image, which enables them to distinguish one type of image from another. However, since the features learned from the network are high dimensional, a huge amount of training data is needed to guaranteed a robust and general output.

2.2.2 Visual-based navigation

Most current mobile robotic navigation systems are formulated as a Simultaneous Localization and Mapping (SLAM) problem, in which a robot is tasked with navigating through an a priori unknown environment using a combination of odometry and perceptual sensors. Refer to [29] for a comprehensive survey of solutions to SLAM problem. Cameras have commonly been used as the primary sensor modality for mobile robots, and the ability for visual place recognition has been improved dramatically in recent years. Representative works in visual-based place recognition include Fast Appearance-based Mapping algorithms (FAB-MAP) [24], MonoSLAM [26], and FrameSLAM [60]. However, these benchmark systems are focused more on searching for a location with the largest probability given matching results from standard point-based features. Less work has been done on operation under dramatic appearance changes of the scene.

More recently, some representative work that focuses on the dramatic scene changes in place recognition problems has been developed. SeqSLAM [75], proposed by Milford et al., estimates the topological location of the vehicle by matching two segments of the robot's trajectory instead of matching two individual images. The robustness of the system is improved when neighboring images are involved in the final matching results. However, a strict assumption is made that the two matching segments share the same route. This assumption holds for ground robots like autonomous cars, which drive on the road. For underwater robots, on the other hand, it's almost impossible for two individual segments of the path to share a same trajectory, since the vehicle can move freely in space. Naseer et al. [77] proposed a framework that builds upon SeqSLAM, which used a Flow Network technique that loosens the shared-route assumption to allow for partial matching. Additionally, they also propose to use grid HOG as the descriptor of an image, and match two images based on the dot product of two HOG vectors. Though the HOG representation dramatically improves the performance of image matching in the dataset taken over multiple seasons, it relies on the orientation of the vehicle relative to the scene, which is almost static for autonomous cars. McManus and his colleagues also explore the possibility of matching through high level features, which is similar to our method. They proposed to train Support Vector Machine (SVM) classifiers of signature HOG patch features for a set of neighboring images using an unsupervised method. However, the restriction of small or no pose change relative to the scene limits their approach. Their unsupervised signature patch searching depends on the fact that the image is feature-rich. Neither of these two basic assumptions hold in the underwater environment.

Some effort has gone into addressing the image registration problem for underwater images. Eustice et al. propose to use pose-constraints from SLAM graphs to pre-constrain the searching area of point-based feature matching within one SLAM graph [32]. Carlevaris-Bianco et al. propose to do place recognition by matching a set of neighboring images which provide a larger field of view in matching and more visual features [19]. Both approaches assume SLAM prior is available to provide a constraint in feature matching. However, no prior information is available between images from two independent SLAM missions. Ozog et al. propose a registration between different SLAM graphs that selects candidate images before visual matching by comparing the vehicle poses with respect to the ship's hull using planar features estimated from data of the Doppler Velocity Log(DVL) [82]. However, this work focuses on data collected within a small time interval–a week at most.

2.3 High-level feature matching

2.3.1 System Overview

An outline of the proposed image matching method with high-level features is shown in Fig. 2.2. Given a current image in the SLAM graph of a new mission, the goal of our system is to find the image taken at the nearest location from a set of images collected in previous missions. Given a current image, a set of neighboring images is collected and salient patch features are proposed in this image set. For each patch, a binary Support Vector Machine (SVM) classifier is trained to detect similar patches in the dataset. Finally, all possible matching features are fed into a geometry verification method to reject outliers and achieve a robust matching result.

2.3.2 Salient Patch Proposal

As shown in Fig. 2.1, most images in the underwater environment, especially those involved in ship hull inspection, are of low feature density. To address this problem, we propose the

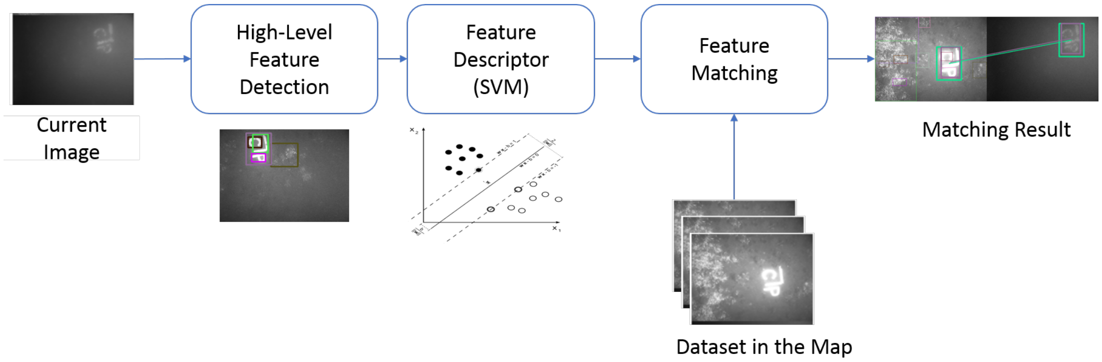


Figure 2.2: Flowchart of our high-level feature image matching approach. First, a neighboring image set is gathered for a current image. A set of patch features is detected and described with a set of SVM classifiers based on HOG feature. Patch feature matching is then carried out on the dataset followed by a geometric validation method to reject outliers.

use of segmentation techniques to select visually salient regions for feature matching, and make use of neighboring image sets to provide more visual evidence to achieve a robust matching result.

2.3.2.1 Single Image Patch Proposal

Based on human perception principles, two main characteristics are shared by visually salient regions: i. obvious and complete boundaries; and ii. unique color compared to the surrounding area. These principles are widely used in most state-of-the-art salient object extraction algorithms such as [52] [5].

Based upon these characteristics, a salient patch selection algorithm is proposed as outlined in Algorithm 1. First, a graph-based segmentation method [33] is used to decompose the image into a set of components with strong boundaries between them. The approach makes use of a graph-based representation of an image: $G = (V, E)$, where each pixel is a vertex, and each vertex is connected to its 8-neighbors. The edges are undirected and come with a weight according to pixel intensity similarity $w(e)$. The segmentation is initialized as a set of single pixel components $S_0 = (C_i, ..., C_N)$. Then, neighboring components are merged by comparing the minimum weight connecting them and a measure of internal consistency which is defined as:

$$\text{Int}(C_i) = \max_{e \in MST(C_i,E)} w(e)$$

$MST(C_i, E)$ is the *minimum span tree* of C_i constructed upon the edge set E. A minimum

Algorithm 1 Salient Patch Proposal

```
**Initialization:** a graph of the image $G = (V, E)$; Initial segments $S^0 = (C_1^0, ..., C_N^0)$. $\mathrm{Int}(C_i^0) = \infty$.
**return** Final segments $S^* = (C_1, ..., C_r)$
Sort $E$ into $E = (e_1, ..., e_M)$ by non-decreasing edge weight.
**for** $i = 1$ to $M$ **do**
    $S^i = S^i - 1$
    Consider $e_i = (v_j, v_k)$ with weight $w(e_i)$, $v_j \in C_{v_j}^i, v_k \in C_{v_k}^i$
    **if** $C_{v_j}^i != C_{v_k}^i$ **and** $w(e_i) < \mathrm{MInt}(C_{v_j}^i, C_{v_k}^i)$ **then**
        merge $C_{v_j}^i$ and $C_{v_k}^i$.
    **end if**
**end for**
$S^* = S^M$
**for** $i = 1$ to $|S^*|$ **do**
    $\mathrm{Sal}(C_i^*) = \Sigma_{B_j \in B} |\, \mathrm{Mean}(C_i^*) - \mathrm{Mean}(B_j / C_i^*)|)$
    **if** $\mathrm{Sal}(C_i^*) < \beta$ **then**
        remove $C_i^*$
    **end if**
**end for**
```

internal difference is also defined between neighboring components as a merging threshold.

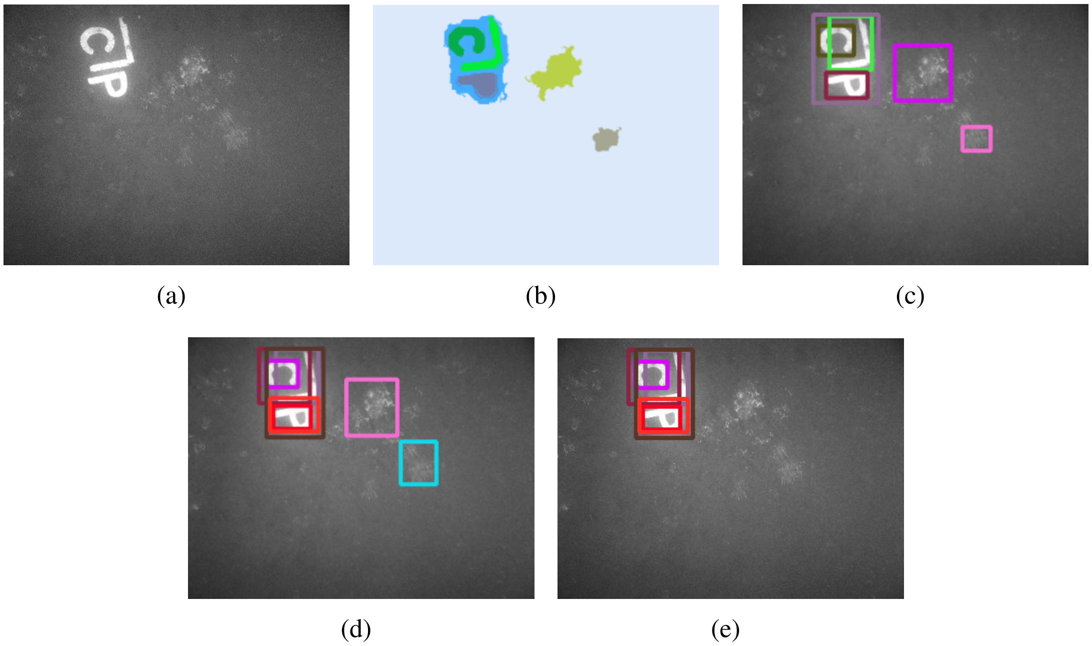
$$\mathrm{MInt}(C_1, C_2) = \min\left(\mathrm{Int}(C_1) + \tau(C_1), \mathrm{Int}(C_2) + \tau(C_2)\right)$$

The threshold function τ controls the tolerance level, which is a function of component size. $\tau(C_i) = k/|C_i|$, where k is a manually selected parameter. For a smaller component, merging is encouraged, while for larger components it becomes more difficult. The threshold function also ensures that a larger weight is needed to indicate the existence of a true boundary.

After the segmentation stage, a saliency evaluation is carried out on each component, resulting in candidate patches. The saliency score is based on local contrast and is defined as:

$$\mathrm{Sal}(C_j) = \Sigma_{B_i \in B} |\, \mathrm{Mean}(C_j) - \mathrm{Mean}(B_i / C_j)|$$

where $\mathrm{Mean}(C_j)$ is the mean pixel value in component C_j, and B is a set of bounding boxes containing C_j in different scales. $\mathrm{Mean}(B_i / C_j)$ refers to the mean value of pixels in B_i but not belonging to C_j. Under this definition, a component is compared to its surrounding area at different scales to determine its contrast level. A similar saliency score is defined in some recent salient region/object detection algorithms [3, 39, 86]. The patches are dropped if the saliency score is lower than a manually set threshold β.



(a) (b) (c)

(d) (e)

Figure 2.3: The intermediate steps in salient patch proposal where a pool of candidate patches are extracted. (a) Original current image. (b) Segmentation results of the current image. (c) Patch proposal of single current image without saliency judgment. (d) Patch pool from all neighboring images projected onto current image frame. (e) Selected salient patches after saliency judgment.

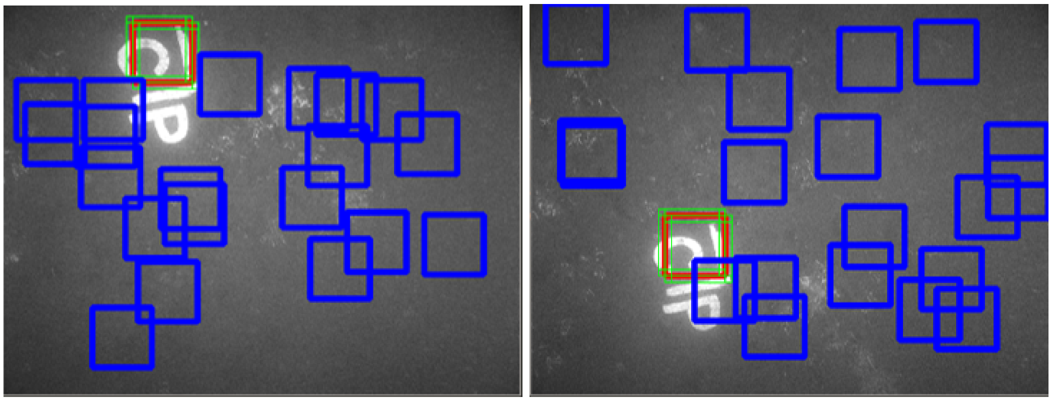


Figure 2.4: Training samples are generated in current image and the neighboring images. Red: positive samples; Green: positive samples generated by shifting; Blue: negative samples

2.3.3 Patch Pooling in Neighboring Image Set

Given an current image I, a set of neighboring images R, including I, is considered. Salient patch proposal is performed on each image in R. All of the proposed patches are included in a patch pool P and considered in the next stage. As shown in Fig. 2.3(d), the neighboring image helps to propose more combinations of patches. For this step, SIFT matching is used since the neighboring images are collected during the same mission.

2.3.3.1 SVM Classifier for Patch Features

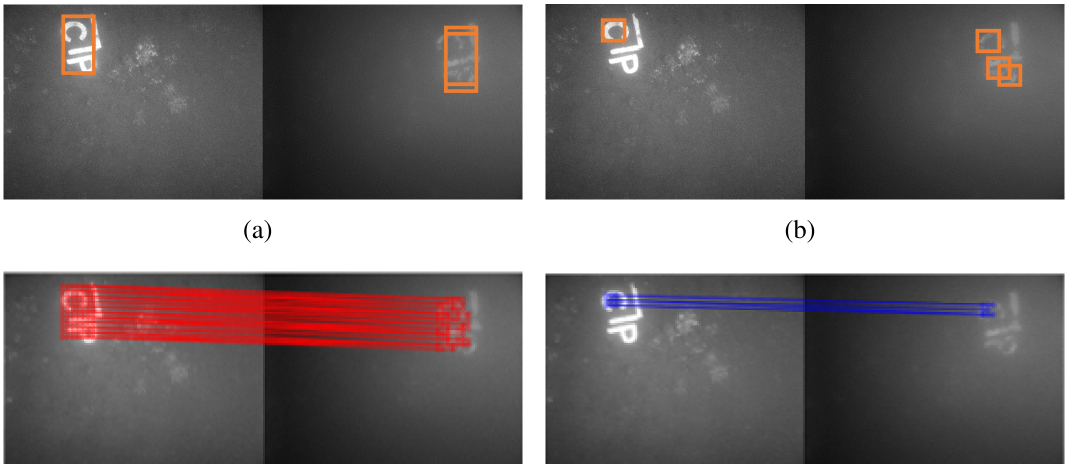
Given the location and scale of a salient patch, a description of the patch and a distance metric is used for patch feature matching. A binary classifier is trained to discriminate matches.

As shown in Fig. 2.4, HOG features [25] are used to provide a description for the patch, capturing the outline and shape information of a patch. A linear SVM classifier is then trained to detect corresponding patches in a current image. The whole procedure can be summarized in the following steps:

1. For each patch feature p_i, search for corresponding patches in the neighboring images using the propagation method mentioned in Section 2.3.2. Extract HOG features for all these correspondent patches as positive samples for SVM training.

2. In every neighboring image, generate motion compensated positive samples by slightly shifting the corresponding patches, as shown in the green boxes in Fig. 2.4.

3. In every neighboring image, randomly sample N patches that are off the location of



(a)

(b)

(c)

(d)

Figure 2.5: This figure illustrates how the fundamental matrix is estimated and tested in multiple positive responses returned by SVM classifiers. (a) Left: Patch 1. Right: SVM positive responses. (b) Left: Patch 2. Right: SVM positive responses. (c) Fundamental candidate F is estimated by the best match of Patch 1. (d) The Candidate fundamental matrix estimated from Patch 1 is tested on Patch 2 by searching for the best response of Patch that satisfied F.

positive patches and treat them as negative samples, as shown in the blue boxes in Fig. 2.4.

4. Train an SVM for the patch feature given all positive and negative samples. Discard the patch features whose samples cannot be separated successfully by linear SVM, which means they are not salient enough to be distinguished from other neighboring patches.

The output of SVM training is a set of salient image segments $S = \{C_i\}$ and their corresponding SVM classifiers over histogram of oriented gradients (HOG) features $D = \{d_i\}$. In the image matching procedure, d_i is used as feature descriptors to search for features similar to p_i.

2.3.4 Image Feature Matching with Geometry Model Selection

We measure the similarity between two images through a similarity quantification of corresponding proposed patch features. For each patch feature C_i, the corresponding SVM classifier d_i can be used to search for similar patches in the matching image. The downside of the SVM classifier is false alarms, as shown in Fig. 2.5(a) and Fig. 2.5(b). In order to

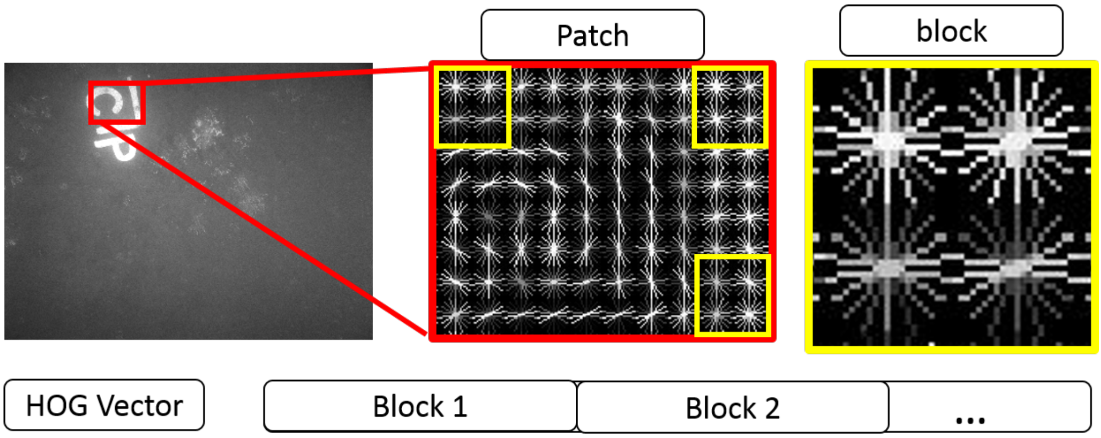


Figure 2.6: HOG Structure: A patch is divided into a set of *blocks* by a sliding window. a *block* is divided by 4 *cells*. Within each *cell*, a 9-bin histogram of oriented gradient is accumulated. The *block feature* is a vector of 4 9-bins histograms. The *HOG feature* is a vector of all the *block features*.

reject false positive SVM matches and achieve a more reliable result, a model selection strategy is employed based on a geometry model.

The relationship between two image points looking into a same scene point is constrained by the underlying geometry relationship. One of the most popular definitions for this constraint is the *fundamental matrix* F [40], which is widely used for point-based feature matching and structure estimation in two-view or multi-view systems. It defines this relationship as follows:

$$x_1^T F x_2 = 0 \tag{2.1}$$

where x_1 and x_2 are homogeneous coordinates of corresponding image points. F is defined up to scale with one free dimension, so at least 8 pairs of points are needed for estimating F using the standard linear algorithm. However, 8 pairs of patch correspondences are hard to satisfy given the low density of features in underwater images. To deal with this problem, the sub structure of HOG features are used. As depicted in Fig. 2.6, the HOG feature is constructed from a set of sub-regions (blocks) in a larger bounding box. Each block in the bounding box contains four cells. A 9-dimension gradient histogram will be extracted for each cell. The block itself is a more localized feature describing a small area relative to the whole HOG feature. By matching block features between two matched patches, a fundamental matrix that is consistent with the matched pair can be estimated.

An outline of our proposed image matching algorithm using the patch features is given in Algorithm 2. Each pair of potential matched image patches (C_i, M_i^u) will be used to propose a hypothetical fundamental matrix F_j. For all proposed hypothetical F_j, the test of all the matching patches is done following the relationship defined in Equation 2.1. The F^* that achieves the maximum number of supporting patch pairs is the winner of the model selection and its supporting image patch $S^* = \{C_i^*\}$ is the successfully matched patch in the image matching. The final matching score is given by :

$$S_m = \Sigma_{j=1}^{|S^*|} Area(C_j^*)/Area(Img1) \tag{2.2}$$

where $|S^*|$ is the number of successfully matched patches.

Algorithm 2 Image matching using geometry model selection

```
input
img1, img2: Matching images.
S: Salient segments in img1.
D : SVM classifiers corresponding to S.
initialization
$F = \phi$: the set of geometry models between two images
for $i = 1$ to $|D|$ do
    $M_i := \text{SVMTest}(d_i, img2)$
    for $u = 1$ to $|M_i|$ do
        $F = F \cup \text{ModelExtract}(C_i, M_i^u))$
    end for
end for
$Best\_F := \phi$
$Max\_Num\_Support := 0$
for $j = 1$ to $|F|$ do
    $Num\_Support = \text{ModelTest}(S, \{M_i\}; F_j)$
    if $Num\_Support > Max\_Num\_Support$ then
        $F^* = F_j$
    end if
end for
return $F^*$
```

As shown in Fig. 2.5, the model selection based on geometric constraint is able to improve the system by filtering out false alarms in the SVM response. A verification of the effectiveness of the geometry constraint is provided in Section 2.5 by comparing performance of the system with and without it.

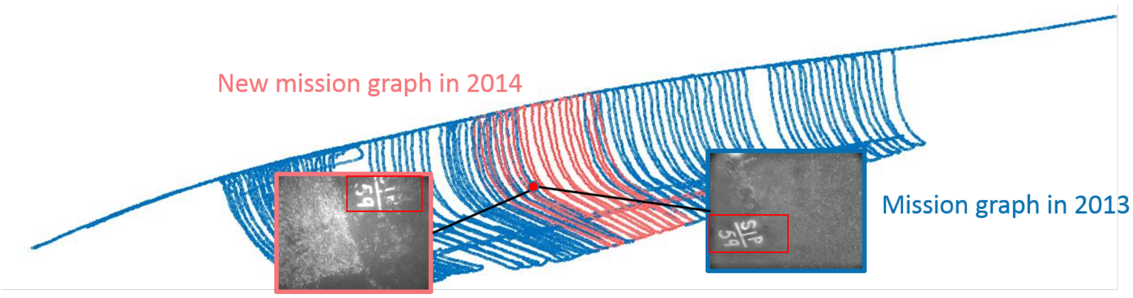


Figure 2.7: Multi-session graph merging overview. Here we show two graph SLAM graphs merged in a common frame using our graph merging algorithm. A graph from inspection conducted in 2013 is depicted in blue, while a new mission graph from 2014 is colored red. The task of merging is to determine the 6 DOF rigid body transformation between the two graphs through all the sensor measurements onboard an HAUV, which can be used by the SLAM back-end to merge the two graphs as a single optimization problem. An example of matched images between the two graphs found by our approach is shown in the image block.

2.4 Multi-session SLAM Graph Merging for Long-term Hull Inspection

Periodic inspection and structural defect assessment of large ships is an essential but expensive and time-consuming task in vessel maintenance, which is most often done by dry-docking the vessel. Recently, the task has evolved to include the deployment of a variety of remote-controlled or unmanned underwater robots to inspect and reconstruct the underwater portion of the ship hull *in situ* [59, 74, 82]. A consistent task includes periodic inspection conducted every few months or once a year. Being able to register inspection results across the inspection circle is of great importance to evaluate the structural changes and prioritize inspection plans.

In this section, we adapt the proposed high-level feature matching approach in a practical underwater localization task that enables graph merging across years in a Multi-session SLAM pipeline. A illustration of the expected scenario is depicted in Fig. 2.7.

The task is fulfilled using a Particle Filtering (PF) framework taking all the measurements onboard an HAUV. This work extends the previous work of Ozog and Eustice [82], which uses PF before the visual evidence being considered, then conducts a best-of-all image matching strategy to identify the best estimated geometry relationship. In our work, we explicitly model high-level image feature similarity matching within the PF framework to include more pairs of visual evidence to make a robust decision.

A flowchart of the proposed algorithm is given in Fig. 2.8. An on-line PF estimates the

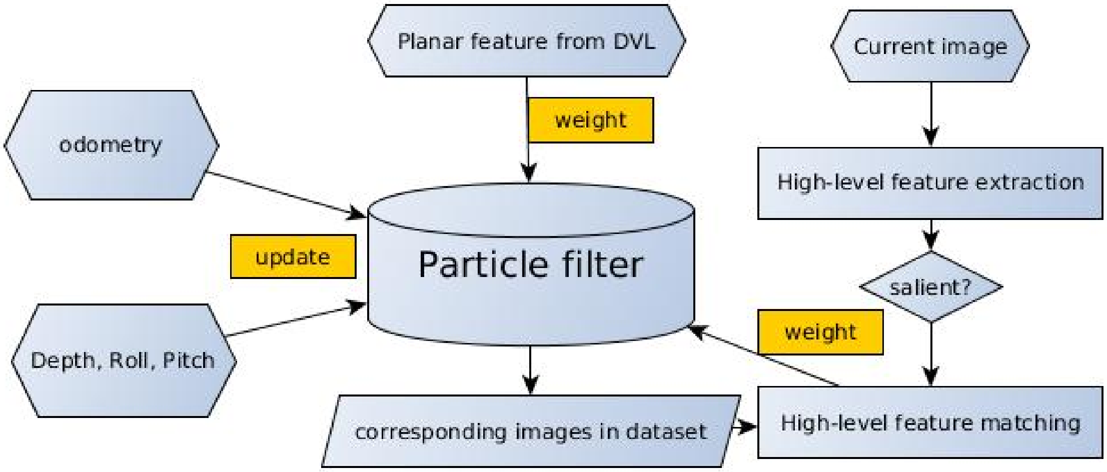


Figure 2.8: System Flowchart. A particle filter is initialized on a previous SLAM graph that we will localize against. Onboard sensors including depth and IMU are used to update the particles when the vehicle is moving. Planar features estimated from DVL as well as high-level features extracted from visual images are used to make measurement corrections by updating the importance weight of each particle.

6 DOF current position of the robot with respect to a previous graph as the robot moves. All sensor measurements on an HAUV are used, including an onboard depth sensor, an inertial measurement unit (IMU), a Doppler velocity log (DVL) and visual features from camera images. In the following section, we describe in detail how we model these measurements in the PF system.

2.4.1 Particle Filtering

In the proposed method, we use a particle filter to provide an estimate of the vehicle's pose distribution, which enables us to incorporate measurements from different modalities and control inputs from other onboard sensors:

$$p(x_t|Z_{1:t}, U_{1:t}) \propto$$

$$p(Z_t|x_t)p(x_t|x_{t-1}, U_t)p(x_{t-1}|U_{1:t-1}, Z_{1:t-1}).$$

Vehicle pose consists of position and orientation, $x_t = x, y, z, roll, pitch, yaw$. U is the set of control inputs used to propagate particles and Z is the set of observations (i.e., measurements).

1. Updating: We incorporate two different control inputs for particle propagation. An

onboard depth sensor and IMU provide direct measurement of depth, roll and pitch. The odometry input estimated from DVL provides a standard odometry update: $p(x_t|u_t^{odom}, x_{t-1}) \sim N(x_{t-1} \oplus u_t^{odom}, \Sigma_{U_z^{odom}})$.

2. Weighting: Two measurements are considered in the system: planar measurements (z_t^{dvl}) and visual measurements (z_t^{cam}); $Z_t = \{z_t^{dvl}, z_t^{cam}\}$ are assumed to be independent given a vehicle pose. We use the planar feature proposed in [82] to evaluate DVL measurements. Given the planar-like shape of the ship hull in the camera field of view, [82] estimates a planar feature $z_{\pi t}$ using Principal Component Analysis (PCA) on DVL point-based outputs. We set up the measurement model as follows: $w_p^f = p(z_t^{dvl}|x_t^p) \sim ||z_{\pi t} - \hat{z}_{x_p}||$, where $\hat{z}_{x_p}$ is the expected planar factor estimated from planar features in the old graph nearest to the particle x_p. For visual measurements, $w_p^{dvl} = p(z_t^{dvl}|x_p)$ is assigned according to the high-level feature matching result between the current observed image and the nearest image in the old graph corresponding to state x_p. More details of visual feature weighting will be discussed in 2.4.2.

2.4.2 Particle Weighting Using Visual Matching

When an image with its high-level feature descriptors is passed into the particle filter, the image will be matched against all the images in the previous dataset corresponding to all the particle positions. A matching score s_m is calculated following Equation 2.2 for each particle.

To maintain the diversity in the particle filter representation, the number of particles is often larger than the number of nodes in the old graph. Thus, multiple particles are associated with the same candidate image from the old graph. In other words, they share the same s_m as visual measurement confidence.

Given the geometric relationship we estimated from the image matching procedure, a particle weight is calculated based on the image matching score as well as the geometric consistency: $w_p^{cam} = s_g^p \cdot s_m^p$, where s_g^p is the geometric consistency score of the current particle position given the estimated position from the epipolar constraint in image matching: $s_g^p = \frac{TT'}{|T'|}$. T is the estimated transformation from the position of the current image to the candidate image position, and T' is the putative position calculated between the particle position and the candidate image position. This step also increases the convergence speed of the particle filter in that it utilizes the output information of image matching and penalizes particles that are inconsistent with the underlying image geometry.

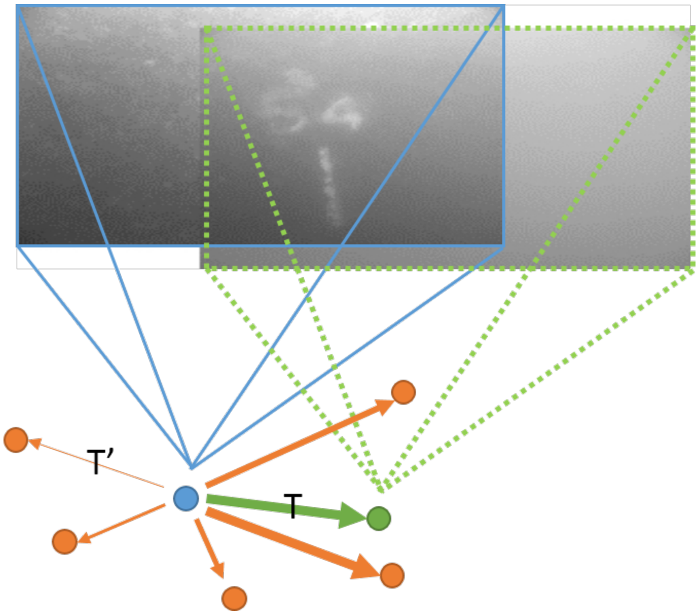


Figure 2.9: This figure gives an example when a set of particles (orange dot) is associated with a single image node in a previous graph (blue dot). The transform vector T from the old graph node to the new one (green dot) is estimated in image matching. The transform vector T' from old graph node to particles are compared with T. The line weights of T' in the figure indicate the relative value of s_g in this example.

2.5 Experimental Results and Discussion

To evaluate the robustness of the proposed image matching algorithm for place recognition, a set of experiments and analyses were performed using real-data collected in a multi-year underwater ship hull inspection task. We first introduce the data and ground truth used in the experiments in Section 2.5.1. Experimental results of image matching compared to other state-of-the-art algorithms are given in Section 2.5.2. Detailed analysis of the image matching performance of the intermediate steps of the framework are discussed in Section 2.5.2. Qualitative analysis of multi-session graph merging in the HAUV navigation system is given in Section 2.5.3.

2.5.1 Experimental Setting

Experiments are performed on data collected by a Hovering Autonomous Underwater Vehicle (HAUV) over three years (2011, 2013, 2014) on the SS Curtiss [59, 82]. External ground truth for underwater data is extremely challenging to gather, so we elected to use hand-labeled data for place recognition ground-truth.

Seven places, which included 358 images, were manually labeled from the dataset,

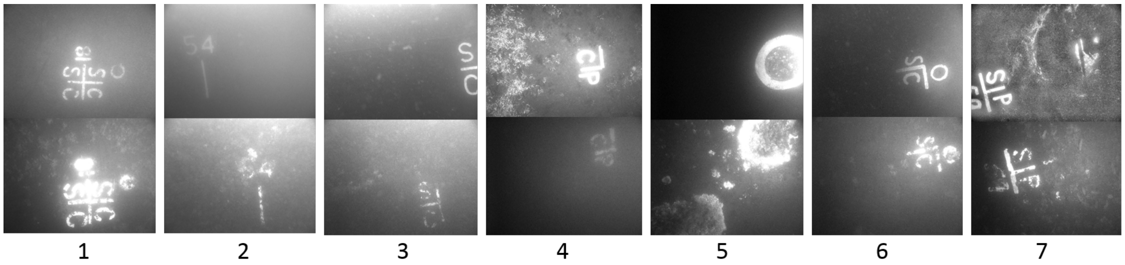


Figure 2.10: Examples of manually labeled corresponding images from 7 places on the SS Curtiss across different years. The images in the first row come from the dataset we are searching in. The images in the second row are current images. It can be seen that the data is quite challenging, due to the dramatic decay of patterns and changing light conditions.

where correspondences could be identified by a human. We used the data from the year 2013 (118 images) as the *query* dataset, and the data from the years 2011 and 2014 (240 in total) for matching (what we will refer to as the set of *current* images). One hundred unlabeled images from the mission data in 2013 were randomly sampled (images in unknown places) to be included as noise images to ensure the matching is robust enough to distinguish true matches from non-salient noise. Some examples of correspondences in the manually labeled dataset are shown in Fig. 2.10. It can be seen that the dataset is quite challenging with dramatic changes in the scene due to decay and biofouling across the years.

2.5.2 Image Matching using High-level Feature

2.5.2.1 Comparing to Standard Place Recognition Methods

In this experiment, we compare the image matching performance of our proposed algorithm with the performance of two widely-used representative point-feature based place recognition methods. The scale-invariant feature transform (SIFT) feature matching with random sample consensus (RANSAC) for geometric model estimation and outliers rejection is included as a benchmark approach for image matching. A bag-of-words (BoW) model on SIFT is also compared in the experiment. BoW trains a vocabulary of SIFT features from the query dataset and describes images using the BoW vocabulary. For each image in the testing set, all the SIFT features detected are converted into a single description vector based on the vocabulary. Image matching is then done based on $L2$ distance between description vectors between images.

The confusion matrix of matching results for different places in the manual labeled dataset is shown in Fig. 2.11. The result shows that the proposed method is able to provide reasonable matching results in a challenging dataset, while the other two standard methods underperform. The benchmark methods mostly get distracted by the noise image set included in the comparison. This is mainly attributable to the fact that these images capture some biofouling that is more rich in feature points leading to false positive matching. This comparison is expected since point-based features are often unable to generate any meaningful correspondences between images with dramatic appearance changes.

It is also clear in the confusion matrix that the proposed method can perform quite differently in different places. For some places, the proposed method is able to provide almost 80% accuracy given the dramatic appearance changes, while the performance is less ideal for other locations. To discuss the experiment result in a more thorough manner and understand what makes this difference, we show some representative examples of both true positive and false positive matches in Fig. 2.12 and Fig. 2.13. From Fig. 2.12 we can tell that reliable matching is often made when multiple pairs of salient features can be found, in which case the model selection step in the pipeline can take place and reject the outliers from weak SVM. In Fig. 2.13, false positive matches defeat the correct ones when the true matching is only supported by a single pair of features. The system is not able to properly determine which matches are of greater importance within one image in these cases. This indicates that a frame-related patch importance weight is needed to improve the robustness of our proposed algorithm. In some other extreme cases, the decay is just too advanced for our image segmentation approach to extract a meaningful patch, and the proposed method fails to provide a robust match.

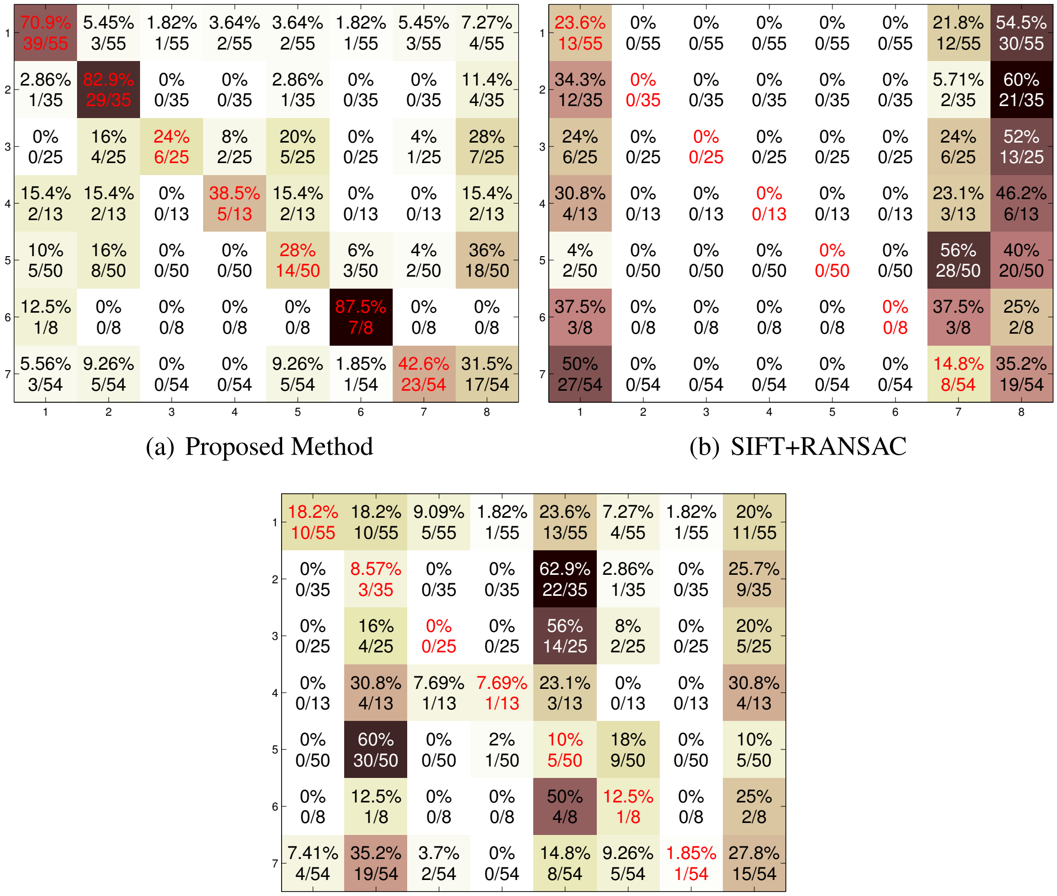
However, these results are from single image matching, and can be improved within a complete system. By incorporating the proposed matching algorithm in a recursive filtering framework, the final localization performance will increase when more matching results are accumulated, as shown in Section 2.5.3.

Although the approach is proposed for underwater imagery addressing feature sparsity and appearance changes, it can also be used in other domains. Some matching examples of the proposed method used in above-water ship hull inspection images are also provided in Fig. 2.14 to show that the approach is able to be generalized to non-underwater images.

2.5.2.2 Self-comparison for System Component Analysis

To get a better idea of how the proposed approach is able to make reasonable correspondences between images across different years, we also analyzed the performance with and without certain components of the framework. The results are shown in Fig. 2.15.

We replaced the SVM classifier of HOG descriptors with Normalized Cross Correlation (NCC) in searching for matched patches. The result is displayed in Fig. 2.15(a). Comparing that to the original proposed method (Fig. 2.11), it can be seen that SVM with HOG features



	1	2	3	4	5	6	7	8
1	70.9% 39/55	5.45% 3/55	1.82% 1/55	3.64% 2/55	3.64% 2/55	1.82% 1/55	5.45% 3/55	7.27% 4/55
2	2.86% 1/35	82.9% 29/35	0% 0/35	0% 0/35	2.86% 1/35	0% 0/35	0% 0/35	11.4% 4/35
3	0% 0/25	16% 4/25	24% 6/25	8% 2/25	20% 5/25	0% 0/25	4% 1/25	28% 7/25
4	15.4% 2/13	15.4% 2/13	0% 0/13	38.5% 5/13	15.4% 2/13	0% 0/13	0% 0/13	15.4% 2/13
5	10% 5/50	16% 8/50	0% 0/50	0% 0/50	28% 14/50	6% 3/50	4% 2/50	36% 18/50
6	12.5% 1/8	0% 0/8	0% 0/8	0% 0/8	0% 0/8	87.5% 7/8	0% 0/8	0% 0/8
7	5.56% 3/54	9.26% 5/54	0% 0/54	0% 0/54	9.26% 5/54	1.85% 1/54	42.6% 23/54	31.5% 17/54

(a) Proposed Method

	1	2	3	4	5	6	7	8
1	23.6% 13/55	0% 0/55	0% 0/55	0% 0/55	0% 0/55	0% 0/55	21.8% 12/55	54.5% 30/55
2	34.3% 12/35	0% 0/35	0% 0/35	0% 0/35	0% 0/35	0% 0/35	5.71% 2/35	60% 21/35
3	24% 6/25	0% 0/25	0% 0/25	0% 0/25	0% 0/25	0% 0/25	24% 6/25	52% 13/25
4	30.8% 4/13	0% 0/13	0% 0/13	0% 0/13	0% 0/13	0% 0/13	23.1% 3/13	46.2% 6/13
5	4% 2/50	0% 0/50	0% 0/50	0% 0/50	0% 0/50	0% 0/50	56% 28/50	40% 20/50
6	37.5% 3/8	0% 0/8	0% 0/8	0% 0/8	0% 0/8	0% 0/8	37.5% 3/8	25% 2/8
7	50% 27/54	0% 0/54	0% 0/54	0% 0/54	0% 0/54	0% 0/54	14.8% 8/54	35.2% 19/54

(b) SIFT+RANSAC

	1	2	3	4	5	6	7	8
1	18.2% 10/55	18.2% 10/55	9.09% 5/55	1.82% 1/55	23.6% 13/55	7.27% 4/55	1.82% 1/55	20% 11/55
2	0% 0/35	8.57% 3/35	0% 0/35	0% 0/35	62.9% 22/35	2.86% 1/35	0% 0/35	25.7% 9/35
3	0% 0/25	16% 4/25	0% 0/25	0% 0/25	56% 14/25	8% 2/25	0% 0/25	20% 5/25
4	0% 0/13	30.8% 4/13	7.69% 1/13	7.69% 1/13	23.1% 3/13	0% 0/13	0% 0/13	30.8% 4/13
5	0% 0/50	60% 30/50	0% 0/50	2% 1/50	10% 5/50	18% 9/50	0% 0/50	10% 5/50
6	0% 0/8	12.5% 1/8	0% 0/8	0% 0/8	50% 4/8	12.5% 1/8	0% 0/8	25% 2/8
7	7.41% 4/54	35.2% 19/54	3.7% 2/54	0% 0/54	14.8% 8/54	9.26% 5/54	1.85% 1/54	27.8% 15/54

(c) Bag-of-Words

Figure 2.11: Results from our proposed technique and from point-based approaches from the literature, presented as confusion matrices where the rows and columns refer to places or clusters of co-located images of which samples are shown in Fig. 2.10. Column 8 represents the set of noise images added to the dataset to make the task more challenging. The red font indicates the accuracy rate of the matching result for each place, while the rest of the row indicates how the false matching is distributed in the query dataset.

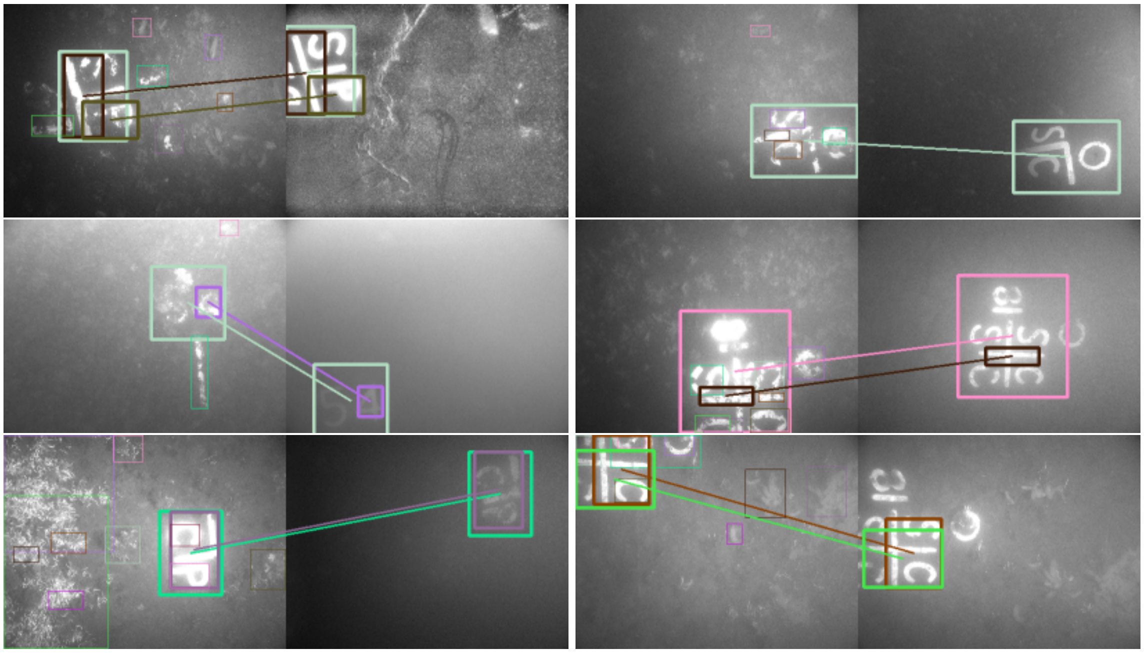
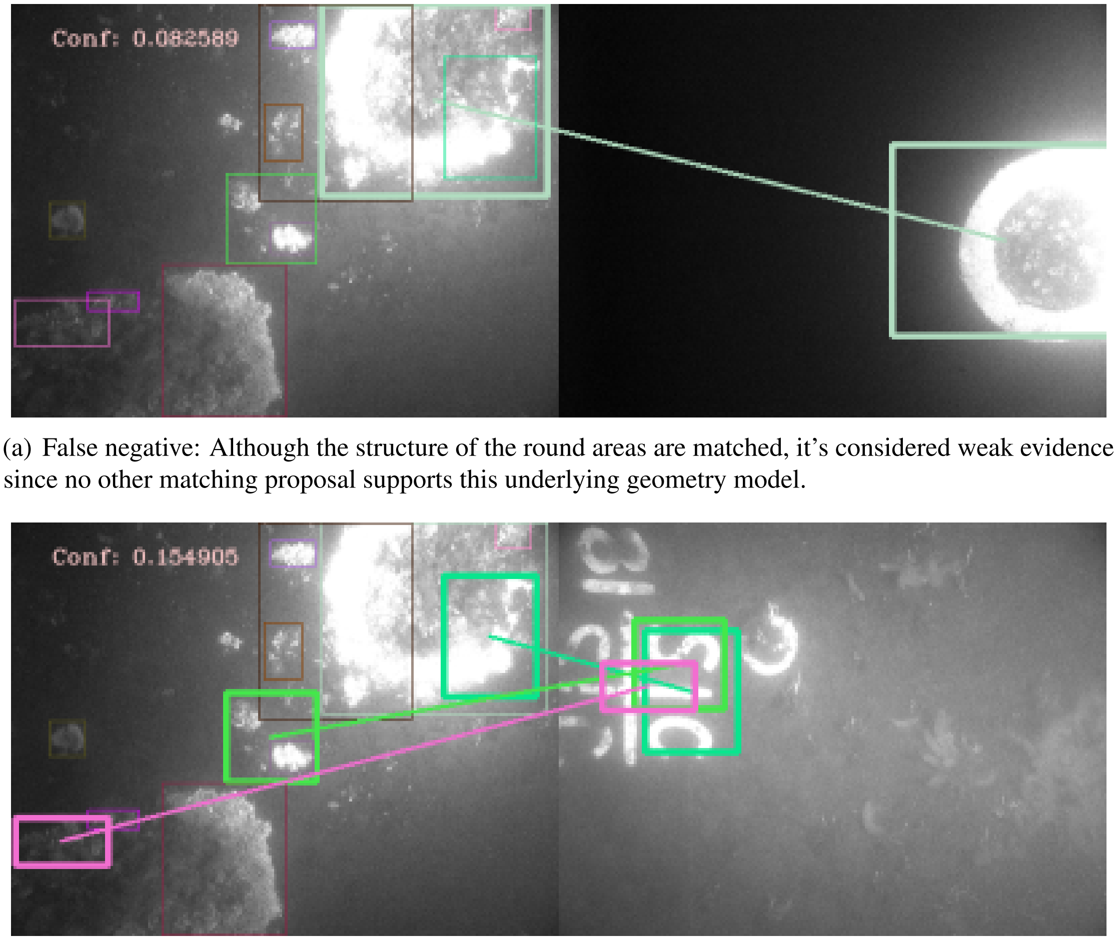


Figure 2.12: Examples of successfully matched pairs. The system is able to make valid correspondences between images with dramatic appearance changes.

has greater robustness with respect to providing valid matches, compared to NCC. This is because HOG features have a higher tolerance for small rotation and illumination changes than the image patch intensity comparison.

The contribution of geometric validation on false positive rejection is also demonstrated by the comparison experiment with and without it, as shown in Fig. 2.11 and Fig. 2.15(b) respectively. It can be seen that the geometric constraint improves the system's robustness significantly by rejecting false positive responses returned by the SVM classifiers. Without the geometric constraint, matches tend to happen between two images with several salient regions, in which case many false positive matches are returned and are included in the final matching score.

To evaluate how the use of neighboring images is contributing to the algorithm, we conduct the comparison experiment without the use of neighboring images. The result is shown in Fig. 2.15(c). The proposed method supports the use of a default maximum number of neighboring images ($N = 4$), which improves the final performance of the proposed method. The importance of neighboring image sets is not obvious in this specific dataset, since the light conditions are quite consistent between neighboring images. However, it is still an important part of the system in other scenarios when light conditions have a greater impact on image view point or when neighboring images taken from different periods can be considered.



(a) False negative: Although the structure of the round areas are matched, it's considered weak evidence since no other matching proposal supports this underlying geometry model.

(b) False positive: Many small areas are matched, and the system determines it is a stronger overall match.

Figure 2.13: Example of false positive and false negative matches using the proposed system.

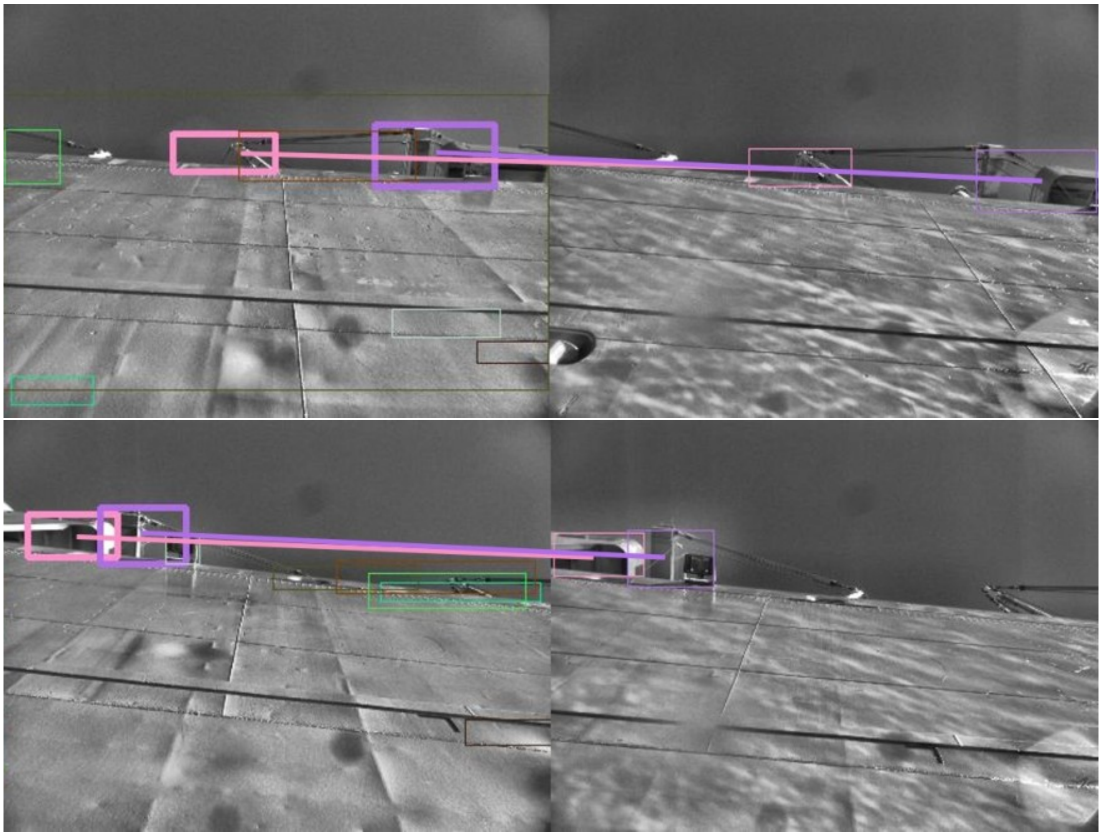
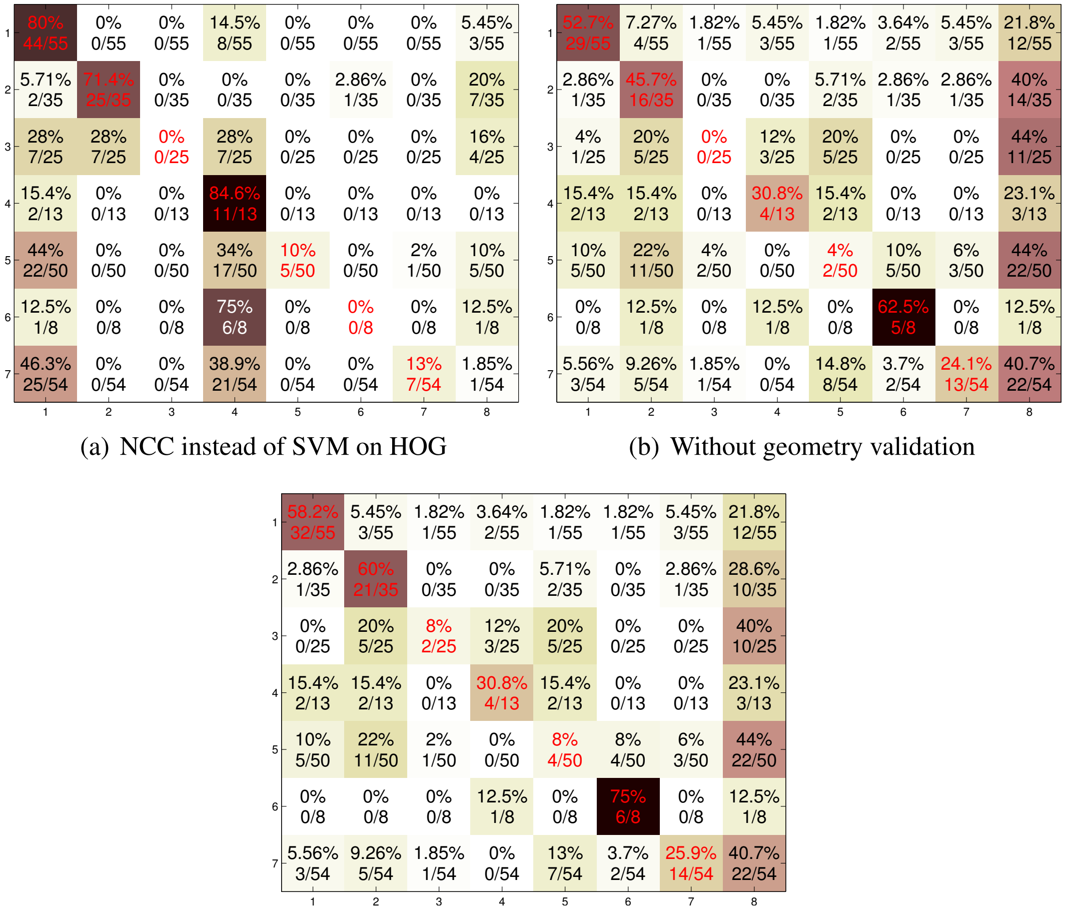


Figure 2.14: Examples of image matching using the proposed method in above-water images.



	1	2	3	4	5	6	7	8
1	80% 44/55	0% 0/55	0% 0/55	14.5% 8/55	0% 0/55	0% 0/55	0% 0/55	5.45% 3/55
2	5.71% 2/35	71.4% 25/35	0% 0/35	0% 0/35	0% 0/35	2.86% 1/35	0% 0/35	20% 7/35
3	28% 7/25	28% 7/25	0% 0/25	28% 7/25	0% 0/25	0% 0/25	0% 0/25	16% 4/25
4	15.4% 2/13	0% 0/13	0% 0/13	84.6% 11/13	0% 0/13	0% 0/13	0% 0/13	0% 0/13
5	44% 22/50	0% 0/50	0% 0/50	34% 17/50	10% 5/50	0% 0/50	2% 1/50	10% 5/50
6	12.5% 1/8	0% 0/8	0% 0/8	75% 6/8	0% 0/8	0% 0/8	0% 0/8	12.5% 1/8
7	46.3% 25/54	0% 0/54	0% 0/54	38.9% 21/54	0% 0/54	0% 0/54	13% 7/54	1.85% 1/54

(a) NCC instead of SVM on HOG

	1	2	3	4	5	6	7	8
1	52.7% 29/55	7.27% 4/55	1.82% 1/55	5.45% 3/55	1.82% 1/55	3.64% 2/55	5.45% 3/55	21.8% 12/55
2	2.86% 1/35	45.7% 16/35	0% 0/35	0% 0/35	5.71% 2/35	2.86% 1/35	2.86% 1/35	40% 14/35
3	4% 1/25	20% 5/25	0% 0/25	12% 3/25	20% 5/25	0% 0/25	0% 0/25	44% 11/25
4	15.4% 2/13	15.4% 2/13	0% 0/13	30.8% 4/13	15.4% 2/13	0% 0/13	0% 0/13	23.1% 3/13
5	10% 5/50	22% 11/50	4% 2/50	0% 0/50	4% 2/50	10% 5/50	6% 3/50	44% 22/50
6	0% 0/8	12.5% 1/8	0% 0/8	12.5% 1/8	0% 0/8	62.5% 5/8	0% 0/8	12.5% 1/8
7	5.56% 3/54	9.26% 5/54	1.85% 1/54	0% 0/54	14.8% 8/54	3.7% 2/54	24.1% 13/54	40.7% 22/54

(b) Without geometry validation

	1	2	3	4	5	6	7	8
1	58.2% 32/55	5.45% 3/55	1.82% 1/55	3.64% 2/55	1.82% 1/55	1.82% 1/55	5.45% 3/55	21.8% 12/55
2	2.86% 1/35	60% 21/35	0% 0/35	0% 0/35	5.71% 2/35	0% 0/35	2.86% 1/35	28.6% 10/35
3	0% 0/25	20% 5/25	8% 2/25	12% 3/25	20% 5/25	0% 0/25	0% 0/25	40% 10/25
4	15.4% 2/13	15.4% 2/13	0% 0/13	30.8% 4/13	15.4% 2/13	0% 0/13	0% 0/13	23.1% 3/13
5	10% 5/50	22% 11/50	2% 1/50	0% 0/50	8% 4/50	8% 4/50	6% 3/50	44% 22/50
6	0% 0/8	0% 0/8	0% 0/8	12.5% 1/8	0% 0/8	75% 6/8	0% 0/8	12.5% 1/8
7	5.56% 3/54	9.26% 5/54	1.85% 1/54	0% 0/54	13% 7/54	3.7% 2/54	25.9% 14/54	40.7% 22/54

(c) Without neighboring image information

Figure 2.15: System component analysis. Identification of the contribution of each system component by comparison experiment without certain components or with those components replaced by other simpler alternatives. Results presented as confusion matrices where rows and columns refer to places or clusters of co-located images of which samples are shown in Fig. 2.10. Note the last column has no matched row as it refers to non-labeled random images introduced to the dataset to make matching more challenging/realistic.

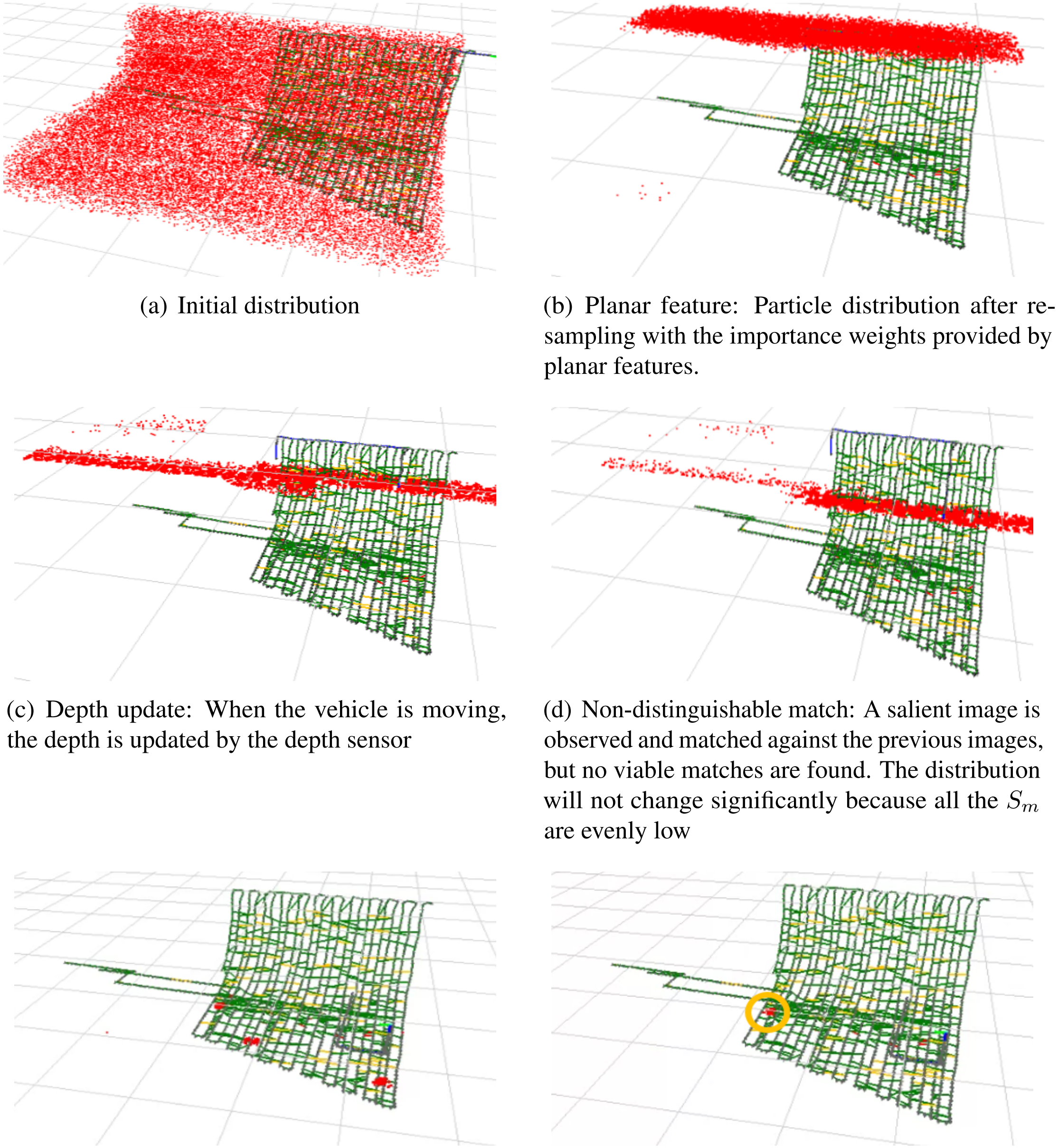
2.5.3 Relocalization using Particle Filtering

In this section, we evaluate the performance of the proposed PF-based localization framework by localizing the vehicle to a SLAM graph collected years ago. The proposed method is able to localize the vehicle in a previously-built SLAM graph given dramatic appearance changes.

One localization mission that is illustrative of the system is shown in Fig. 2.16. As can be seen in Fig. 2.16(a), the particles are initialized with uniform distribution and updated using estimates from the depth sensor. When a salient image is observed, but the system is unable to find a viable match, the score reflects this and the distribution remains unaffected (see Fig. 2.16(d)). When a salient image produces a viable match, the particles start to converge to the locations where the matching scores are high, as shown in Fig. 2.16(e). However, the visual evidence provided by a single salient image is insufficient for localization. The particles converge to some candidate regions where the matching scores are much higher than the rest. As the mission continues, more salient images are used to correct the particle distribution and particles converge to the correct localization solution. When the determinant of the covariance passes a threshold, the system considers itself localized. Fig. 2.17 displays the matching pair with the highest score one time-step before the system converges. The highly structured visual features are identified in the boxes and correctly matched across years despite the significant ship hull biofouling.

2.6 Conclusion

A high-level visual feature image matching approach is presented to address the challenge of underwater image matching under dramatic appearance changes. A salient region identification method is proposed to locate visually salient regions and find high-level features. SVM classifiers based on HOG features are used for feature description and matching. Geometric constraints are employed to reject false positive matches provided by SVM. The approach is evaluated on real data collected in multi-year ship hull inspection missions and the performance is compared with other standard place recognition methods. The proposed method strongly outperforms traditional point-based feature matching techniques. To further illustrate the practical value of the proposed method, we incorporate it in a PF-based localization framework and the result indicates that the localization accuracy can be improved through a real-world system.



(a) Initial distribution

(b) Planar feature: Particle distribution after resampling with the importance weights provided by planar features.

(c) Depth update: When the vehicle is moving, the depth is updated by the depth sensor

(d) Non-distinguishable match: A salient image is observed and matched against the previous images, but no viable matches are found. The distribution will not change significantly because all the S_m are evenly low

(e) Distinguishable match: A salient image is observed and matched against the previous image corresponding to particle positions. However, when more than one previous image is similar to the current observation, the particles converge to multiple candidate regions

(f) Particles converge: For the particles in the correct region, the highly salient images upweight the particles causing the distribution to converge to a small area

Figure 2.16: Typical particle filter converging procedure.

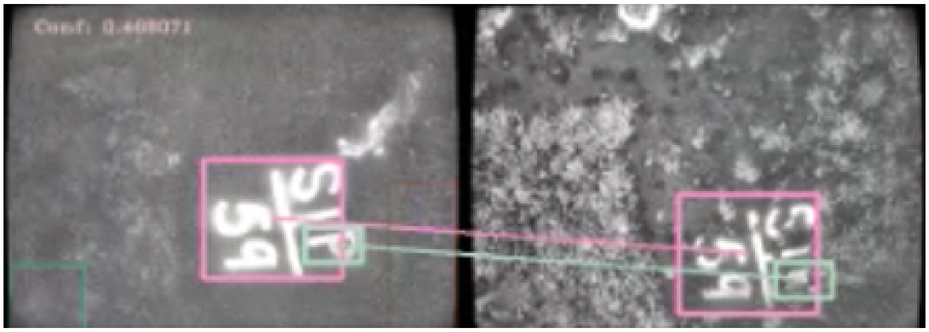


Figure 2.17: The best matched pair at the convergence location. Left: current image. Right: the best matched candidate image in the previous graph.

CHAPTER 3

Imaging Sonar Feature Learning

3.1 Introduction

In Chapter 2, we develop an advanced high-level feature detection and matching approach for optical images that increases the robustness and reliability of optical image-based place recognition systems for underwater navigation. However, optical sensors suffer from many range-limited effects in turbid water. In these conditions, acoustic sensors are most commonly employed over optical sensing due to issues of visibility, attenuation, and range. A new generation of high-definition Forward-looking Sonar (FLS) providing acoustic imagery with a wide field-of-view (FOV) [90] has recently introduced a promising alternative to optical cameras for underwater visually-based navigation. However, information extraction from sonar images suffers from unique challenges: *i)* the projection geometry for imaging sonar necessitates a departure from the standard optical pinhole camera models; *ii)* sonar-derived images have weak feature textures; and *iii)* despite their large field-of-view, sonar-derived images have high amounts of noise. These problems make it hard to directly harness well-established optical-based place recognition algorithms for sonar images, in the sense that traditional computer vision feature descriptors are typically not able to make correspondences between sonar frames [8, 47].

In this work, we address the sonar-based place recognition problem by using learned feature descriptors instead of traditional hand-crafted ones. We propose explicitly training sonar image descriptors that capture the image similarity by training a convolutional neural network (CNN) that maps sonar images onto Euclidean poses, as shown in Fig. 3.1. The proposed network architecture is built upon a network designed for image classification tasks (GoogLeNet [92]) and adjusted according to the training objective and sonar image properties. The proposed method is able to obtain strong performance in place recognition and global localization on real-world data. In addition to the proposed feature learning approach, we also propose an uncertainty score that estimates the saliency of a sonar image

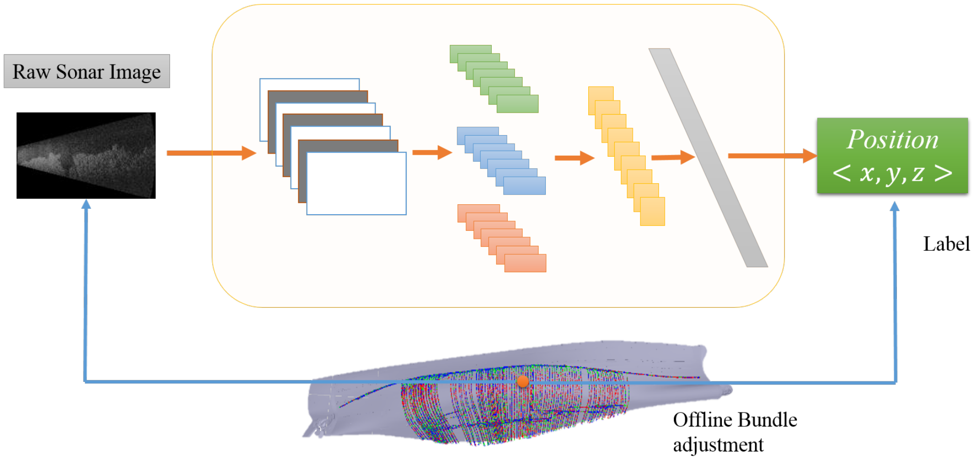


Figure 3.1: This work proposes to learn descriptive sonar image features using CNNs for underwater place recognition and localization. The learned feature space provides a similarity metric for sonar images that enables real-time place recognition. For sonar images collected within a short time period, the features can be directly used for global localization with respect to a known map. The data used in the experiment comes from a hovering autonomous underwater vehicle (HAUV) operating in a ship hull inspection application.

with respect to the localization task. In practice this can be used to rule out non-salient images to increase robustness and real-time system efficiency.

The contributions of this chapter are represented in our publication [67], which include:

1. A CNN architecture based on GoogLeNet [92] for general descriptive sonar image feature learning, which we apply for global localization.

2. A learning framework for estimating a sonar image saliency level for the task of place recognition.

3. A thorough set of experimental evaluations on real sonar images under varied feature dimensions and network architectures.

3.1.1 Outline

The rest of this chapter is laid out as follows: Section 3.2 gives a brief introduction of related work in visual-based place recognition and forward-scan sonar mapping. Section 3.3 presents the architecture of the proposed networks used to learn descriptive sonar image features and sonar image non-saliency scores. Section 3.4 introduces the multi-year ship hull inspection problem and associated dataset including the methodology for training label generation. In Section 3.5, we evaluate the proposed framework for the task of global localization as well as place recognition. Section 3.6 summarizes this chapter.

3.2 Background

3.2.1 Visual Place Recognition

For mobile ground robots, cameras have commonly been used as the primary perceptual sensor modalities. A great deal of work employing these sensors for visual place recognition has been discussed in recent years. Benchmark systems include fast appearance-based matching (FAB-MAP) [24], MonoSLAM [26], and FrameSLAM [60]. Most of these prior approaches are dominated by hand-crafted features such as scale-invariant feature transform (SIFT) [72] or speeded up robust features (SURF) [13]. More recently, the literature has focused on robustness against dramatic appearance changes for long-term simultaneous localization and mapping (SLAM) applications. Several new strategies have been introduced for this objective, such as matching sequences of images [75], predicting appearance change trends over time [80], and using higher-level features such as the histogram of oriented gradients (HOG) [65, 77]. All of the features used in the aforementioned approaches were designed for optical camera systems and as such have met with limited success in the acoustic domain [53, 78]. As forward-looking sonar has an imaging geometry that is different from optical cameras, most invariance properties of classical visual features are lost.

Only very recently have CNNs been considered in the field of place recognition. Chen et al. [22] proposed the first place recognition based on the mid-layer feature of a pre-trained model. Sünderhauf et al. [91] provide an extensive evaluation on the usage of mid-layers of pre-trained models for the task of visual place recognition. In that work they proposed using landmark features extracted from the mid-level layers of traditional pre-trained models. However, in contrast to the work presented here, the features used in those approaches are extracted from generalized networks not trained explicitly for the purpose of place recognition. We build on the work of Gomez-Ojeda et al. [36] and Kendall, Grimes, and Cipolla [58], who proposed the training of a network using a loss function in the Euclidean space specifically for place-recognition.

3.2.2 Imaging Sonar

Recent work in FLS imaging mostly focuses on image registration between two sonar images. Hurtos et al. [48] provide a thorough review on recent registration methods of sonar images. Popular approaches span from spectral methods [47] to feature-based methods [8, 53]. These approaches have shown promising results in pairwise motion estimation under certain assumptions about platform motion and scene geometry. These approaches are inapplicable for the general place recognition tasks as they assume a high degree of overlap and high frame rates. In fact, our proposed real-time sonar image place recognition method is complementary to such techniques and could be used as a pre-processing step for such registration methods in online navigation systems.

3.3 Methodology

3.3.1 Network Architecture

To explicitly train sonar image descriptors that provide an image similarity metric, we express network loss over the Euclidean distance of the sonar's position from ground-truth locations derived from a traditional SLAM system on a HAUV with odometry sensors. We build up the network structure based on a state-of-the-art CNN architecture originally proposed for object classification tasks, GoogLeNet [92]. In [92], the authors proposed a promising network architecture basic module called an 'inception module', which is shown in Fig. 3.2. In this module, three stacks of convolution layers with different kernel sizes are considered, capturing feature information at several different scales. Following each stack of convolution layers, a rectified linear unit (ReLu) layer is employed. We use this module and build up the proposed network, as depicted in Fig. 3.3. The input sonar images are fed into an initial convolution layer followed by 9 inception modules. Down-sampling pooling layers are used after the $\mathrm{inception\_2}$, $\mathrm{inception\_4}$ and $\mathrm{inception\_7}$ modules to compress the dimensionality of the feature space. The output of module $\mathrm{inception\_9}$ is passed to a fully connected layer, which generates the image feature vector. Finally, the feature vector goes through another fully connected layer that estimates the corresponding sonar position for the input image.

The loss function is defined as the L^2 distance between the estimated position and the ground-truth position, which is labeled in the training data:

$$loss(I) = ||\tilde{p}(I) - p_{gt}(I)||_2,$$

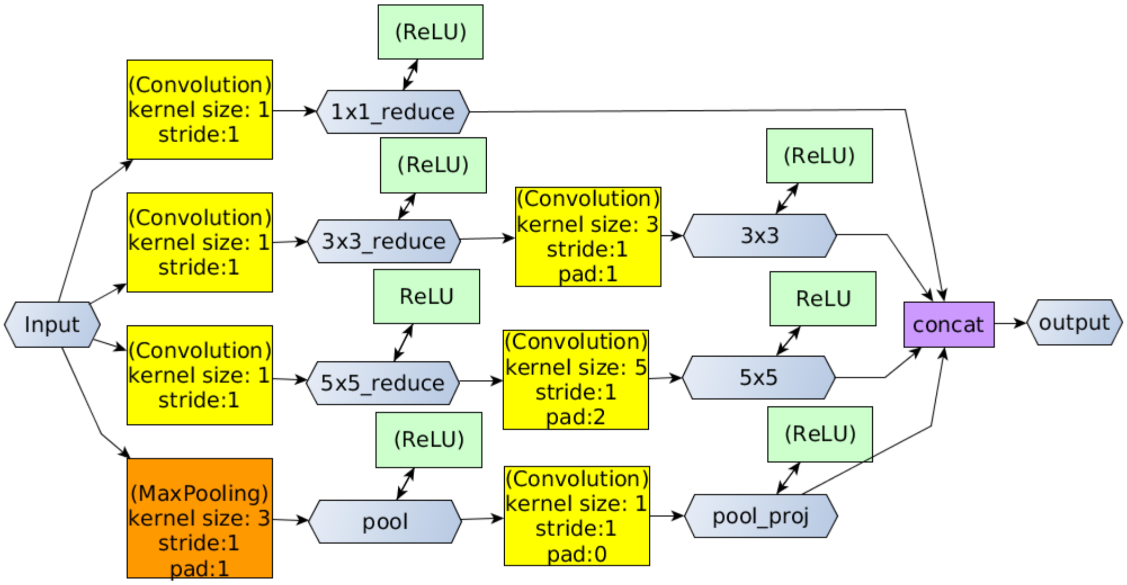


Figure 3.2: Inception Module: We use the inception module proposed by Szegedy et al. [92] as the basic module in our network architecture.

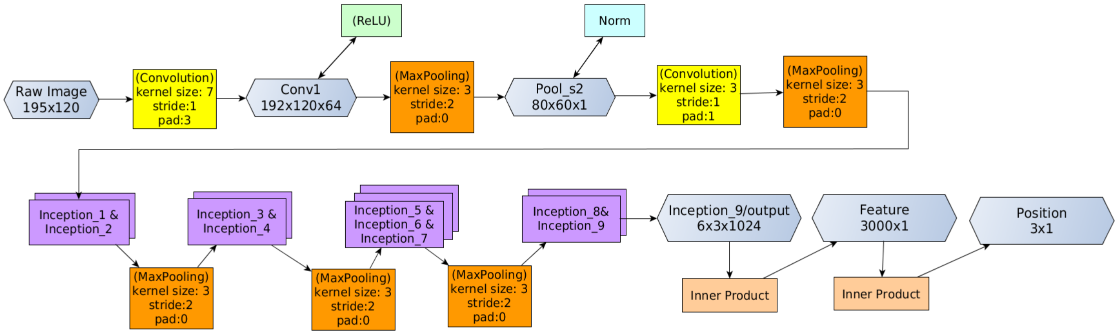


Figure 3.3: Network architecture used for feature learning. A sonar image down-sampled to 195×220 is the input of the network. The second top layer of the network is the learned descriptive feature. The top layer is a 3-element position estimator (i.e., x, y, z).

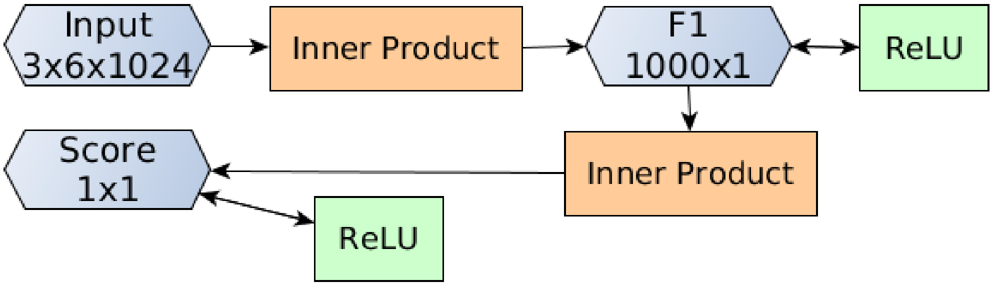


Figure 3.4: Simple neural network for uncertainty score learning.

where $\tilde{p}(I) = [x, y, z]^\top$ is the final 3-vector output of the proposed network, and $p_{gt}(I)$ is the ground-truth position. For underwater navigation, where global positioning system (GPS) position is not available, it is not trivially possible to obtain precise ground-truth. Instead, we use the output an off-line bundle adjustment as an alternative ground-truth for training and evaluation. More detail is given in Section 3.4 on how this ground-truth is estimated.

The major changes in our proposed network with respect to the original GoogLeNet are as follows. We propose: *i)* a different loss function; *ii)* a new feature layer; and *iii)* a different arrangement for dimensional compression. Compared to camera images, on which GoogLeNet is designed and tested, sonar images contain fewer structural features. To capture the low level information of sonar images, we increase the number of layers where input data is evaluated without dimension compression and apply the dimensionality reduction in later stages of the network. It is shown in Section 3.5 that our modified network achieves a better learning rate and is more robust to long-term appearance changes in the data.

3.3.2 Non-Saliency Score

In underwater applications such as ship hull inspection or seafloor survey, sonar images contain very low feature density. In some extreme cases, the images contain so little salient information that they cannot achieve meaningful place recognition. For real applications, the capability to evaluate the saliency of a sonar image prior to attempting localization is useful in increasing system efficiency and reliability.

We evaluate the saliency of a sonar image by learning a model between the image space and the localization uncertainty in the place recognition task. To achieve this we apply a supervised-learning approach to the features extracted from the third layer from the top in the proposed network. The network structure used is depicted in Fig. 3.4. The labels for the training stage are produced as follows:

1. For each image in the training set $\{I_i\}_{train}$, extract the learned feature vector f_i.

2. For each f_i, search for its nearest neighboring set in feature space:

$$NN_i = \{j \mid ||f_i - f_j||_2 \leq (1 + \lambda) \min ||f_i - f_k||_2\},$$

   where λ is a tolerance parameter controlling the size of neighboring set considered. In our experiments, $\lambda = 0.3$ was used.

3. Compute the positional standard deviation of the resulting neighboring set:

$$\sigma_{I_i} = \mathrm{std}(\{p_j \mid j \in NN_i\}),$$

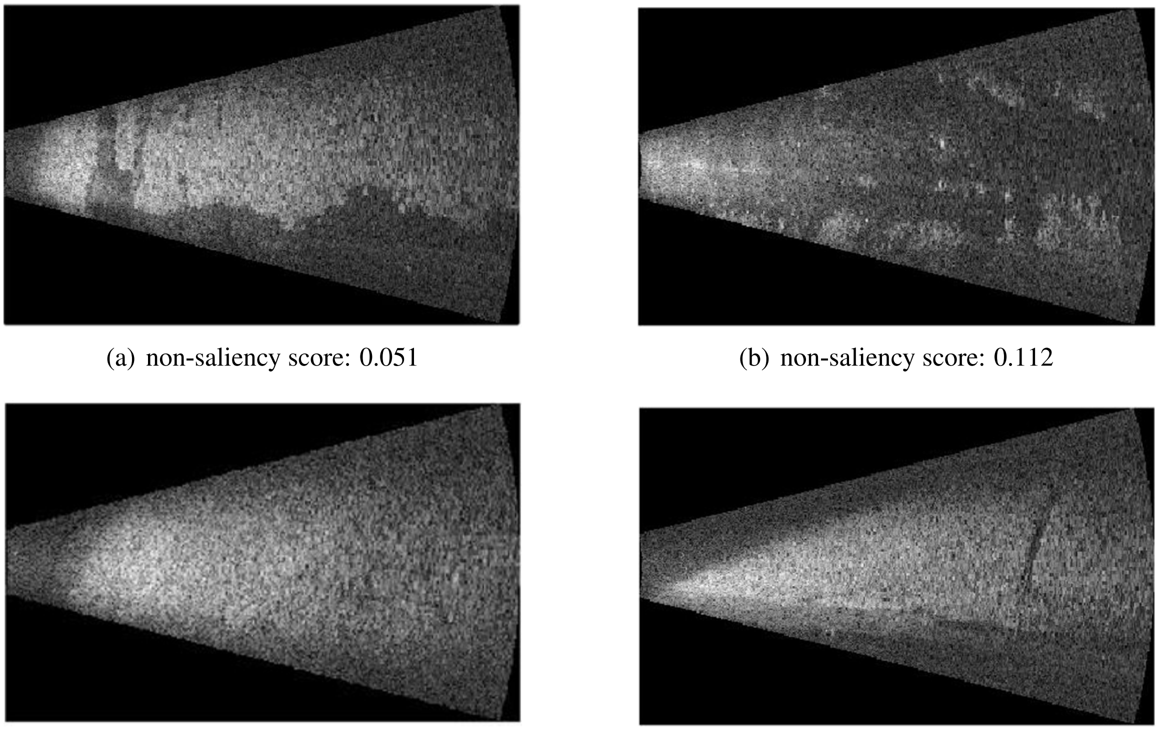
   where p_j is the 3D position of sample j. The larger the positional standard deviation, the more uncertain the image location is, hence the less salient the image must be for place recognition.

If the sonar image is salient and feature matching result is sensitive, then the NN_i is very small, resulting in a small uncertainty score; if the feature matching is not sensitive, NN_j might be large or contain features from an incorrectly associated position, resulting in a high uncertainty and consequently a low effective saliency for the image. Sample sonar images with different non-saliency scores are shown in Fig. 3.5. It can be seen that the images with high non-saliency scores contain fewer features or distinct information than the low score ones.

3.4 Dataset and Training

3.4.1 Label Generation

We trained the proposed CNN and evaluated its performance using data from a multi-year ship hull inspection application introduced in Section 1.1.2. The datasets used in the experiment were collected on a 183 m vessel, the *SS Curtiss* (Fig. 1.7). Two sets of data collected at different times are considered: March 2014 and June 2015. During the inspection mission, an online real-time visual SLAM system [82] provided vehicle navigation. An onboard depth sensor and an inertial measurement unit (IMU) were used to provide odometry. The optical cameras provided a visual constraint that corrects drift while the Doppler velocity log (DVL) range beams were used to estimate planar features to provide an additional drift correction constraint as described in [82]. The estimated



(a) non-saliency score: 0.051

(b) non-saliency score: 0.112

(c) non-saliency score: 18.875

(d) non-saliency score: 28.287

Figure 3.5: Sample sonar image with different non-saliency scores.

Table 3.1: Data Arrangement

Dataset	2014-03	2015-06
Training frames	25 191	35 751
Testing frames	2799	3972

trajectory of the online SLAM system was then used to provide an initial guess of sonar pose in an off-line bundle adjustment (BA) step that optimized all vehicle positions using all measurements. This served as our ground-truth estimate of vehicle position, which was then used to train the network.

3.4.2 Dataset Composition

To train the proposed network, we randomly separate the data into a training set and a testing set for each year. The statistics about each dataset are given in Table 3.1. Fig. 3.6 displays the spatial distribution of the dataset in the three collected missions. Different combinations of training and testing samples are used in different experiments to evaluate the system characteristics. The details specific to each experiment are described in §3.5.

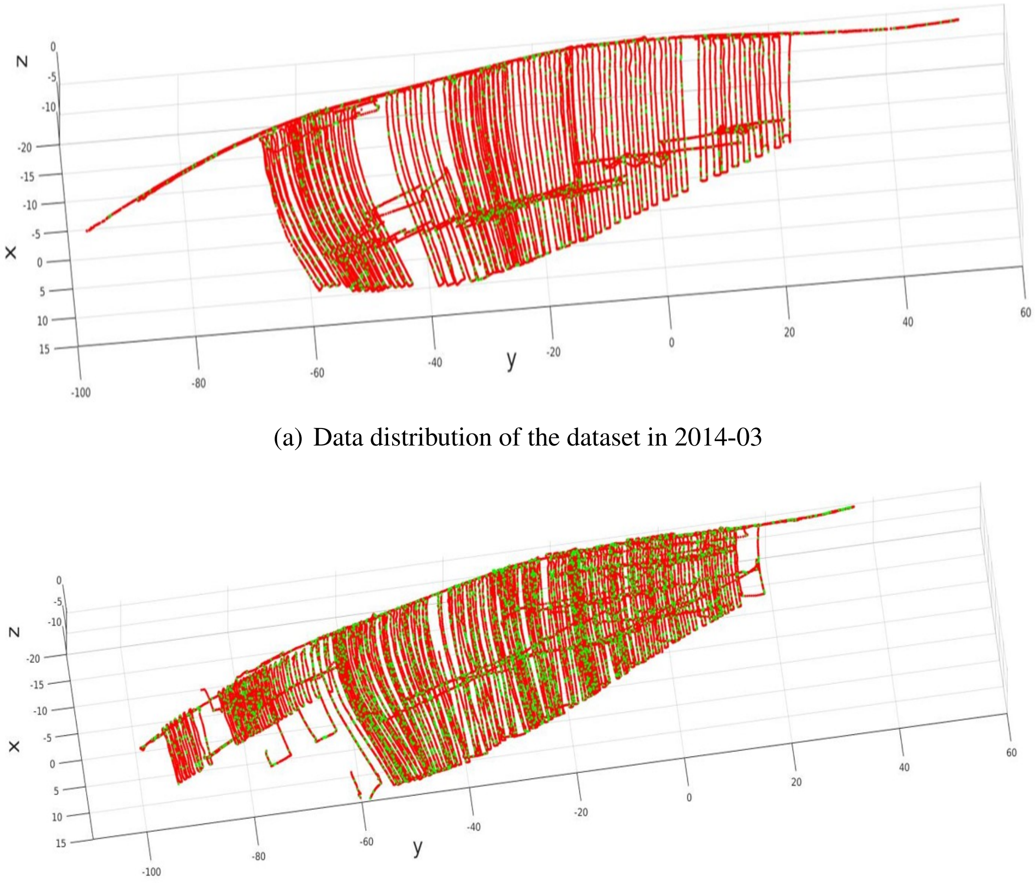
3.4.3 Training and Validation

We carry out the network training and testing on a desktop computer with 3.07GHz CPU and a Nvidia GTX Titan X. All the networks are defined and trained through Caffe [51], with 0.9 momentum, batch size 270 frames and a fixed learning rate of 10^{-4}.

The convergence trend during the training is given in Fig. 3.7. Since we include both the original GoogLeNet and AlexNet in the experiment in Section 3.5, the convergence trend of these two architectures on our dataset is also given in Fig. 3.7.

To validate the network, we test the complete network on the testing dataset of each year. The testing accuracy in network validation is shown in Fig. 3.9. The validation indicates that the network is not over-fitting to the training data. It can be seen that the test error is relatively low with respect to the sonar FOV size, as shown in Fig. 3.8.

It is important to note that this validation accuracy is not the performance of the proposed approach. We propose to use the second top layer as the feature output from this network instead of the final network output. The final output of the network is highly related to the underlying global frame of the ground truth labels, while the feature layer provides more information that can be used in a wider range of conditions. More discussion will be provided in the following section. However, the low validation error with respect to the FOV



(a) Data distribution of the dataset in 2014-03

(b) Data distribution of the dataset in 2015-06

Figure 3.6: Spatial distribution of data used in the experiment: testing samples (green), training samples (red).

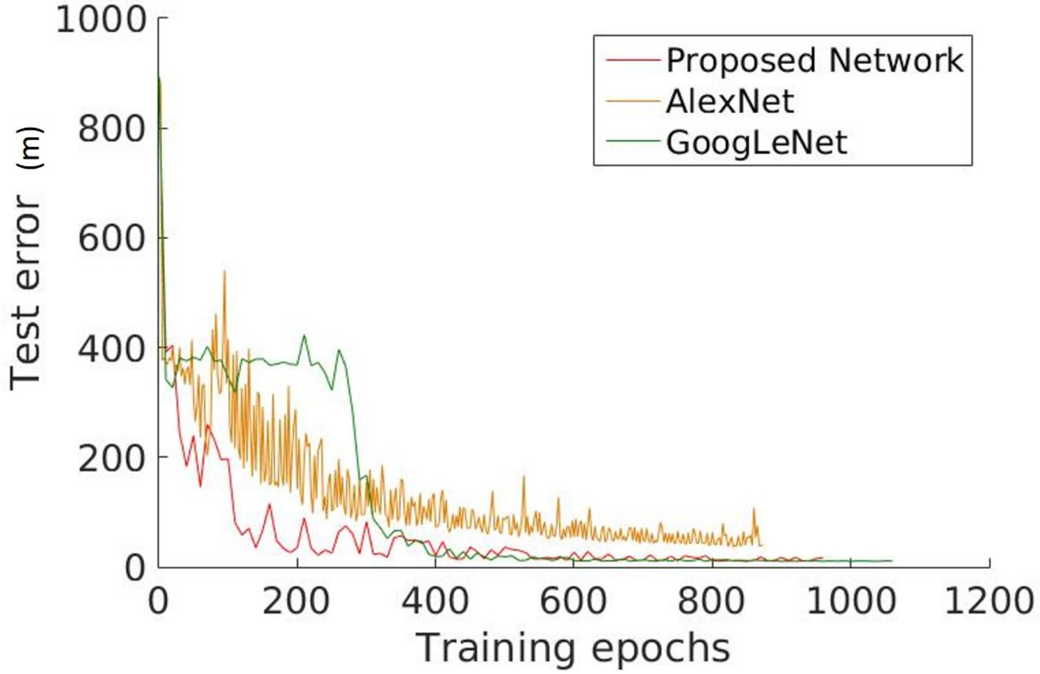


Figure 3.7: Convergence trend on dataset of 2014-03. Training convergence trend of three network architectures considered in the experiment. The original GoogLeNet and the modified version (proposed) converge to similar error as they are similar in structure. Note AlexNet which has a simpler architecture converges to a relatively higher error.

of sonar frames indicates that the feature layer contains sufficient information for vehicle position inference.

3.5 Experiment and Discussion

In this section, we analyze the proposed method as applied to two tasks: place recognition and global localization. Section 3.5.1 presents evaluation and discussion of single image-based place recognition using the proposed learned features and non-saliency score. Section 3.5.2 presents experimental results of global localization using the proposed features in a particle filtering framework. Finally, Section 3.5.3 gives test results in real-world applications, in which the trained sonar features serve as an environment observation used in the on-line multi-session SLAM registration task.

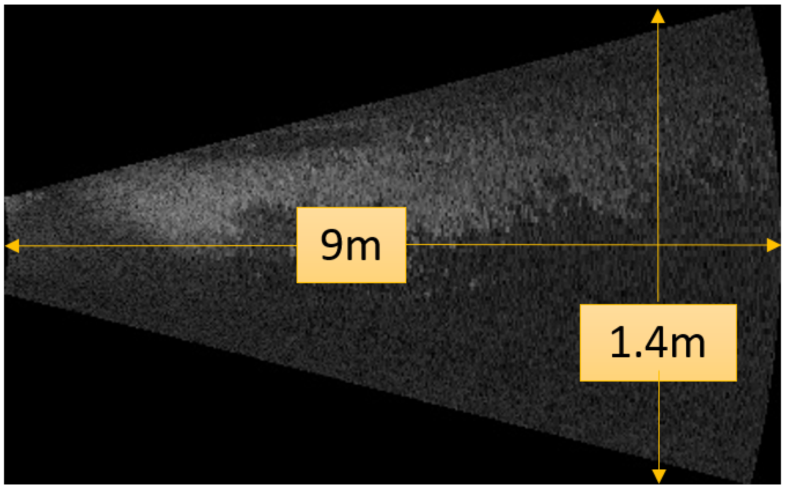


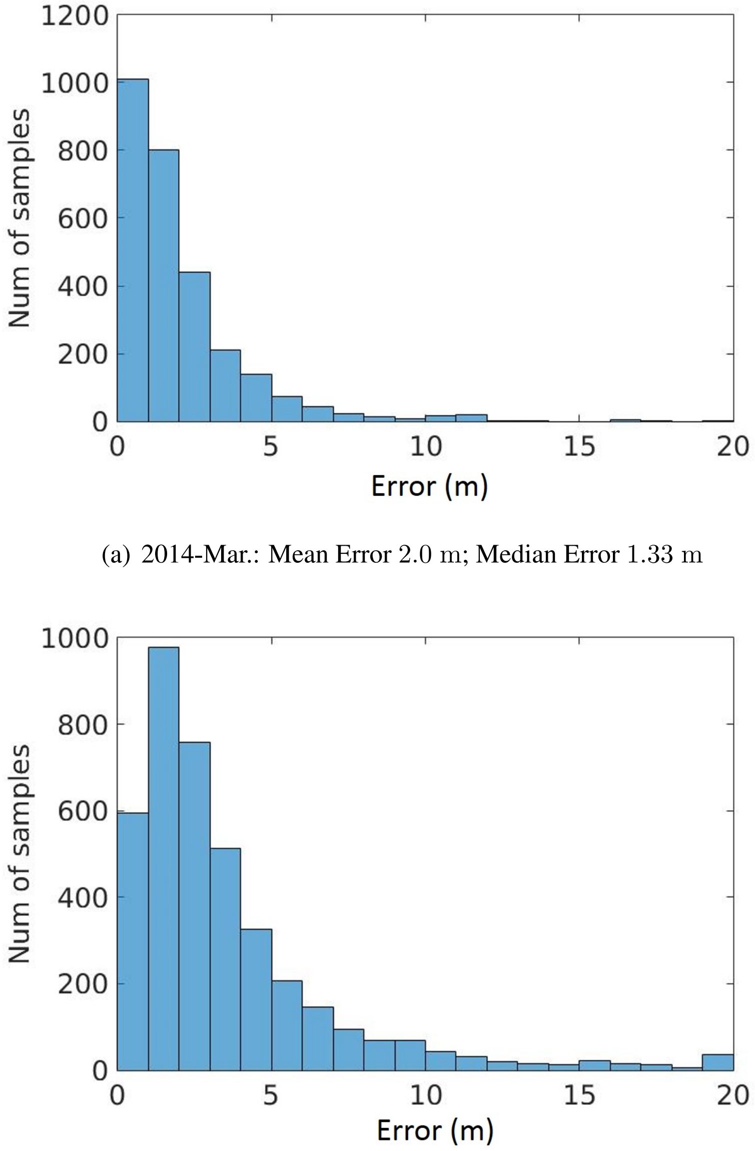
Figure 3.8: Field of view in sonar images during ship hull inspection missions.

3.5.1 Naive Place Recognition

To evaluate the performance of place recognition using the learned features, we match the features of the testing samples against the features of training samples. L^2 distance is used for similarity matching. In the following experiments, an image matching is treated as correct if the overlap between the fields of view of the two sonar frames is greater than 50%. This overlap is sufficient for most sonar image registration methods to provide a reasonable relative pose estimation. Different experimental settings are evaluated to intensively evaluate the properties of the learned feature.

First, we evaluate how the dimensions of the learned feature will affect the matching accuracy. We train the models with different numbers of feature dimensions on the training samples in 2014-03, then test the feature on the testing set of 2014-03. The precision-recall for this evaluation is shown in Fig. 3.11(a). The graph shows that when testing images and training images are close temporally, the learned feature is able to provide a very precise place recognition match. The trend shows the higher the feature dimension the higher the recognition accuracy. However, it should be noted that higher dimensional features result in higher computational costs for feature matching. So there is a trade-off between matching accuracy and computational expense when choosing a dimension for the learned feature space.

To evaluate whether the learned features can be generalized to a dataset that is dramatically different from the training samples, we use the model trained on 2014-03 and test it on the data of 2015-06. The result of this experiment is shown in Fig. 3.11(b). It can be seen that the feature is still able to provide a high matching accuracy. The trend



(a) 2014-Mar.: Mean Error $2.0\ \mathrm{m}$; Median Error $1.33\ \mathrm{m}$

(b) 2015-Jun.: Mean Error $3.12\ \mathrm{m}$; Median Error $2.26\ \mathrm{m}$

Figure 3.9: Error distribution in network validation.

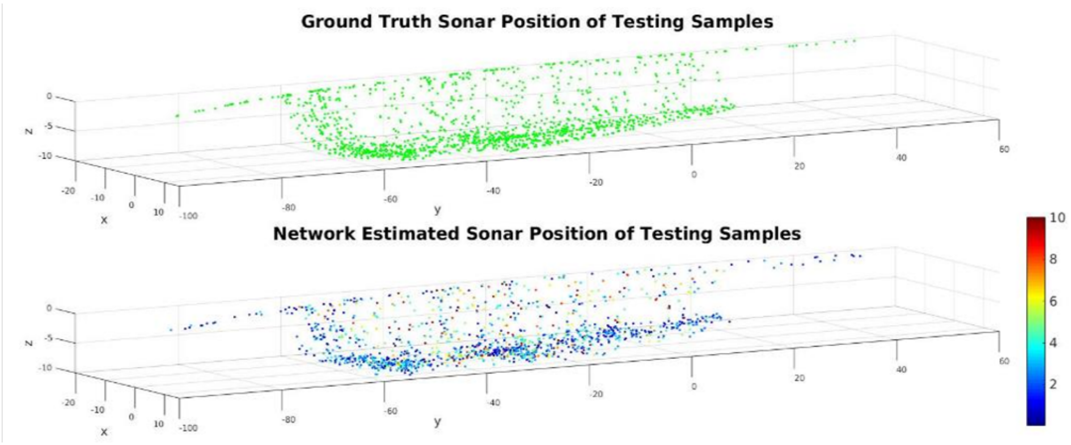
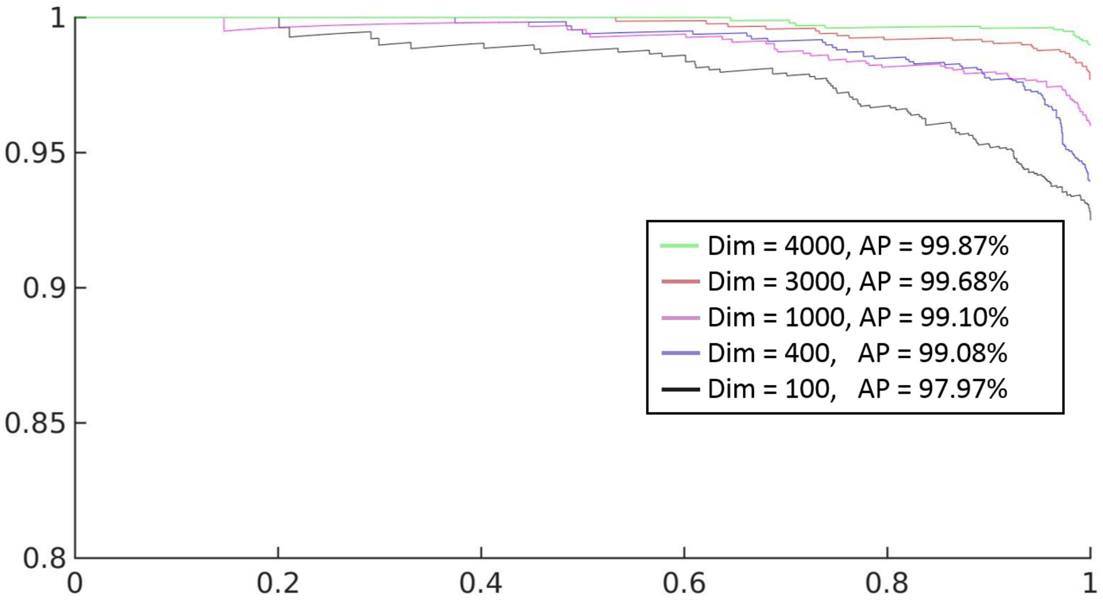


Figure 3.10: Visualization of validation result in dataset 2014-03. This figure gives a visualization of the validation result corresponding to Fig. 3.9(a). Both the ground truth positions and estimation positions in the testing set are shown. For the estimated positions, estimated error is color coded according to the color map.

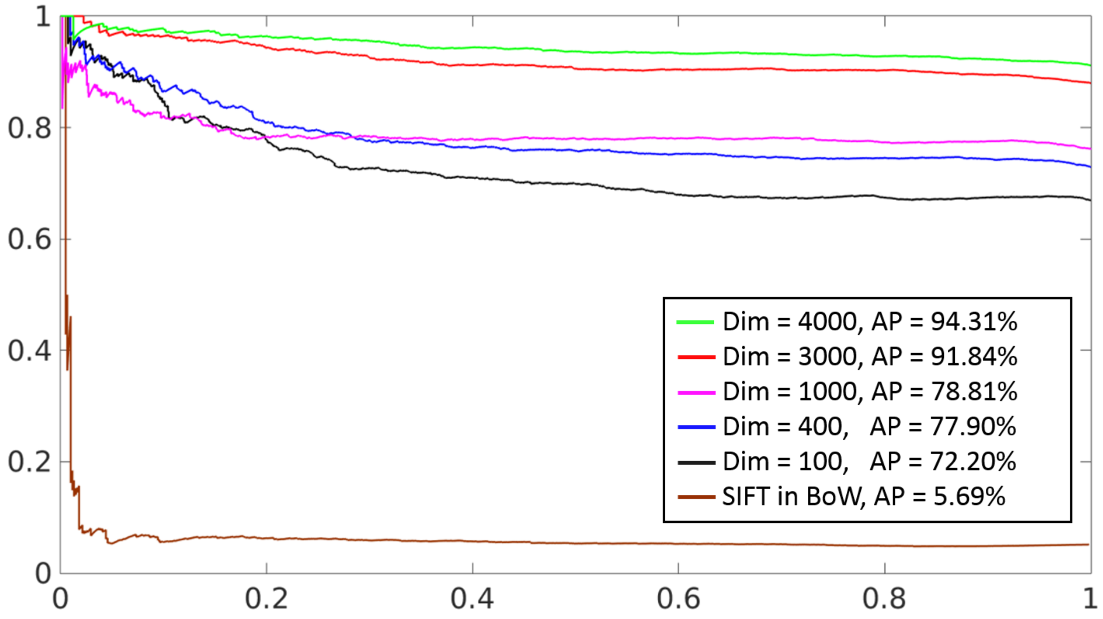
between the accuracy and the feature dimension is similar to the results on dataset 2014-03. In Fig. 3.12, we depict some examples of the matched images in this experiment with the feature dimension of 3000. It can be seen that the data set is very challenging with low texture levels in the image frames. But the trained feature is able to provide promising matching results in the dataset.

To explore the possibility of improving the feature matching accuracy by ruling out non-salient images using our learned non-saliency score, we calculate the average precision of place recognition where images with high non-saliency scores are discarded. The results are shown in Fig. 3.13. The model used in this experiment is trained on both of the training sets of 2014-03 and 2015-06. The average precision for testing sets matching against corresponding training is given. Since the average precision under the original criterion is too high to see any difference, we increase the required overlap to accept a match to 70% as compared to 50% in the other experiments. The trend in three different datasets indicates that the non-saliency score could increase the matching accuracy by filtering out images with little information. In real applications, such filtering can also help to increase computational efficiency by decreasing the number of processed images.

To evaluate how the network architecture would affect place-recognition performance, we compare the proposed network with the original GoogLeNet [92] and another state-of-the-art architecture called AlexNet [5]. We replace the output layer of these two networks with a feature layer and a final output layer to meet the objective. In this comparison experiment, the networks are trained on the training data 2014-03 and tested on the data



(a) 2014-2014



(b) 2014-2015

Figure 3.11: PR curves for place recognition testing. Models are trained on the 2014-03 training samples, and tested on (a) the 2014-03 testing samples and (b) the 2015-06 testing samples. 'Dim' is the dimension of the feature trained in the network. 'AP' is the average matching precision.

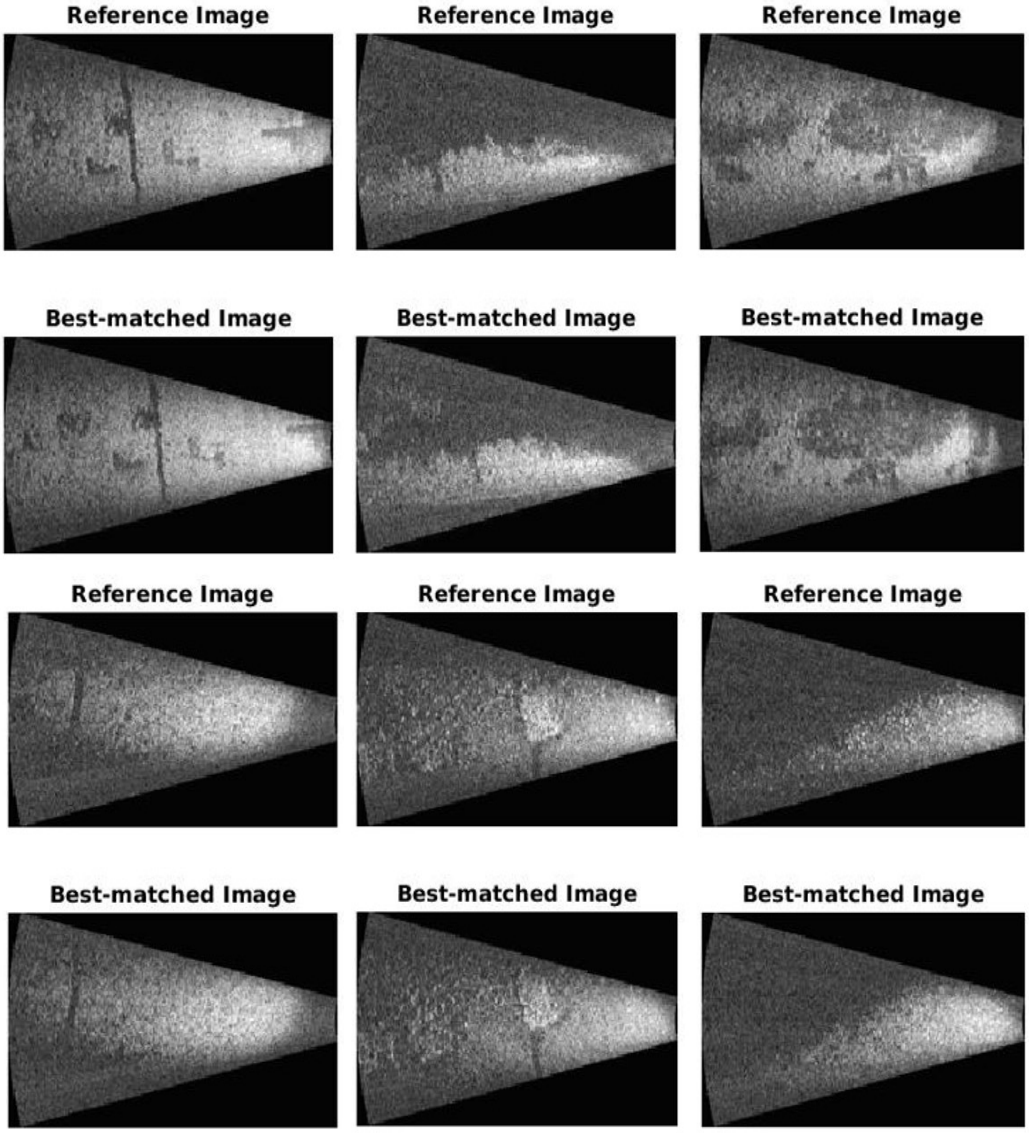


Figure 3.12: Examples of image matching results in the dataset of **2015-06**. Pairs of corresponding sonar images are detected using feature similarity. One reference image from the testing set is matched against **all** images in the training set.

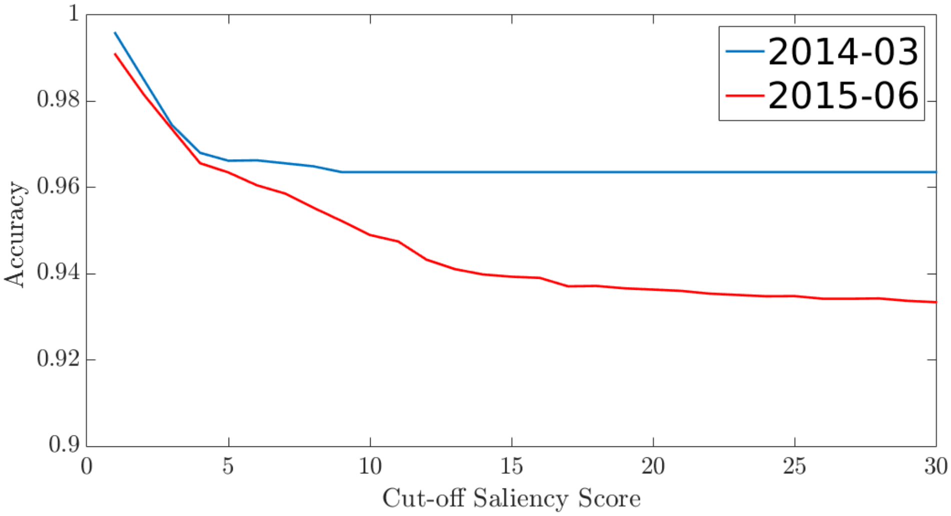


Figure 3.13: Average precision for different cut-off thresholds of non-saliency score. The model is trained on training sets 2014-03 and 2015-06.

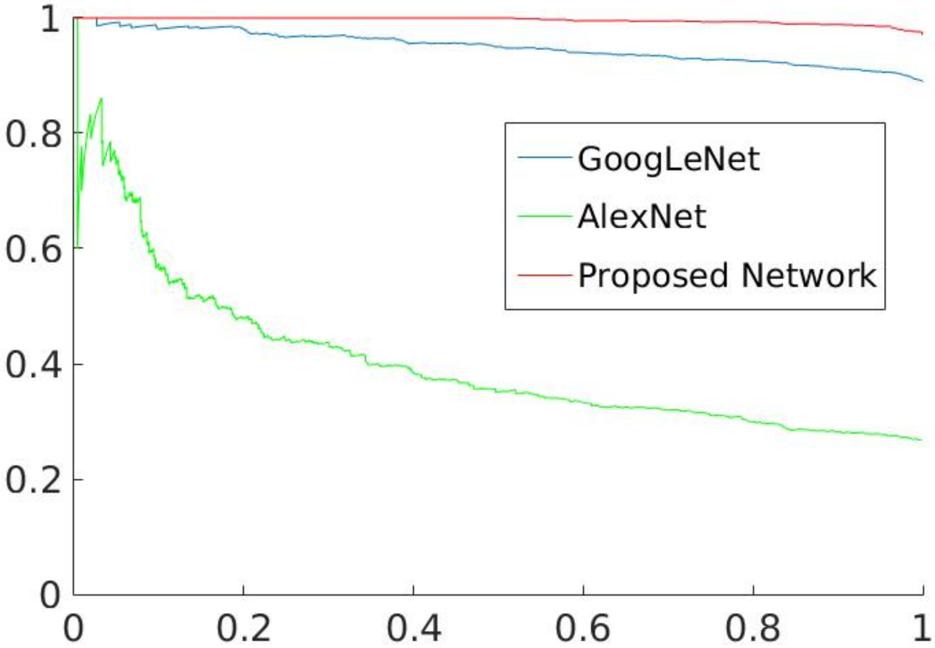
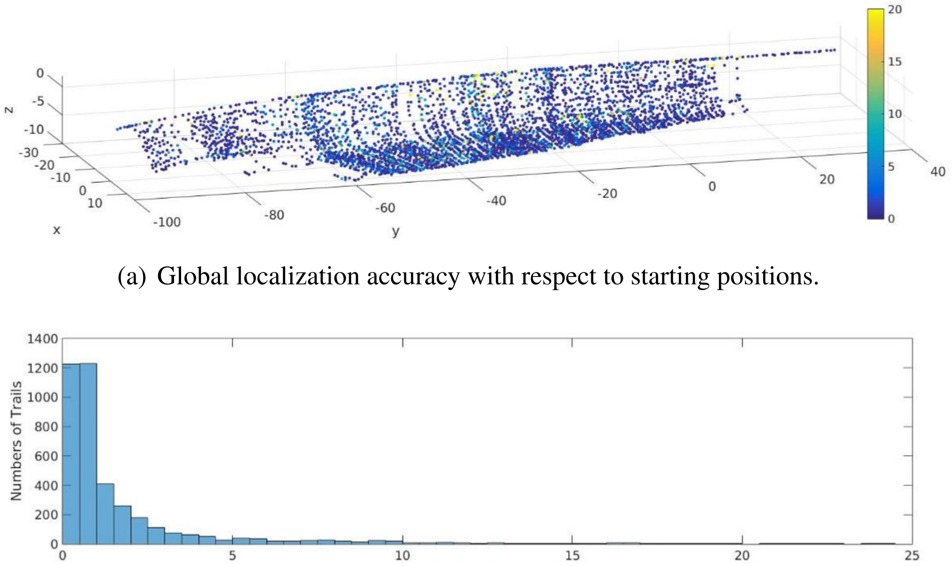


Figure 3.14: PR curve for place recognition as compared across different network architectures. The networks are trained on the training data 2014-03 and tested on the data 2015-06.



(a) Global localization accuracy with respect to starting positions.

(b) Error histogram.

Figure 3.15: Global localization Accuracy: The colors of the dots indicate the final localization accuracy of the global localization mission in 10 steps of PF. The locations of the dots indicate the locations where a mission starts.

of 2015-06. The precision-recall curve is given in Fig. 3.14. The accuracy of the proposed network and GoogLeNet are quite similar, while the AlexNet performs more poorly.

3.5.2 Global Localization

For the test of global localization, the goal is to initially localize the robot in a map with no prior information. We use particle filtering with a high number of particles to provide high-quality localization results when real-time performance is not critical. We carry out a large number of trials with random starting vehicle positions around the ship hull. The particles are evenly sampled in the map area at the beginning of each trial, then 10 iterations of particle filtering are performed. In each iteration, the weight of each particle is assigned according to the L^2 distance between current observed f_{test} and the particle's closest neighbor in the map $f^p_{closest}$.

To rule out any potential over-fitting in the localization task, the model is trained on the data of 2014-03, and the experiment is carried out on the data of 2015-06.

The final localization accuracy is given in Fig. 3.15. It can be seen from the results that the sonar features perform very well in the task of localization from a variety of starting locations. Localization error converges quickly and localization fails in only a few extreme initial starting conditions.

3.5.3 Real-world Multi-session SLAM Registration

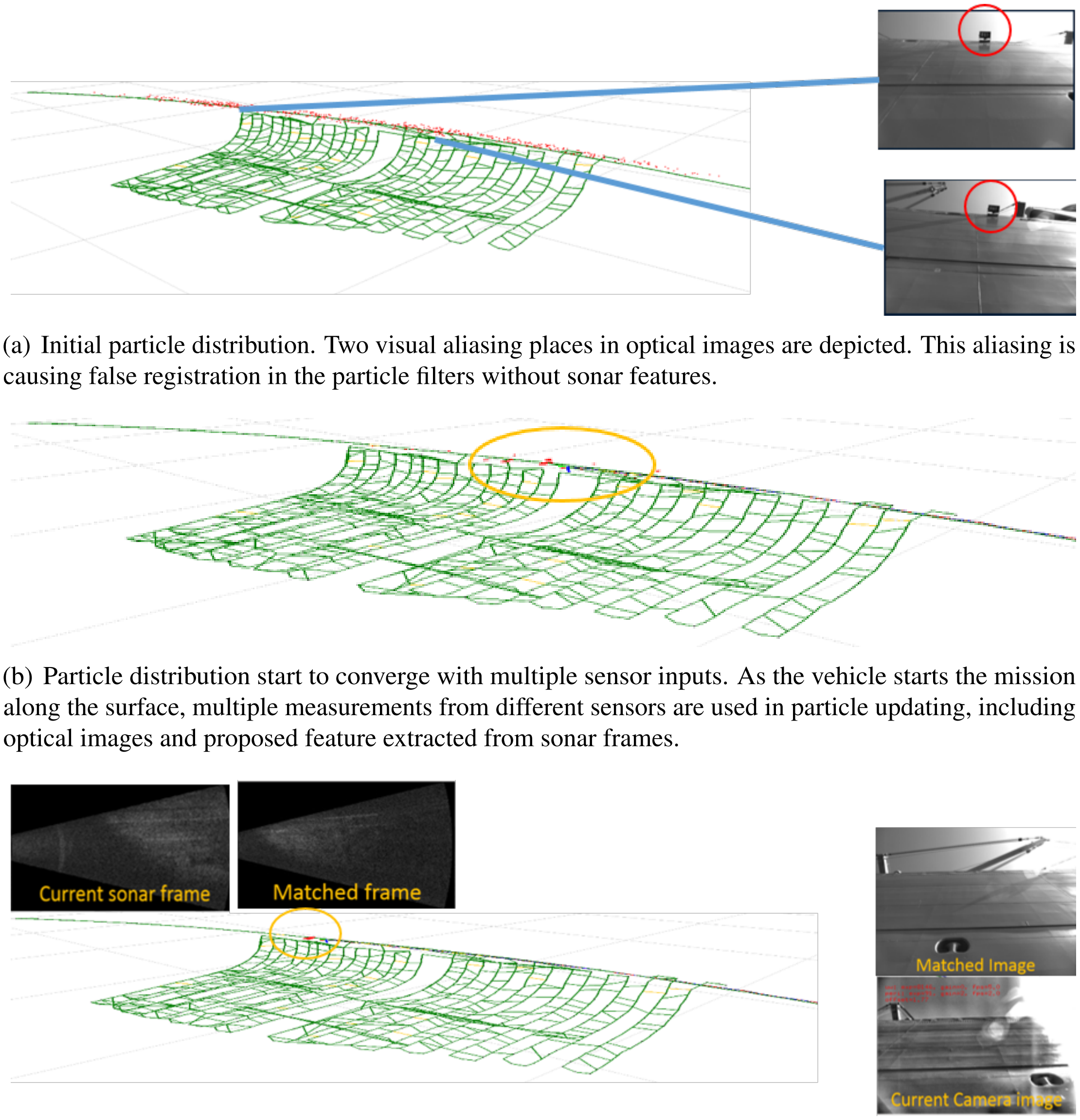
To further concretize our proposed feature in a real-world application scenario, we employ the proposed feature in an on-line multi-session SLAM registration framework.

The basic registration pipeline was developed within the HAUV project in [84] without incorporating sonar information. We use PF to estimate the probability distribution of the vehicle pose in a new SLAM with respect to a reference SLAM graph given observations from different sensors on-board: depth sensors, DVL and IMU. Since these sensors provide insufficient cues in the horizontal location, a one-versus-all image matching is conducted as a final pass within the PF believed areas to search for the best location. Before the employment of sonar information, the camera images are the only sensor measurement that provides constraint for horizontal direction. This limits the robustness of the system in that it suffers from false registration due to visual aliasing existing in the super structure on the ship. A typical example is shown in Fig. 3.16(a).

In this experiment, we improve the multi-session registration framework by incorporating the proposed method as an extra measurement for PF update. A registration example is given in Fig. 3.16. We found that since the proposed sonar feature provides extra information in both the horizontal and vertical directions, the false registration case can be easily avoided. False positive registration evidence from visual images are suppressed by sonar features.

3.6 Conclusion

In this chapter, we proposed a novel descriptive feature for FLS imagery using CNN techiniques, which can be used in underwater vehicle place recognition and localization. We also proposed a supervised-learning approach to evaluate the saliency of sonar images within the learned feature space. The approach was trained and tested on real data from a multi-year ship hull inspection project. Several important conclusions can be drawn from the experimental analysis: 1) the proposed approach can be used to provide high-accuracy global localization to a previously-seen map without any prior information; 2) the learned sonar feature can be used to carry out high-performance place recognition in real environments; 3) a pre-trained model from one dataset can be quickly fine-tuned on another dataset with reduced training time while still achieving state-of-the-art performance; and 4) the proposed saliency estimation method can increase place recognition accuracy by discarding the non-salient measurements before attempting localization.



(a) Initial particle distribution. Two visual aliasing places in optical images are depicted. This aliasing is causing false registration in the particle filters without sonar features.

(b) Particle distribution start to converge with multiple sensor inputs. As the vehicle starts the mission along the surface, multiple measurements from different sensors are used in particle updating, including optical images and proposed feature extracted from sonar frames.

(c) Particles converge at the right location which is supported by both matching camera images and sonar frames. The particle filter is able to converge at the right location without being misled by the visual aliasing shown in Fig. 3.16(a)

Figure 3.16: Real-world example of sonar feature prone to suppress false registration due to visual aliasing in optical image using multi-session SLAM registration in HAUV navigation system.

CHAPTER 4

Utilizing High-dimensional Features in Real-time Bayesian Filtering

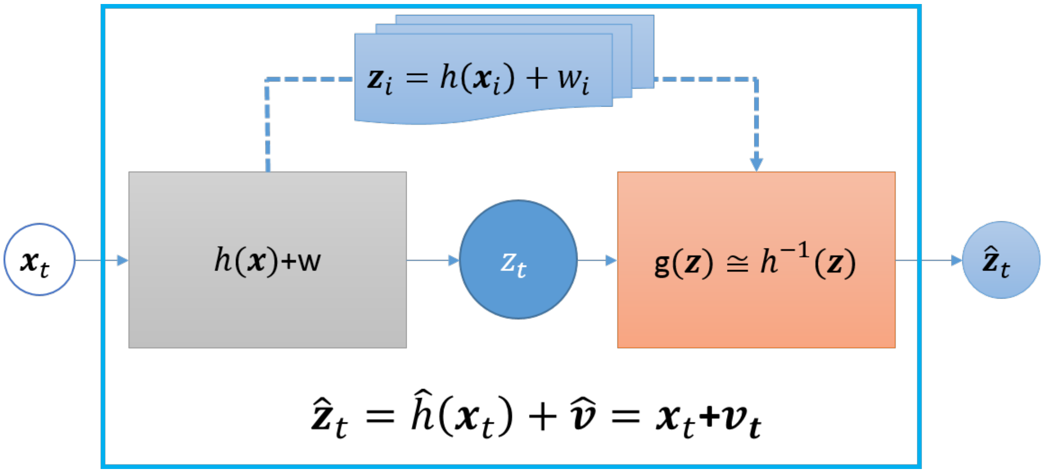
4.1 Introduction

The sonar feature learning framework introduced in Chapter 3 indicates the great potential of feature learning using CNNs in underwater mobile robotics.

However, current utilization of deep learning techniques in the robotics community has been primarily limited to topological or discrete problems, such as place recognition or semantic labeling. In the previous chapter, we have showed that CNNs can be used to train strong image-level high-dimensional features that can be used in a topological application: place recognition. In this chapter, we explore the possibility of leveraging these types of features in metric and recursive problems, which involves the estimation of continuous states in dynamic systems; potential applications include robot localization and stochastic control. Probabilistic filtering frameworks have been developed to address these types of problems. Such approaches have served as the backbone of modern robotics for the last two decades [94]. However, utilizing learned features as system observations in a probabilistic filtering framework, such as Bayesian filtering, is non-trivial due to the following factors:

1. **High dimensionality**: The dimensionality of the learned features with great success in the literature is typically high, which results in large computational complexity for filtering approaches such as the extended Kalman filter (EKF) or the unscented Kalman filter (UKF) as each requires a matrix inversion whose computational complexity is higher than quadratic in the dimension.

2. **Expense of evaluation**: Most current deep learning networks in robotics applications are evaluated on high dimensional inputs such as camera images or 3D laser scans. To evaluate the mapping between an unknown high dimensional function (for example a



$\mathbf{z}_i = h(\mathbf{x}_i) + w_i$
$\mathbf{x}_t$
$h(\mathbf{x})$+w
z_t
$g(\mathbf{z}) \cong h^{-1}(\mathbf{z})$
$\hat{\mathbf{z}}_t$
$\hat{\mathbf{z}}_t = \hat{h}(\mathbf{x}_t) + \hat{\mathbf{v}} = \mathbf{x}_t + \mathbf{v}_t$

Figure 4.1: Observation pre-linearization. $\mathbf{x}_t$ is the estimation state and $\mathbf{z}_t$ is the original observation. The knowledge of the original observation model $\mathbf{z}_t = h(\mathbf{x}_t)$ is limited to evaluation at certain points. We estimate the local inverse function of $h()$, resulting in a new observation $\hat{\mathbf{z}}_t$ with an approximately direct linear observation model $\hat{\mathbf{z}}_t \simeq \mathbf{x}_t + \mathbf{v}_t$.

.

CNN) and the system variable of interest at any given query point can be expensive to achieve or approximate. For example, to evaluate an image-based CNN model at an un-visited position requires a synthetic or approximated image for network input.

This results in prohibitively-large computational complexity for sample-based filters such as the PF or UKF, as each requires an evaluation for each sample location.

3. **Lack of a Noise Model**: Uncertainty models are readily available for simple learning algorithms—for example, Bayesian linear regression provides a straightforward estimation of prediction error. On the other hand, the black-box nature of many modern algorithms (e.g. CNNs) do not give an explicit framework for reasoning about uncertainty, yet this uncertainty model is absolutely necessary when plugged into probabilistic frameworks such as Kalman Filtering (KF).

The above properties of high-dimensional learned features make them unsuitable for direct use as observations in standard real-time filtering frameworks.

In this chapter, we propose an approach that we call pre-linearization, which transforms a black-box high-dimensional observation function into a new form that can be trivially integrated into a traditional Bayesian filtering approach. As depicted in Fig. 4.1, we achieve this aim by linearizing an arbitrary observation using its approximated local inverse function. The new resulting observation is both *linear* w.r.t the system's estimated state and has the *same dimension* as the estimated system state (an upper-bound for the theoretical dimensionality reduction in the system).

In addition to the efficiency gains of this pre-linearization process, we also present a technique to measure the uncertainty of the processed observation using *data perturbation*. This technique has shown great promise in several fields, including machine learning [16], optimization [79], and data privacy [21], to enable the understanding of functions without direct knowledge of the underlying transformation being performed.

To concretize the application of the proposed method, we address the traditional mobile robotics problem of localization and demonstrate our approach in the application of underwater real-time localization using CNN learned features from FLS images in a real-time KF. We selected this domain and modality as it represents the challenging areas, where no hand-crafted features can be used easily in a real-time framework without strict assumptions to the scene structure, as we discussed in Section 1.3.3. We will show how all the benefits of a Bayesian filtering approach can be garnered, including physically interpretable uncertainty ellipses and innovation gating of new measurements, while using a CNN learned high-dimensional feature representation that greatly outperforms traditional hand-designed features.

The contributions of this chapter are represented in our publication [67], which include:

- An efficient pre-linearization process that transforms an arbitrary high-dimensional observation into a new linear observation with constrained dimensions.
- An uncertainty measurement approach for the pre-processed observation model using data perturbation.
- The first real-time localization solution utilizing an imaging sonar to perform underwater localization using a CNN learned feature representation without further assumption to the scene structure.

4.1.1 Outline

This chapter is laid out as follows: Section 4.2 gives a brief background introduction to three topics considered in this work: limitation in Bayesian filtering, uncertainty measurement in CNNs and data perturbation techniques. Section 4.3 provides a description of the proposed approach to construct a linear observation from an arbitrary observation function in a principled manner. Section 4.4 introduces how we apply the proposed approach using an imaging sonar to better localize an underwater robot using a CNN learned feature representation. Section 4.5 evaluates the proposed method in a real-time KF localization system using CNN learned features from FLS sonar images. In this section, we also provide comparison of the proposed pre-linearization method to other traditional approaches of using abstracted features utilization. Section 4.6 provides evaluation on real-time localization tasks using CNN mid-layer features from camera images to further validate the proposed method, and to show generalization capability of our proposed algorithm. Section 4.7 summarizes the chapter and discusses future work.

4.2 Background

4.2.1 Restriction on Observations Used in Bayesian Filtering

Bayesian filtering, which has been the primary solution to these problems for decades, provides a probabilistic framework incorporating a variety of observations in a dynamic system. Most widely-used filters share the typical Gaussian noise assumption. Widely-used methods include KF [104], EKF [31] and UKF [55].

The curse of dimensionality has long been one of the main challenges that prevent Bayesian filtering from being used in large scale systems with high dimensionality in model functions. In addition, the high cost in computational complexity, the high non-linearity that often comes along with high dimensionality, could result in a high approximation error in the linearization procedure shared by a lot of Bayesian filtering approaches. In the context of robotics estimation problems, where real-time is a crucial property of the system, this limitation becomes more significant.

For methods like KF and EKF where matrix inversion is necessarily involved, time complexity can be higher than quadratic. It is $O(n^{2.37})$ using an optimized inversion solver. Although there have been efforts made to reduce the computational complexity for these methods [7], the requirement that the observation model be fully defined also adds a strong restriction on the flexibility of these methods.

UKF only relies on numerical evaluation of the observation function. The computational cost quickly precludes real-time use as the evaluation at sigma points is expensive as discussed in Section 4.1. Given that the observation function is known analytically, the propagation of the filters above involve matrix inversion in the dimension of observation states, which is far from real-time performance in high dimensional cases.

PF or Sequential Monte Carlo (SMC) is the most friendly method to the partial defined and high-dimensional observations within the Bayesian filtering family. However, similar to UKF, a PF approach requires evaluation at each particle and real-time constraints that severely limit the number of particles that can be used in practice, which might result in undesired collapsing of PF as discussed in [14].

4.2.2 Uncertainty Measurement in CNN

CNNs have achieved great success in a broad range of fields, including robotics. Chen et al. [22] proposed one of the pioneer place recognition systems based on a feature generated using mid-level layers of a pre-trained CNN. Sünderhauf et al. [91] provide an extensive evaluation on the usage of mid-layers of pre-trained models for the task of visual place recognition. Gomez-Ojeda et al. [36] address the issue of severe appearance change in the place recognition problem by training an appearance robust network over a high variant visual-image dataset. In Chapter 3, we also show that a CNN can be used in feature learning for challenging modalities such as FLS.

However, most current work is applying CNN techniques to place recognition or topology estimation problems. The state spaces of those problems are often discrete, where an explicit measurement model is not required and a strict uncertainty measurement is less crucial. In this work, we examine the less-often-addressed metric and recursive estimation problems in continuous state spaces, where a more strict knowledge of the measurement model and an explicit uncertainty quantification are often required. Examples of such applications include mobile robot localization for navigation systems and pose estimation in stochastic control.

In addition to the issues around observation models discussed in the previous subsection, another important and un-addressed problem is the formulation of a noise model for such features. Dropout layers [43] are considered in some recent work as a promising direction to provide uncertainty measurement for CNNs [34, 57]. Here in this chapter, we propose to work around this problem and provide an observation uncertainty measurement using data perturbation. This method captures a similar idea to dropout layers in CNNs while no extra training is needed and existent pre-trained models can be used directly.

4.2.3 Data Perturbation

Underlying this technique is a growing body of work that supports the use of input perturbations as a form of regularization for learning and estimation problems. The key idea is that the addition of noise into our data allows for an understanding of the stability of the output of the function.

Perturbation methods typically involve repeating the optimization or decision procedure a number of times, with different noisy samples on each run. When the (randomized) outputs are averaged together, a la a *bootstrap* approach, the resulting average can be considered a *regularized solution* to the original problem. In fact this is almost precisely the design behind the Random Forest algorithm [69], where the "perturbation" of the data is via a random subselection of the input features. But the connection between perturbation and regularization becomes even more explicit in the case of Least Squares Regression. It was shown by Bishop [16] that minimizing squared loss with the introduction of noise is equivalent to so-called Tikhonov regularization input noise; in the case of Gaussian noise one obtains the usual $L2$ regularization. It should be noted that the latter result has often been the central argument in favor of "drop-out" tools for training deep neural networks [95, 97].

Another benefit of the perturbation method for learning problems is that it provides a simple and efficient replacement for a full generative model. In many scenarios one has a natural generative model associated with a learning procedure, but there is a growing trend towards using essentially black-box tools such as CNNs as part of a larger pipeline (as in the case of the present work) where there is not a canonical probabilistic model to reason about posteriors, likelihoods, etc.

Perturbation techniques have also found their way into many aspects of learning but also in other applications. One of the seminal results in online learning by Kalai and Vempala involves an algorithm they coined *Follow the Perturbed Leader* [56]. There have been several works that argue for the use of so-called stochastic smoothing for optimization, where a noisy version of the objective function is optimized, as this can make the problem more robust and efficient to solve [28]. Finally, there has been a lot of recent work on using added noise to protect data privacy, in a growing area known as differential privacy [30].

4.3 Observation Pre-linearization

In this section, we introduce the proposed algorithm that transforms the original observation into one that can be used efficiently by Bayesian filters. To begin with, we give a brief overview of general Bayesian filters and specify the notation that will be used in Section 4.3.1. Then we describe, in detail, the proposed pre-linearization approach of arbitrary observation features in Section 4.3.2. Finally, we introduce a uncertainty measurement estimation approach to the generated observations model from the pre-processing step in Section 4.3.3.

4.3.1 Problem Statement for General Bayes filter

Recursive Bayesian estimation, also known as a Bayes filter, is a general probabilistic approach for estimating unknown states in a Markov process using observed states and known control inputs. The underlying model for a Bayes filter is given by Fig. 4.2(a). Considering a time-invariant system with additive noise, the system can be defined by a stochastic discrete-time state space dynamic (motion) model

$$\mathbf{x}_t = \mathrm{f}(\mathbf{u}_t, \mathbf{x}_{t-1}) + \mathbf{w}_t, \tag{4.1}$$

and the stochastic observation (measurement) model

$$\mathbf{z}_t = \mathrm{h}(\mathbf{x}_t) + \mathbf{v}_t, \tag{4.2}$$

where $\mathbf{x}_t \in \chi$ is the true state of an unobserved Markov process that needs to be estimated, $\mathbf{u}_t$ is control input to the system, and $\mathbf{z}_t$ is the observed state from the hidden Markov model. $\mathbf{w}_t$ and $\mathbf{v}_t$ are noise in the system, which is often modeled as independent Gaussians with mean $\mathbf{0}$ and covariance $\mathbf{Q}$ and $\mathbf{R}$.

The goal of the Bayes filter is to estimate the probability distribution of $\mathbf{x}_t$, given all the control inputs $\mathbf{u}_{1:t}$ and observation states $\mathbf{z}_{1:t}$, in a recursive way:

$$\mathrm{bel}(\mathbf{x}_t) := p(\mathbf{x}_t | \mathbf{u}_{1:t}, \mathbf{z}_{1:t}), \tag{4.3}$$

$$\begin{aligned} \mathrm{bel}(\mathbf{x}_t) &= \int p(\mathbf{x}_t | \mathbf{u}_t, \mathbf{z}_t, \mathbf{x}_{t-1}) \, \mathrm{bel}(\mathbf{x}_{t-1}) d\mathbf{x}_{t-1} \\ &= \eta p(\mathbf{z}_t | \mathbf{x}_t) \int p(\mathbf{x}_t | \mathbf{u}_t, \mathbf{x}_{t-1}) \, \mathrm{bel}(\mathbf{x}_{t-1}) d\mathbf{x}_{t-1}. \end{aligned} \tag{4.4}$$

The main task of the Bayes filter can be decomposed into a **prediction** step

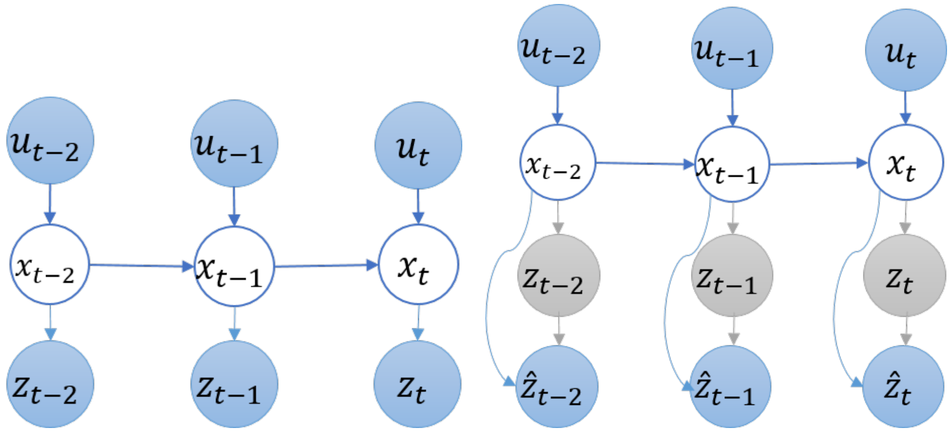
$$\bar{\mathrm{bel}}(\mathbf{x}_t) := \int p(\mathbf{x}_t | \mathbf{u}_t, \mathbf{x}_{t-1}) \, \mathrm{bel}(\mathbf{x}_{t-1}) d\mathbf{x}_{t-1}, \tag{4.5}$$

and a **correction step**

$$\mathrm{bel}(\mathbf{x}_t) = \eta p(\mathbf{z}_t | \mathbf{x}_t) \bar{\mathrm{bel}}(\mathbf{x}_t). \tag{4.6}$$

Based on the assumption of an independent Gaussian noise model, the resulting $\bar{\mathrm{bel}}(\mathbf{x}_t)$ and $\mathrm{bel}(\mathbf{x}_t)$ are both Gaussian distributions given by $\bar{\mathrm{bel}}(\mathbf{x}_t) \sim N(\bar{\mu}_t, \bar{\Sigma}_t)$ and $\mathrm{bel}(\mathbf{x}_t) \sim N(\mu_t, \Sigma_t)$.

To achieve $\mathrm{bel}(\mathbf{x}_t)$, the observation model defined in Equation 4.2 is required to be known in either analytical or numerical means, which can be unavailable or hard to achieve as discussed in Section 4.1. Even if the observation model is known, the high dimensionality could incur high computational expense. In the case of the KF and its nonlinear variants,



(a) Markov model in Bayes Filter.

(b) Equivalent Markov model when pre-linearization process is utilized.

such as the EKF and the UKF, $\mathrm{bel}(\mathbf{x}_t)$ is estimated using a best linear estimator as follows:

$$\mu_t = \bar{\mu}_t + \bar{\Sigma}_t H^T S^{-1}(\mathbf{z}_t - \mathrm{h}(\bar{\mu}_t)), \tag{4.7}$$

$$\Sigma_t = \bar{\Sigma}_t - \bar{\Sigma}_t H^T S^{-1} H \bar{\Sigma}_t, \tag{4.8}$$

where H is the Jacobian matrix of the observation function and S is the residual covariance of the observation. Regardless, the method to approximate H and S in different filters all require a matrix inversion of the dimension of both $\mathbf{z}$ and $\mathbf{x}$.

Thus, in the direct use of a high-dimensional observation model, the filter correction step is often too expensive for real-time applications. To address these issues and make efficient use of high-dimensional features in a general Bayes filter, we propose the pre-linearization process below.

4.3.2 Pre-linearization

Inspired by the idea of pre-whitening, we propose a pre-processing step for an arbitrary high-dimensional observation with no explicit evaluable observation model, so that it can be used efficiently in a filtering approach. As shown in Fig. 4.1, we propose a new observation defined by:

$$\hat{\mathbf{z}}_t = \mathrm{g}(\mathbf{z}_t), \tag{4.9}$$

where $\mathrm{g}()$ is designed to approximate the local inverse function of original observation $\mathrm{h}()$. Although the method requires no analytical knowledge of the observation function $\mathbf{z} = \mathrm{h}(\mathbf{x})$, two basic criteria have to be met:

1. **Sufficient Numerical Knowledge**: The observation function $h()$ is known numerically at some discrete points with enough density, if not known everywhere:

$$\forall \mathbf{x} \in \chi, \quad \exists \mathbf{x}_0 \in \chi \& ||\mathbf{x}_0 - \mathbf{x}|| < \sigma \quad \text{s.t.}$$

$$\mathbf{z}_0 = \mathrm{h}(\mathbf{x}_0) \text{ can be directly evaluated.}$$

   σ defines the requirement of density. In the case of learned features from an image, σ should represent a field of view captured by the camera.

2. **Correlated Observation**: To infer x from z, similarity in z should imply similarity of x at least locally:

$$\exists \epsilon > 0, for \quad \forall \mathbf{x}_1, \mathbf{x}_2 \in B(\mathbf{x}_0; \epsilon)$$

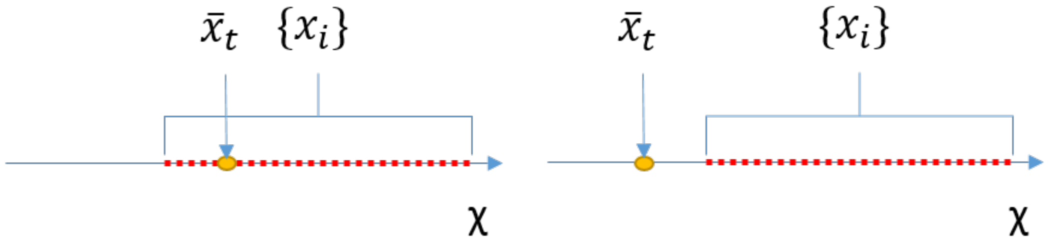
$$||\mathbf{z}_1 - \mathbf{z}_0|| > ||\mathbf{z}_2 - \mathbf{z}_0|| \Longleftrightarrow ||\mathbf{x}_1 - \mathbf{x}_0|| > ||\mathbf{x}_2 - \mathbf{x}_0||$$

   This criteria also implies that the function is locally invertible which holds for most high-dimensional learned features. Later in Section 4.3.3 we will discuss how this criteria can be adjusted by the uncertainty measurement. Section 4.3.3 discusses how uncertainty measurement can compensate for potential errors the more the underlying observation function deviates from these assumptions.

Given a feasible observation $\mathbf{z}$ that meets the assumptions, the pre-linearization function can be generated as described in Algorithm 3. We construct $\hat{\mathbf{z}} = \mathrm{g}(\mathbf{z})$ to fit the inverse

Algorithm 3 Pre-linearization function

```
Input:
$\mathbf{z}_t$: Original observation value
$B(\bar{\mu}_t) \subset \chi$: Confidence area given $\overline{\text{bel}}(\mathbf{x}_t)$.
Output:
$\hat{\mathbf{z}}_t$: linearized observation value.
$isValid$: Is the output $\hat{\mathbf{z}}_t$ valid or not.
$S_t := \{(\mathbf{z}_j, \mathbf{x}_j)$ : known evaluable points within $B(\bar{\mu}_t)$
if $\bar{\mathbf{x}}_t \notin \text{ConvexHull}(\{\mathbf{x}_j\})$ then
    $\text{Return}(isValid = 0;\ \hat{\mathbf{z}}_t = \{\})$
end if
for $j = 1$ to $|S_t|$ do
    $\alpha_j := \text{Sim}(\mathbf{z}_t, \mathbf{z}_j)$
end for
$\hat{\mathbf{z}}_t := \Sigma_j \frac{\alpha_j}{\sum(\alpha_i)} \cdot \mathbf{x}_j$
$\text{Return}(isValid = 1;\ \hat{\mathbf{z}}_t)$
```



(c) valid case

(d) non-valid case

Figure 4.2: The knowledge of $h(\cdot)$ is bounded by sample set S: If S is a bounded set, then the knowledge field of $h(\cdot)$ is bounded by S. In 1-D case, if $\bar{\mu}_t$ falls outside of the bounds of $\{\mathbf{x}_i\}$, the estimation of $h_z^{-1}(\cdot)$ based on S is biased.

observation $h^{-1}()$ in a local area defined by $B(\bar{\mu}_t)$:

$$\hat{\mathbf{z}}_t = \mathrm{g}(\mathbf{z}_t) = \Sigma_j w_j(\mathbf{z}_t) h^{-1}(\mathbf{z}_j) \sim h^{-1}(\mathbf{z}_t), \tag{4.10}$$

$$w_j(\mathbf{z}_t) = \frac{\mathrm{Sim}(\mathbf{z}_j, \mathbf{z}_t)}{\Sigma_i \mathrm{Sim}(\mathbf{z}_i, \mathbf{z}_t)}, \tag{4.11}$$

where $\mathrm{Sim}(\mathbf{z}_1, \mathbf{z}_2) > 0$ can be any similarity measurement, such as norm defined in the space of observation states. An evaluatability detection is also conducted in Algorithm 3 returning a logical value $isValid$. This step detects whether the currently estimated location $\bar{\mu}_t$ falls outside of the bounds defined by all the evaluatable points of $\mathrm{h}()$, which might result in a highly biased estimation of $\mathrm{g}()$. A demonstration of the 1-D case is shown in Fig. 4.2.

Given the pre-linearization procedure $\hat{\mathbf{z}}_t = \mathrm{g}(\mathbf{z}_t)$, the resulting linearized observation $\hat{\mathbf{z}}_t$ is linear with respect to the state $\mathbf{x}_t$:

$$\hat{\mathbf{z}}_t = \mathrm{g}(\mathbf{z}_t) \sim h^{-1}(\mathbf{z}_t) = h^{-1}(\mathrm{h}(\mathbf{x}_t)) = \mathbf{I} \cdot \mathbf{x}_t. \tag{4.12}$$

This observation function can now be directly used in a KF. The underlying graphical model of the overall estimation problem is shown in Fig. 4.2(b).

Some advantages of using $\hat{\mathbf{z}}_t$ over $\mathbf{z}_t$ are readily apparent:

- High Dimension vs Low Dimension: Since $Dim(\hat{\mathbf{z}}_t) = Dim(x_t)$, the computational complexity for estimating $\hat{\mathbf{z}}_t$ and involving it into the filter is fixed and the dimensionality equals the lower bound of the filtering system ($Dim(x_t)$).

- Nonlinear vs Linear: $\hat{\mathbf{z}}_t$, on the other hand, $\hat{\mathbf{z}}_t = \mathrm{g}(\mathbf{x}_t)$ is constructed to be a linear observation model. If $h(\cdot)$ is linear, the usage of either $\hat{\mathbf{z}}_t$ or $\mathbf{z}_t$ will degenerate to a linear estimation problem. Similar performance should be expected. If $\hat{\mathbf{z}}_t$ is highly non-linear, high estimation error could be expected during the linearization approximation in a lot of Bayesian filtering methods.

4.3.3 Uncertainty Measurement

We measure the uncertainty of the linearized observation $\hat{\mathbf{z}}_t$ by fitting a Gaussian model:

$$p(\hat{\mathbf{z}}_t|\mathbf{x}_t) \sim N(\bar{\mu}_{\hat{\mathbf{z}}}, \Sigma_{\hat{\mathbf{z}}}). \tag{4.13}$$

The mean value $\bar{\mu}_{\hat{\mathbf{z}}}$ is given by Algorithm 3. We estimate $\Sigma_{\hat{\mathbf{z}}}$ leveraging a data perturbation technique that treats the overall observation function as a black box, as depicted in Fig. 4.3. The idea of data perturbation is to vary the input data to an unknown system, and analyze the system stability and uncertainty by looking at the corresponding outputs. In our case, input data corrupted by noise is fed into the function, and the uncertainty of the system is estimated statistically from the output:

$$\Sigma_{\hat{\mathbf{z}}_t} = \frac{1}{N-1}\Sigma_j(\hat{\mathbf{z}}_t^j - \bar{\mu}_{\hat{\mathbf{z}}})(\hat{\mathbf{z}}_t^j - \bar{\mu}_{\hat{\mathbf{z}}})^T, \tag{4.14}$$

$$\hat{\mathbf{z}}_t^j = \hat{\mathrm{h}}_{\mathrm{t}}(\mathbf{x}_t; \mathbf{z}_t^j, B(\bar{\mu}_t^j)), \tag{4.15}$$

where $\mathbf{z}_t^j$ and $\bar{\mu}_t^j$ are perturbed values from original system input $(\mathbf{z}_t, \bar{\mu}_t)$. N is the total number of perturbation points.

In extreme cases, when the original observation function, $\mathbf{z}_t = \mathrm{h}(\mathbf{x}_t)$ is not locally invertible, for example $\mathbf{z} = \mathrm{h}(\mathbf{x}) \equiv \mathbf{C}$, the new observation value $\hat{\mathbf{z}}_t$ is only determined by $\bar{\mu}_t^j$, which results in a wide Gaussian model in the estimation procedure.

It is important to note that the proposed uncertainty measurement features a desired property, that it is environment dependent. It not only encodes uncertainty coming from the observation feature itself, but also the uncertainty caused by uniqueness of the feature or potential aliasing with respect to the environment. We show that this property provides a more accurate uncertainty modeling to the estimation state in the experiment results discussed in Section 4.5.

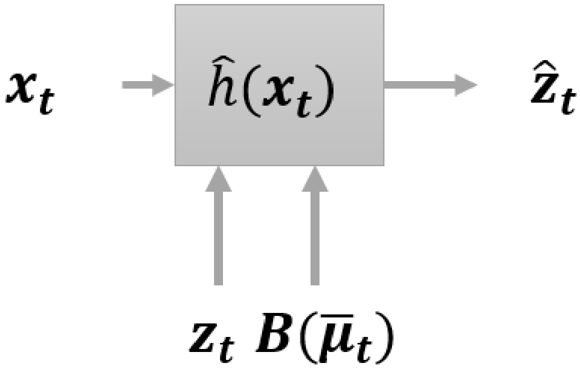


Figure 4.3: Data perturbation approach for black box noise model estimation.

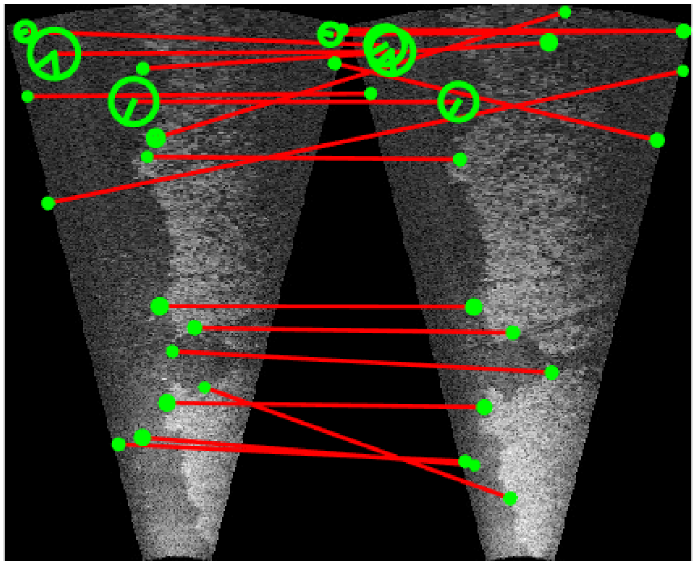


Figure 4.4: Example of traditional visual features (SIFT) in sonar image matching.

4.4 Real-time Sonar-based Underwater Localization Using CNN Learned Features

In this section, we apply the proposed pre-linearizing method described in Section 4.3 to a real-world underwater localization problem.

In the experiment described in Chapter 3, we have shown that the CNN trained features from FLS images provide a robust similarity metric among the sonar frames, which can be achieved in real-time without assumptions about the scene structures. This method can be easily adopted in a place recognition task by comparing L^2 distance between the features. However, the high-dimensional and black box nature of the feature limits the ways in which it can be used in the pose estimation problem. (PF can be a way to use it, but we will demonstrate the advantage of our proposed approach over PF in Section 4.5).

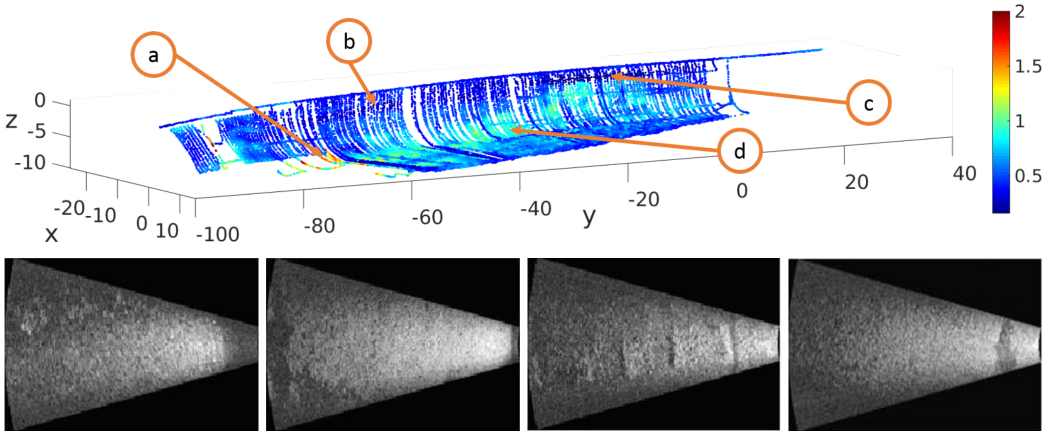
In this section, we utilize the high dimensional learned sonar feature introduced in Chapter 3 and apply our pre-linearization process to construct a real-time KF framework for FLS-based underwater localization.

The general framework we proposed in the previous section can be applied to this localization task as follows:

- $\mathbf{x}_t = (x, y, z)_t \in R^3$: The estimation state is the underwater robot position time stamped by t.
- $\mathbf{u}_t = (\Delta x, \Delta y, \Delta z)_t$: The control input is an odometry measurement obtained by other on-board sensors on an HAUV; the motion model is given by $\mathbf{x}_t = \mathbf{u}_t + \mathbf{x}_{t-1} + \mathbf{w}_t$.
- $\mathbf{z}_t \in R^{3000}$ is a high-dimensional generated by the network given raw sonar image Img_t;
- $S = (\mathbf{z}_r, \mathbf{x}_r)$: are known evaluation points of the underlying observation function $\mathbf{z}_t = \mathrm{h}(\mathbf{x}_t)$. In this case, $\mathbf{x}_r$ are positions visited in previous missions and $\mathbf{z}_r$ are corresponding features extracted from sonar frames collected at these positions.
- $\hat{\mathbf{z}}_t \in R^3$ is the new observation generated by pre-linearization.

At each correction step in the filtering system, a confidence area $B(\bar{\mu}_t)$ is given by $\bar{\mathrm{bel}}(\mathbf{x}_t)$ and a subset s_t of known points S that falls in the range of $B(\bar{\mu}_t)$ is selected. $\hat{\mathbf{z}}_t$ is given by:

$$\hat{\mathbf{z}}_t = \Sigma_j \frac{exp(-||\mathbf{z}_t - \mathbf{z}_j||_2^2)}{\Sigma_i exp\{-||\mathbf{z}_t - \mathbf{z}_i||_2^2\}} \cdot \mathbf{x}_j, \quad \{(\mathbf{z}_j, \mathbf{x}_j)\} \subset S_t$$



(a) (b) (c) (d)

Figure 4.5: Uncertainty level of $\hat{z}$. The uncertainty level of $\hat{z}$ is shown in det$(\Sigma_{\hat{z}})$ with respect to corresponding vehicle positions. The uncertainty level is strongly correlated with the vehicle position. Sonar frames at positions where uncertainty level is consistently large or small are shown.

In uncertainty estimation, we perturb $B(\bar{\mu}_t)$ by sampling $\bar{\mu}_t^j$ from $\bar{\text{bel}}(\mathbf{x}_t)$. And we perturb $\mathbf{z}_t$ by corrupting the input sonar frame with speckled noise. As shown in Fig. 4.5, the estimated uncertainty level is highly related to the saliency level of sonar frames.

4.5 Experiment on FLS Images

In this section, we evaluate the proposed modeling approach in the application of sonar-based underwater localization using real data collected in the multi-year ship hull inspection mission introduced in Section 4.4. We use the CNN model proposed and trained in Chapter 3. The feature with a dimension of 3000 trained in the 2014 dataset is used. We use this feature network to extract features and conduct experiments on the dataset from 2015. 27806 frames are served as map and a path with 11917 frames is used for a localization test, as shown in Fig. 4.6.

We implement the localization framework using the KF since the resulting observation is linear. The odometry measurements are also used as control inputs as discussed in Section 4.4.

We compare our proposed approach with two approaches that also use learned features as an observation.

1. PF: PF is a specific type of non-parameter filtering that can directly utilize any abstracted feature without explicit observation modeling.

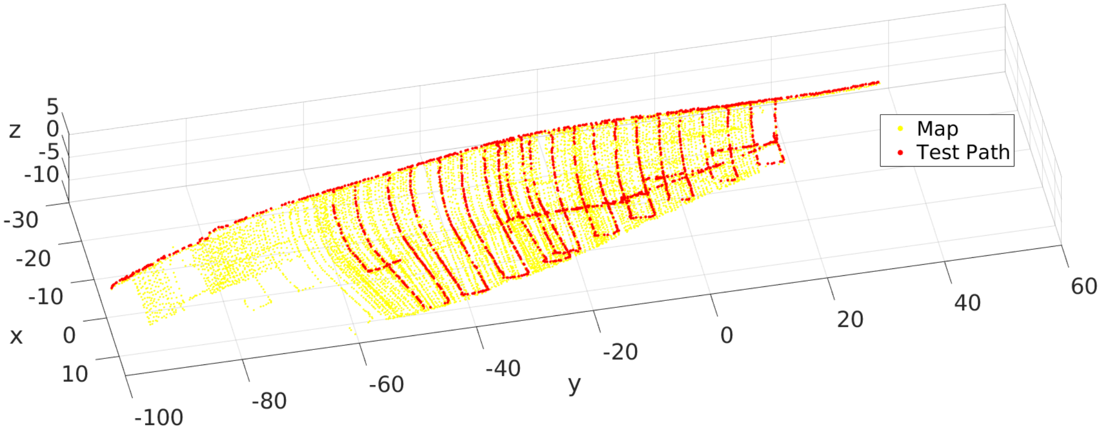


Figure 4.6: Test Path.

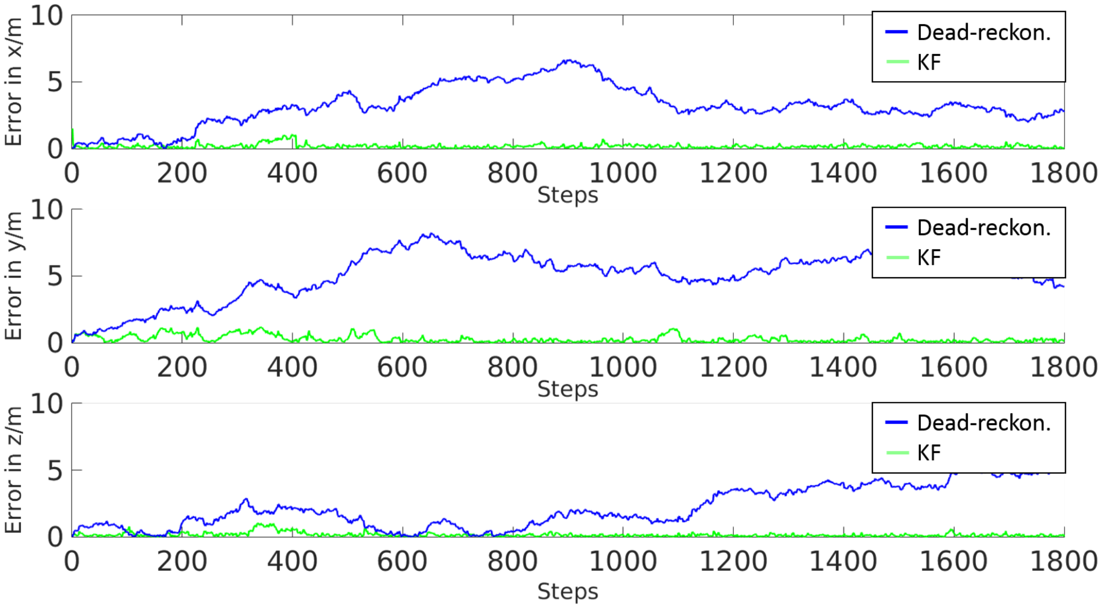


Figure 4.7: Real-time localization error using the proposed pre-linearization observation in Kalman filtering. The average processing time for each loop is $11\ \mathrm{ms}$. Blue lines denote the dead-reckoning localization error using only odometry measurements.

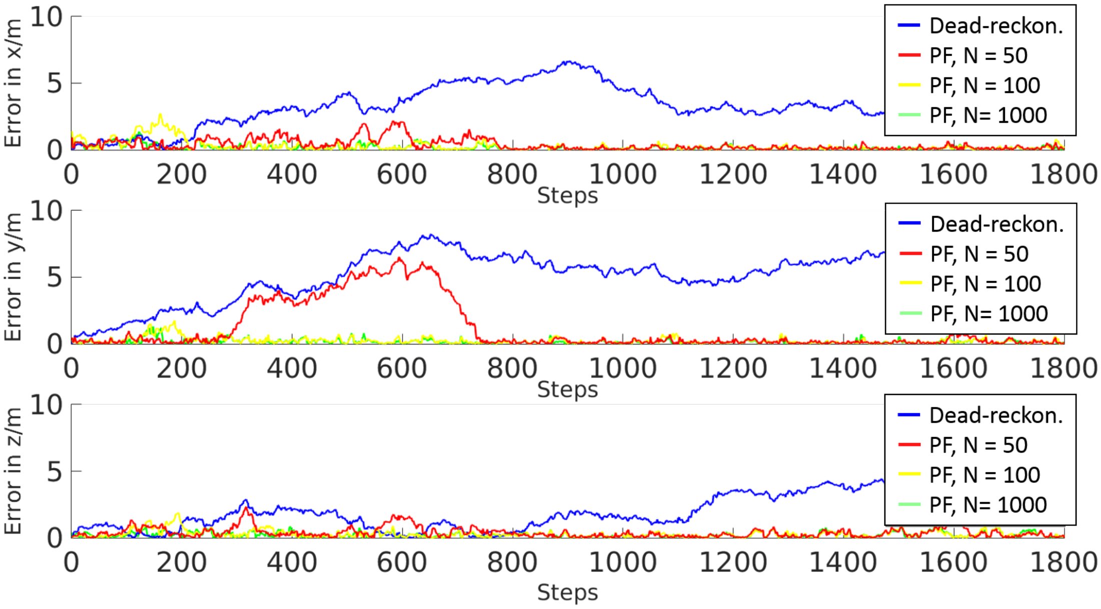


Figure 4.8: Localization using Particle filtering over original observation. N is the number of particles used in the filter. The average processing time for each loop is 109 ms when $N = 1000$ and 12 ms when $N = 50$. Blue lines denote the dead-reckoning localization error using only odometry measurements.

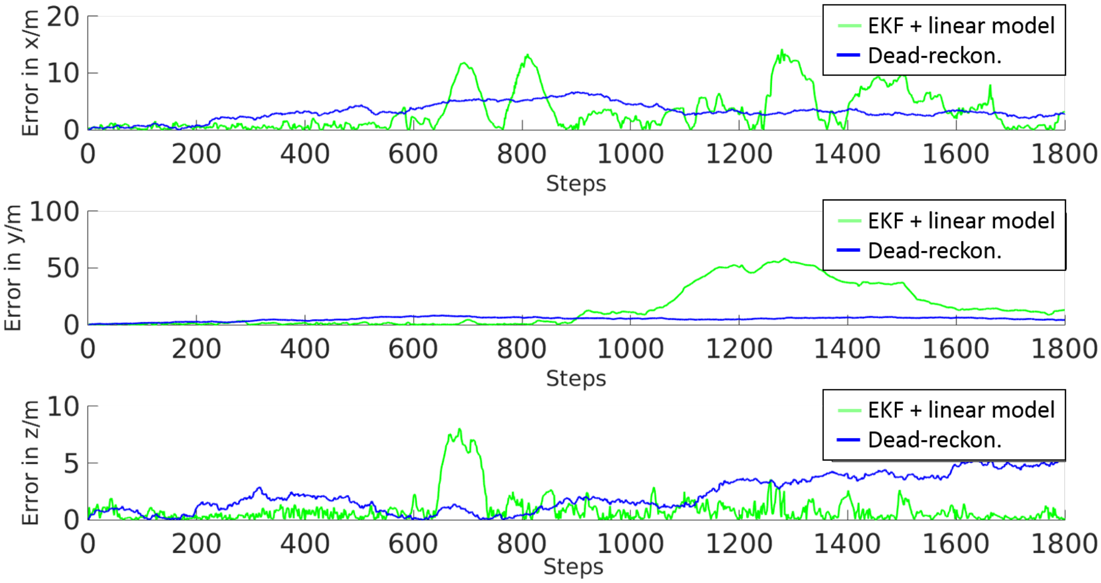


Figure 4.9: Localization using EKF with original observation model estimated from local linear regression. The average processing time for each loop is 1205 ms. Blue lines denote the dead-reckoning localization error using only odometry measurements.

2. EKF: We also compare an EKF framework with a local observation model estimated using linear regression.

For the particle filter, we evaluate the observation value using one of the nearest neighbors in the map, which is achieved using a k-d tree ($O(n \log n)$ complexity). We carry out the particle weighting in parallel using four processors to make a fair comparison to the computational efficiency.

For EKF appraoch, we model the observation function $\mathbf{z}_t$ with a linear model $\mathbf{z}_t = \mathrm{h}(\mathbf{x}_t) + n \simeq \mathrm{H} \cdot \mathbf{x}_t + \mathbf{n}$.

$$\mathrm{H} = \mathrm{argmin}\, ||\mathbf{z}_r - \mathrm{H} \cdot \mathbf{x}_r||_2,\ (\mathbf{x}_r, \mathbf{z}_r) \in S$$

The noise model is also estimated through data perturbation as before.

The localization results are given in Fig. 4.7, Fig. 4.8 and Fig. 4.9. Table. 4.1 summarizes the computational time of each method. It can be seen, as expected, that the estimated vehicle position will 'drift' due to error accumulation without the use of a corrective observation. Both the proposed framework and the particle filter are able to correct the estimated position from drifting by leveraging the sonar imagery. Using our proposed pre-linearization approach, the KF is able to achieve a promising performance in localization accuracy with very low processing time. In the particle filter, when the number of particles is relatively large and able to capture the diversity of the system, the accuracy is comparable to our proposed method. However, in this case, the PF becomes too slow to enable real-time performance in a system with multiple processes. If we drop the number of particles to increase the efficiency to be comparable with our proposed method, the accuracy will drop to unacceptable levels. For the PF method, nearest neighbor searching needs to be done as many times as the number of particles for data association. That means although the particle filter is able to make use of an abstracted feature in a simple fashion, it is too time consuming for real-time localization for high-dimensional features.

It can also be seen that the estimated linear model performs poorly. Two possible reasons might account for this performance. The underlying observation model is too non-linear so the resulting linear estimation is poor. It is also possible that the number of points in the map are not large enough to support the estimation of such a high dimensional function. Given that we can estimate the function properly, the processing time indicates that the direct usage of high dimensional features using the KF family of filters is unachievable without the proposed approach.

The most time consuming part in the localization procedure in our proposed method is nearest neighbor searching in the map. Despite the use of an efficient K-d tree implementa-

Table 4.1: Computational Time

	Prop.	PF-1000	PF-50	EKF
Time per step [s]	0.011	0.109	0.012	>1

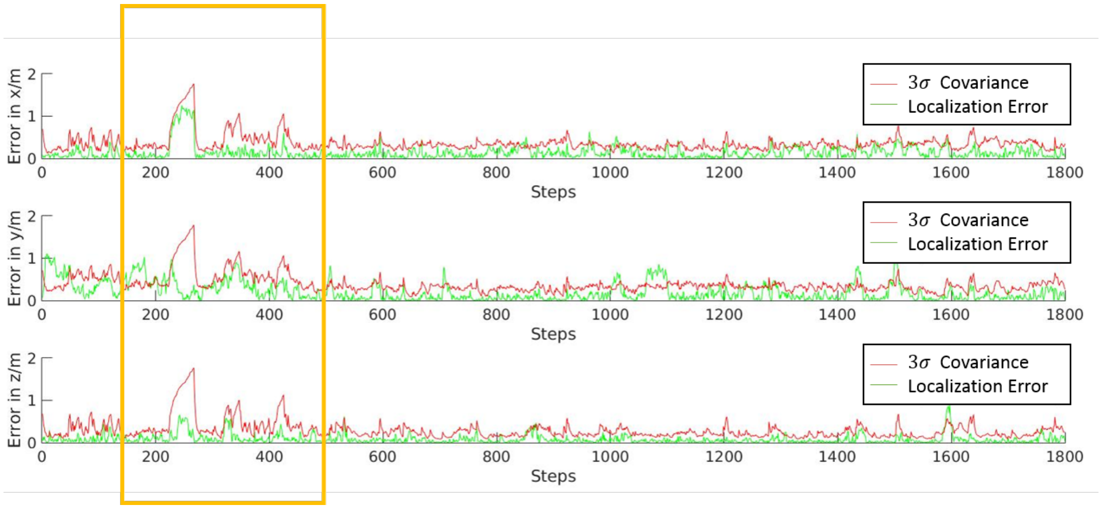


Figure 4.10: Kalman filtering localization with proposed noise estimation.

tion, this is still a somewhat computationally expensive step. The complexity is $O(n \log n)$ with respect to the map for each query. For EKF or other filters that explicitly use the observation model, the inquiry needs to be done once for the calculation of c_t.

The experiment results demonstrate the effectiveness of the feature uncertainty estimation as the localization accuracy in these probabilistic filtering frameworks require reasonable uncertainty models to function properly. However, to better explicitly evaluate the uncertainty measurement approach we proposed, we compared the actual estimation error with believed uncertainty from the filtering system in Fig. 4.10. It can be seen that the uncertainty estimation of the filtering system, which is estimated based on both odometry uncertainty and our generated model uncertainty, is able to bound the actual estimation error in most the cases.

As we discuss in Section 4.3.3, the proposed uncertainty measurement approach encodes the uncertainty specific to the environment. To demonstrate the advantages of this property, we compared the proposed uncertainty measurement approach with a fixed uncertainty model estimated from the mean value across the dataset. The uncertainty measurement results of the vehicle are given in Fig. 4.11. In this comparison, we can tell that the proposed approach is able to provide a better estimation of the uncertainty than even a reasonable fixed model, as expected.

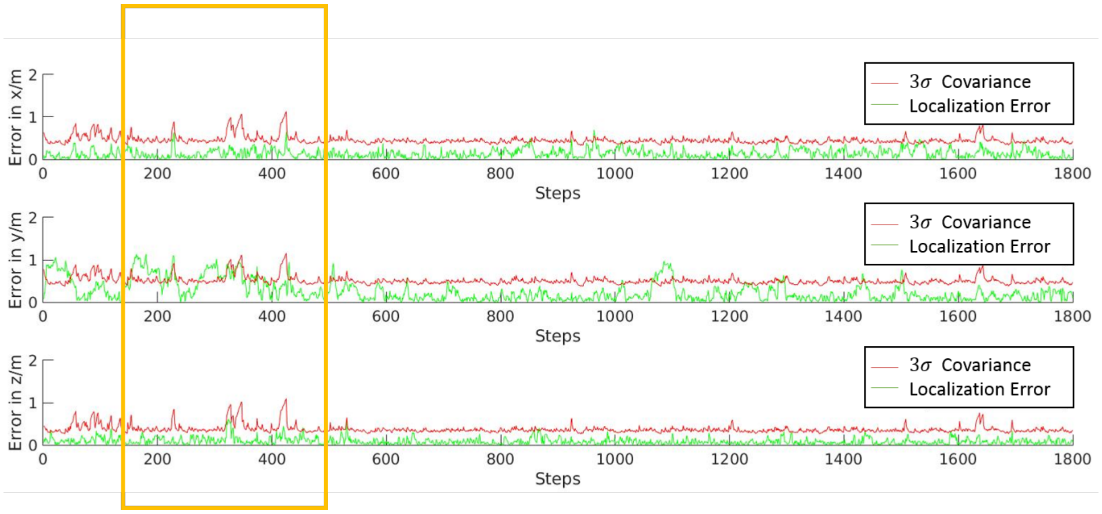


Figure 4.11: Kalman filtering localization with constant noise model.

4.6 Experiments on Other Modalities

To illustrate that the proposed method is able to be generalized to other applications, this section provides experimental validation of a scenario of real-time localization using high dimensional CNN features extracted from camera images. We generated the testing dataset using realistic synthetic camera images from the ICL-NUIM [38] tool, where arbitrary views of a 3D scene model can be rendered with artificial sensor noise. The living room scene in ICL-NUIM, as depicted in Fig. 4.12, is used in the experiment.

In this experiment, we evaluate our approach in the application of camera position estimation. We assume the camera is moving freely on the XY plane. A 2D path of the camera moving in the plane following a spline curve is randomly generated and serves as the ground truth, as depicted in Fig. 4.13. We estimate the camera position x_t within an EKF, using noisy odometry measurement u_t and camera observation z_t, which is pre-processed by the proposed local linearization approach. For raw image features z_t, we use the **fc8** layer of AlexNet [5] with a pre-trained model provided by Caffe [51], which extracts a 1000 dimension feature vector from an input rgb image.

The localization result compared to dead reckoning is given in Fig. 4.13 and Fig. 4.14. It can be seen that our proposed approach is able to efficiently use the high-dimensional and highly abstracted feature from a camera image and contribute to the localization system. As the drifting error accumulates, the camera-based EKF is able to keep a bounded error in localization.

Figure 4.12: Example view of Living Room in ICL-NUIM [38]

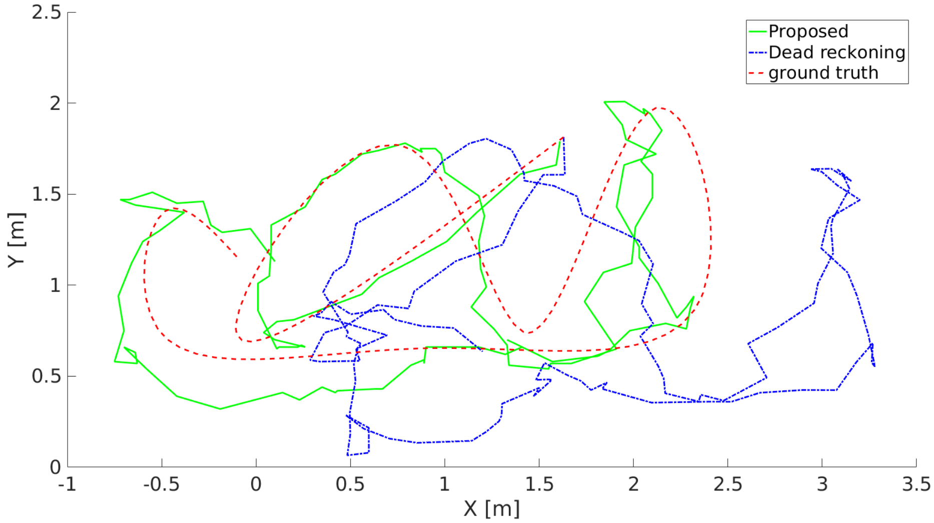


Figure 4.13: Localization result on XY plane.

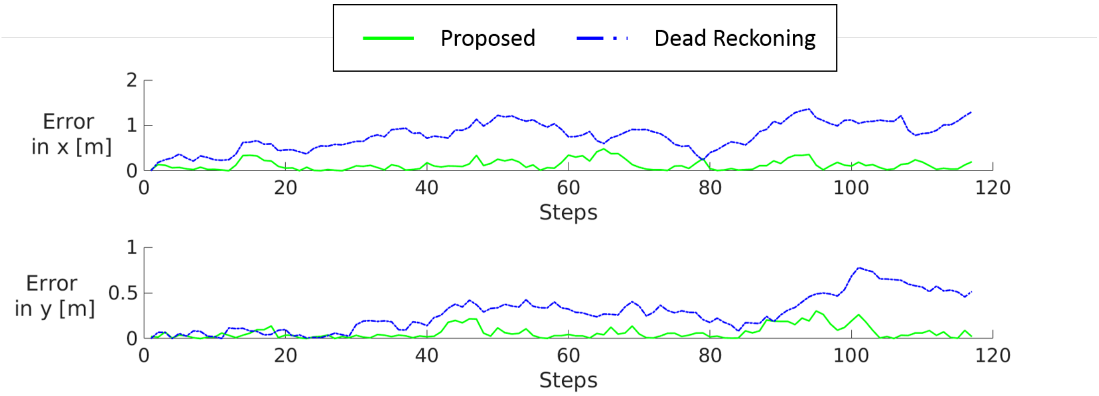


Figure 4.14: Localization Error. We evaluate localization error for both of the axes. The shared trend is that our proposed algorithm is able to directly make use of the high-dimensional and highly abstract feature from an input image to provide localization information within the EKF framework.

4.7 Conclusion and Future Work

In this chapter, we propose an observation pre-linearization approach that enables real-time performance for recursive filtering with high-dimensional features. A practical and efficient uncertainty measurement of the resulting new observation model is provided based on data perturbation. We evaluate the proposed pre-linearization procedure by applying it to the application of imaging sonar-based underwater robot localization, which results in the first real-time solution using this modality. Experimental results using real world data show that the proposed method is able to achieve real-time performance in a KF framework. The accuracy is comparable to sampling-based methods while the computation complexity is much more favorable for real-time applications. In the future we intend to generalize the method to online SLAM approaches where localization and map collection are performed simultaneously.

CHAPTER 5

Forward-looking Sonar Pose-graph SLAM

5.1 Introduction

In the previous chapter, we explore a general pipeline for underwater localization using an image-level descriptive feature of Forward-looking Sonar (FLS). The proposed approach enables an autonomous underwater vehicle (AUV) to perform real-time localization in the environment with a map built by previous observations. In this chapter, we explore the utilization of FLS in a navigation system within an unknown environment. We propose methods to extract geometry information from FLS that improves vehicle pose estimation within a simultaneous localization and mapping (SLAM) framework.

As we discussed in Section 1.3, FLS has become an important alternative that offers greater range and insensitivity to water visibility when compared to optical sensing. Nevertheless, the utilization of this modality suffers from unique challenges including low image texture and high Signal-Noise-Ratio (SNR). Due to these problems, the majority of FLS applications have been using it for environment reconstruction given known vehicle poses. Less work has been done to improve the vehicle pose estimation through FLS images.

In this chapter, we propose a novel pose-graph SLAM algorithm leveraging FLS as the sole perceptual sensor to provide ego-motion drift correction. In this work, we address many of the practical problems associated with leveraging signals from FLS, including feature sparsity, low reliability in data association, and geometry estimation. The contributions of this chapter are represented in our publication [68], which include:

(i) Creation of a system to identify and select the most informative sonar frames for use in improving system efficiency and reliability, while avoiding the singularities which pervade the two-view geometry estimation problem when using FLS;

(ii) Development of a robust sonar feature matching strategy using pose prior within a SLAM framework; instead of global searching within the whole image, believed-region searching is conducted to encode constraints from vehicle pose prior and sonar device field-of-view (FOV); and

(iii) Evaluation of the efficacy of our algorithm for the application scenario of periodic underwater structure inspection and assessment.

Experimental results show that the proposed algorithm is able to provide robust loop closure detection and relative pose constraint estimation from FLS which ultimately corrects the localization of the platform and minimizes drift.

5.1.1 Outline

The rest of the chapter is arranged as follows: (i) In Section 5.2 we give a brief introduction on related work using FLS to help AUV navigation systems within an unknown environment; (ii) in Section 5.3, we provide a thorough introduction to the proposed algorithm; (iii) in Section 5.4, we evaluate the approach through the use of ship hull inspection data; and (iv) finally, in Section 5.5, we summarize the chapter.

5.2 Background

In recent history, pioneering research has been conducted to enable the use of FLS for AUV navigation. Walter, Hover, and Leonard [96] proposed the first SLAM implementation using Exactly Sparse Extended Information Filter (ESEIF) with manually selected features from sonar imagery. While building the foundation of FLS for SLAM, this work focused more on efficiency in building a feature map to emphasize real-time performance. The work avoided directly addressing the geometric ambiguities in elevation embedded in such measurements. Johannsson et al. [53] proposed detecting and matching FLS image features using Normal Distribution Transformation (NDT). This work makes a planar assumption where all feature points are assumed to lie on a plane parallel to the vehicle. This assumption helps to add extra constraints to address the ambiguity in elevation measurements in the SLAM observations. However, it also limits the application scenarios in which such an algorithm can be used.

A similar assumption was made by Shin et al. in their recent work on FLS-based bundle adjustment for seabed mapping. This work also introduces the Accelerated-KAZE (A-KAZE) [4] feature for acoustic sonar images and displays impressive performance on the challenging data association problem.

Aykin and Negahdaripour relax the planar assumption of prior work for the two-view scan matching case [9]. However, they still employ a local planar surface assumption per image patch allowing the use of acoustic shadowing for feature detection and registration. The use of shadow information presents more stable performance for the image registration problem at the cost of longer computation time and greater structural requirements to excite shadowing effects.

In more recent literature, Huang and Kaess [46] introduced Acoustic Structure From Motion (ASFM), which used multiple sonar viewpoints to jointly reconstruct 3D structures as well as sensor motion, with no specific assumption of the scene. In their work, a statistic analysis was conducted to identify and discuss the degeneration cases for which ASFM would fail. Particularly, they analyze which motion patterns make for a poorly constrained optimization problem. In their recent work on acoustic-inertial odometry estimation Yang and Huang [102] extend the discussion of this problem to more abstract theoretical cases. Although ASFM shows promising results for providing sonar-based geometry estimation without underlying scene assumptions, that work was primarily evaluated on simulated data. However, despite ongoing work there are still open issues, particularly when it comes to the practical deployment of an acoustic SLAM system using FLS. These problems include, but are not limited to, robust data association, efficient key frame selection for real-time performance, and identification and removal of degeneration cases in the optimization. In this chapter, we present novel algorithms that make use of ASFM as front-end to develop the missing pieces for a complete end-to-end real-time acoustic SLAM system. The hurdles to creating a deployable framework that can run on an AUV are addressed, providing solutions to the practical problems discussed above.

5.3 Methodology

In this section, we describe the proposed acoustic SLAM system in detail. An overview of the approach is given in Fig. 5.1. A pose-graph SLAM back-end is used to estimate the vehicle's full 6 degree-of-freedom (DOF) pose, $x_i = [x, y, z, \phi, \theta, \psi]$ given all the sensor measurements as constraints between poses. Fig. 5.2 depicts the general graphical model of our pose-graph representation. Each node in the graph, x_i, corresponds to a vehicle pose with a sensor measurement associated. Two types of constraints from sensor measurements are considered in this context: odometry measurements from the Doppler velocity log (DVL) and inertial measurement unit (IMU), and sonar constraints from the FLS local structure from motion (described later in detail in Section 5.3.2). We assume independent Gaussian noise for both odometry and sonar measurements. The measurement model for odometry is

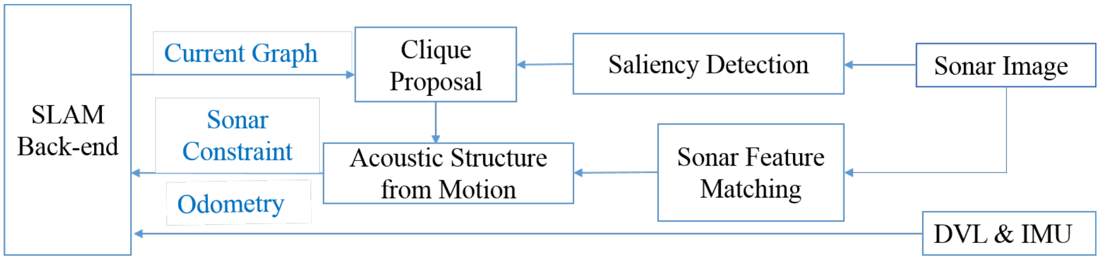


Figure 5.1: System flowchart: The proposed algorithm is supported by a pose-graph SLAM back-end. It optimizes the vehicle poses given all the measurements. Raw sonar frames are passed through a saliency test where salient frames are considered for loop closure proposal based on potential information gain. Marginalized pose constraints, after local structure from motion, will be fed to the SLAM back-end providing drift correction from sonar measurements.

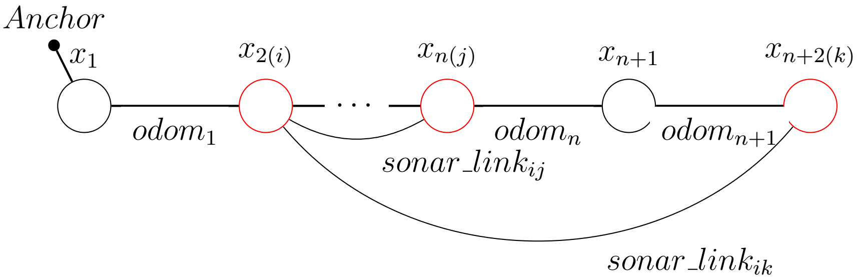


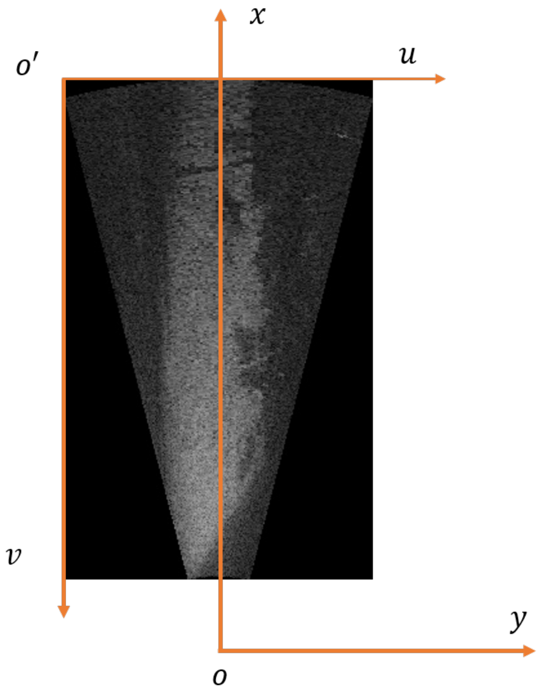
Figure 5.2: Depiction of pose-graph SLAM using constraints from local FLS bundle adjustment. An example of loop closure clique is marked in red (i, j, k).

given by:

$$z_{ij}^k = f(x_i, x_j) + w_k, \tag{5.1}$$

where $f()$ denotes the generative model of odometry between two poses and w_k is Gaussian noise with its covariance scaled proportionally to the elapsed time separating the poses.

To model the constraints provided by FLS images, we propose to use marginalized pose constraints from a local structure from a motion problem within a clique of loop closure sonar images. In this section we start with a brief introduction of sonar geometry and sonar-based structure from motion in Section 5.3.1 and Section 5.3.2. Then we give more details about how we provide an advanced loop closure hypothesis proposal and feature matching that improves the robustness of the sonar-based structure from motion and how we address the challenges in feature sparsity in Section 5.3.3 and Section 5.3.4 respectively.



5.3.1 Sonar Frame Geometry

As introduced in Section 1.3.1, an FLS observes the scene and returns the measurement in the form of range and bearing (r, θ). The measurement for each data point (r, θ) can originate anywhere on the arc defined by (r, θ) in the Spherical coordinate system centered at the projection center of the sonar head, as depicted in Fig. 1.13.

In the local Cartesian coordinates of the sonar frames, a 3D scene point P_s can be obtained from the Spherical coordinates as:

$$P_s = \begin{bmatrix} X_s \\ Y_s \\ Z_s \end{bmatrix} = \begin{bmatrix} r\cos\theta\cos\phi \\ r\sin\theta\cos\phi \\ r\sin\phi \end{bmatrix} \tag{5.2}$$

The raw measurement from the transducer, a 2D array of (r, θ), is then converted to a 2D image for easier interpretation and post processing. The specific manner in which image conversion occurs can vary depending on the application scenario. Here, we only discuss the most common: the Cartesian coordinate image conversion. Under this conversion, a 3D point P_s is projected on the image plane $(\phi = 0)$:

$$\boldsymbol{p_s} = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \gamma \cdot Y_s/cos\phi + \frac{w}{2} \\ \gamma \cdot (r_{max} - X_s/cos\phi) \end{bmatrix} \tag{5.3}$$

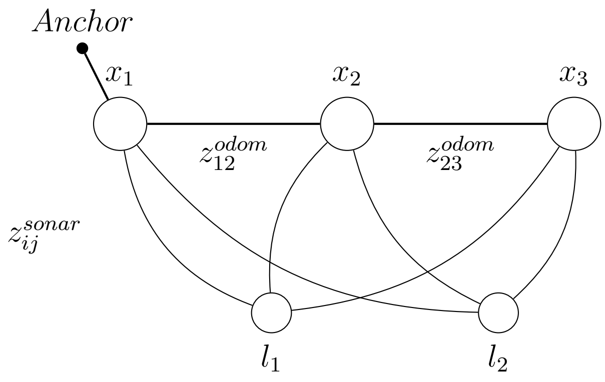


Figure 5.3: Graph Model for Local Acoustic Bundle Adjustment

where (u, v) is the pixel location centered at the upper left corner of the sonar image and $[w, h]$ are the width and height of the image, as depicted in Fig. 5.3.1. γ is a scaling factor between the pixel plane and the physical zero elevation plane, which is given by:

$$\gamma = \frac{w}{2r_{max}\sin(\theta_{max}/2)}, \tag{5.4}$$

where θ_{max} is the sonar FOV of bearing.

5.3.2 FLS Structure from Motion

Given the geometry of imaging sonars described in Section 5.3.1, we can build a local structure from motion problem within a small set of sonar frames by modeling both the odometry measurements and sonar feature measurements. We consider a three-frame structure from motion in the implementation. It is straightforward to generalize the work over any number of frames.

The structure from motion solves for the maximum *a posteriori* (MAP) vehicle poses and landmark positions $\Theta = \{x_i, P_k\}$ given all the measurements $Z = \{z_{ij}^{odom}, z_{im}^{sonar}\}$ in a given clique:

$$\Theta^* = \text{argmax}_\theta\, p(\Theta|Z) = \text{argmax}_\theta\, p(\Theta)p(Z|\Theta). \tag{5.5}$$

We model the odometry measurements again as described in Eq. 5.1. In the structure from motion setting, instead of direct odometry measurements from the sensors, we use relative odometry from the current SLAM estimate, given by:

$$z_{ij}^{odom} = f(\bar{x}_i, \bar{x}_j) + w_{ij}^{odom}, \tag{5.6}$$

where $\bar{x}_i$ and $\bar{x}_j$ are pose estimates from the SLAM back-end, and $\Sigma_{w_{ij}^{ba}}$ is given by the propagated covariance of $f(\bar{x}_i, \bar{x}_j)$.
We define sonar measurement as:

$$z_{im}^{sonar} = h(P_m, x_i) + v_{im}, \tag{5.7}$$

where $h()$ denotes the generative model that converts a landmark position P_m to the local frame of vehicle pose x_i, then projects the landmark on the image plane following Eq. 5.3. We model the measurement uncertainty proportional to the radius of the detected key point. The whole structure from motion is then a solution to the least square problem of:

$$\Theta = \operatorname{argmin}_x [\Sigma_{k=1}^{m} ||h(P_m^g, x_{i_k}) - z_{i_k}^{m}||_{\Sigma}^{2} + \Sigma ||f(x_i, x_j) - z_{ij}^{slam}||_{\Sigma}^{2}]. \tag{5.8}$$

Mahalanobis distance ($||x||_{\Sigma}^{2} = x^T \Sigma^{-1} x$) is used for all the residuals. Note one implementation detail is that we inflate the uncertainty of the relative odometry constraint in the z-axis to compensate for the spatial ambiguities in the biased motion pattern particular to the ship hull dataset. After the optimization, we take the marginalized odometry constraints as new pose constraints for the SLAM back-end, which is given by:

$$z_{ij}^{ba} = f({x^*}_i, {x^*}_j) + w_{ij}^{ba}. \tag{5.9}$$

5.3.3 Saliency Aware Loop-closure proposal

To ensure real-time performance of the system, an efficient hypothesis proposal system is essential. This system will propose loop closures based upon sonar frame cliques in a local acoustic structure from motion. The selection of these loop-closure cliques is based upon the evaluation of both potential information gain given successful registration and the estimated saliency of the underlying sonar frames (a proxy for potential matching quality).

The use of information gain for link proposals was introduced by Ila, Porta, and Andrade-Cetto [50] in which one only adds informative links to keep the SLAM graph compact. Kim and Eustice [59] also extend this approach for loop-closure hypothesis proposals between camera images. The information gain of a new measurement between two nodes under the assumption of a jointly Gaussian distribution is given by:

$$L = H(X) - H(X|z_{ij}) = \frac{1}{2} \ln \frac{|S|}{|R|}, \tag{5.10}$$

where $H(X)$ and $H(X|z_{ij})$ is the prior and posterior entropy of the graph. R is the measurement covariance and S is the innovation covariance:

$$S = R + [H_i, H_j] \begin{bmatrix} \Sigma_{ii} & \Sigma_{ij} \\ \Sigma_{ji} & \Sigma_{jj} \end{bmatrix} [H_i, H_j]^T \tag{5.11}$$

Note that the above equation can be easily extended to the case of several measurements:

$$S_{i,j,k}^{12\times 12} = R_{i,j,k}^{12\times 12} + H \begin{bmatrix} \Sigma_{ii} & \Sigma_{ij} & \Sigma_{ik} \\ \Sigma_{ji} & \Sigma_{jj} & \Sigma_{jk} \\ \Sigma_{ki} & \Sigma_{kj} & \Sigma_{kk} \end{bmatrix}^{18\times 18} H^T, \tag{5.12}$$

where R here becomes the covariance of the new measurement $z_{new} = (z_{ij}, z_{ik})$ and H is the corresponding Jacobian matrix of z_{new}. They are given by:

$$R_{i,j,k} = \begin{bmatrix} \Sigma_{z_{ij}z_{ij}} & \Sigma_{z_{ij}z_{ik}} \\ \Sigma_{z_{ik}z_{ij}} & \Sigma_{z_{ik}z_{ik}} \end{bmatrix} \tag{5.13}$$

and

$$H = \begin{bmatrix} H_i^{z_{ij}} & H_j^{z_{ij}} & \mathbf{0} \\ H_i^{z_{ik}} & \mathbf{0} & H_k^{z_{ik}} \end{bmatrix} \tag{5.14}$$

Eq. 5.10 provides an evaluation of potential information gain that will contribute to the SLAM optimization problem. Another advantage of the information gain metric is that it encourages the cliques with higher complexity of motion, which decreases the chance of a singularity in the local structure from motion, as discussed in [46, 102].

However, as we discussed throughout this thesis, the sparse distribution of strong sonar features is a fundamental challenge in making a successful structure from motion constraint. Proposals based *solely* on information gain can lead to a lot of failed attempts due to lack of features, decreasing the efficiency of the entire system. To address this problem, we combine both information gain and the sonar image saliency to determine which images to use for proposed loop closures.

In Chapter 4, we proposed a global image descriptor feature using a convolutional neural network (CNN) that provides informative features for image matching. Fig. 5.4 depicts the data flow for training and run-time use of the feature model. A saliency score evaluating the uniqueness of the features in a local neighborhood trained on a given *map* is also introduced. However, the map dependent score is not practical in the navigation systems in unknown environments. In this work, we propose an online saliency evaluation based on

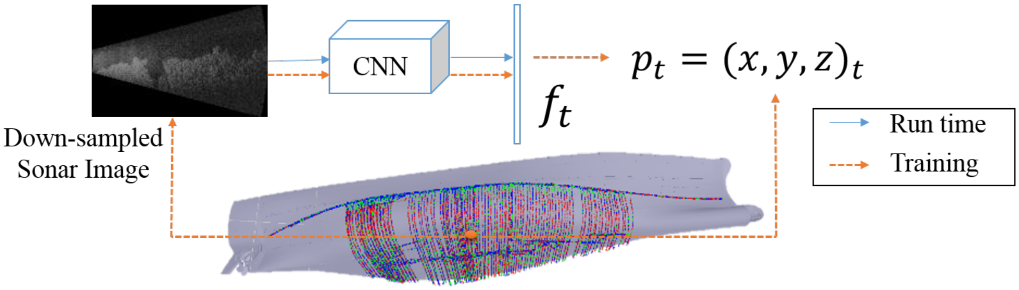


Figure 5.4: Image-level descriptive sonar feature training procedure proposed in Chapter 4.

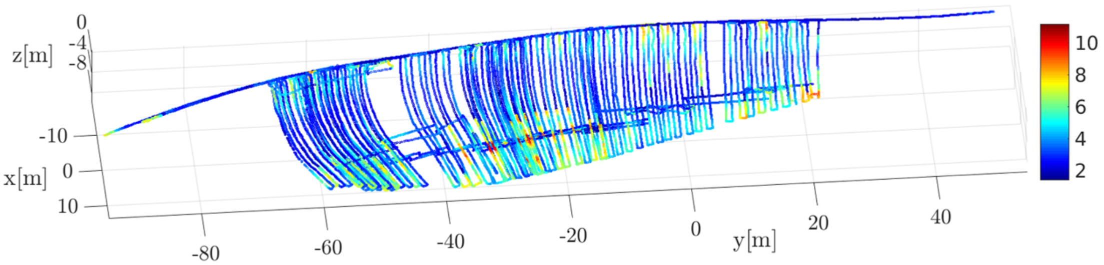


Figure 5.5: Saliency score estimated from image-level sonar feature with respect to the ship hull.

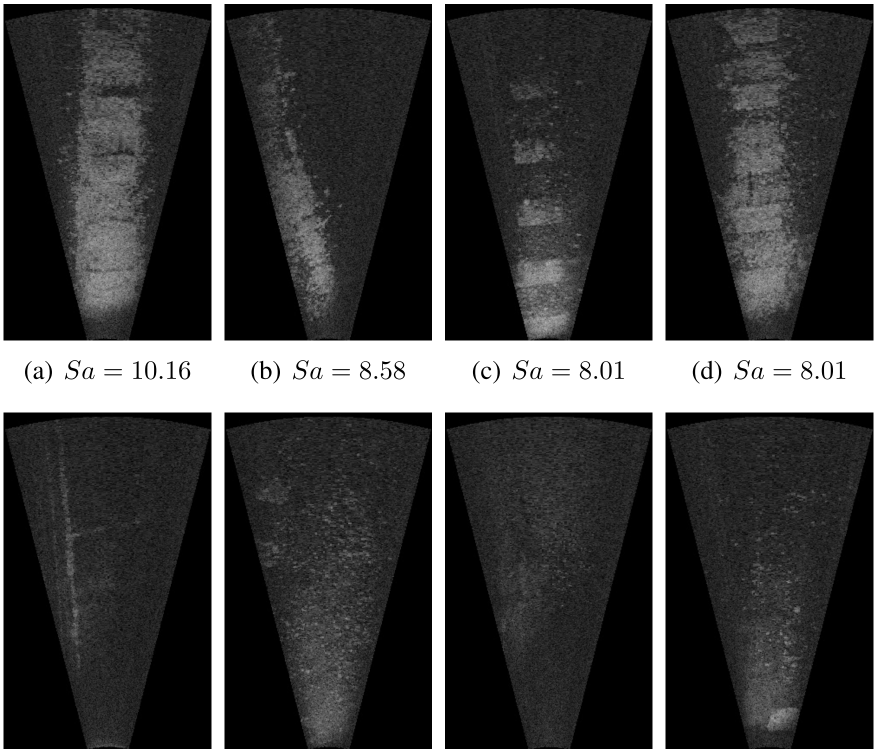
local sensitivity of the CNN-extracted features given by:

$$Sa_j = \frac{\sqrt{\Sigma_{i=1}^{N_j} ||f_i - \bar{f}||_2^2}}{\sqrt{\Sigma_{i=1}^{N_j} ||p_i - \bar{p}||_2^2}}, \ p_i \in B_j, \tag{5.15}$$

where Sa_j is the saliency score of frame j, B_j is the support area of frame j and $p_i = (x, y, z)$ are vehicle locations within the support area B_j. The saliency score captures the variability of sonar features f_i within a fixed physical support area around frame j. Fig. 5.5 depicts the distribution of the scores evaluated online as the vehicle moves with respect to the environment (in this case the ship hull). Examples of the sonar frames with associated saliency scores are given in Fig. 5.6. As can be seen, the saliency score is well correlated with sonar images where strong texture exists.

We use the proposed saliency score to prune the loop closure proposals by evaluating information gain within only the salient frames:

$$L^s(i, j, k) = \begin{cases} L(i, j, k) = \frac{1}{2} \ln \frac{|S_{i,j,k}|}{|R_{i,j,k}|} & \overline{Sa} > \lambda_s, \\ 0 & otherwise. \end{cases} \tag{5.16}$$



(a) $Sa = 10.16$ (b) $Sa = 8.58$ (c) $Sa = 8.01$ (d) $Sa = 8.01$

(e) $Sa = 1.64$ (f) $Sa = 1.99$ (g) $Sa = 1.60$ (h) $Sa = 1.60$

Figure 5.6: Example sonar frames with saliency scores. (a) (b) and (c) give examples of high saliency sonar frames, where we can see strong texture and features. (d) (e) and (f) give examples of low saliency frames, where little texture and few features can be recognized.

$\overline{Sa}$ denotes the mean saliency score in the clique (i, j, k) and λ_s is a cut-off threshold for the saliency score. $\lambda_s = 5$ was experimentally found to be optimal for the data gathered in the ship hull dataset. The saliency-aware information gain $L^s(i, j, k)$ is used to propose and prioritize the cliques considered in the local structure from motion front-end.

Since the proposal of a clique of size m is an $O(n^m)$ problem, we use a sampling-based method for the initial proposal.

In Section 5.4, we show that the saliency-aware information gain helps to increase the chance of successful local structure from motion of sonar frames with sufficient valid matches.

5.3.4 Sonar Feature Matching Using Pose Constraint

The problem of constructing feature point associations between sonar frames is one of the most challenging tasks in using an imaging sonar in a pose estimation and reconstruction problem. In this work, we propose to use pose priors that are available from the SLAM system to constrain the search area for feature matching within sonar frames. For a feature point p_m^i detected on frame i, we expect the corresponding feature point location in frame j to be defined as:

$$p_j^m = \mathbf{h}(P_m(p_i^m, \phi_m), x_i, x_j), \tag{5.17}$$

where ψ_m is the elevation of underlying 3D scene point P_m and (x_i, x_j) are the corresponding vehicle poses. Given the elevation ϕ_m and feature point ψ_m, P_m can be uniquely reconstructed following Eq. 5.2 and Eq. 5.3.

From Eq. 5.17, we can approximate the distribution probability of p_j^m using linear covariance propagation:

$$\mu_j^m = \mathbf{h}(P_m(p_j^m, \phi_m), x_i, x_j)|_{\phi_m=0} \tag{5.18}$$

$$\Sigma_j^m = J\Sigma_{p_i^m,\phi_m,x_i,x_j} J^T, \tag{5.19}$$

Where J is the Jacobian matrix of function h. $\Sigma_{p_i^m,\phi_m,x_{ij}}$ denotes the joint covariance matrix of p_i^m, ϕ_m, x_{ij}. Note that p_i^m, ϕ_m, and (x_i, x_j) are independent variables. We can simplify Eq. 5.19 by:

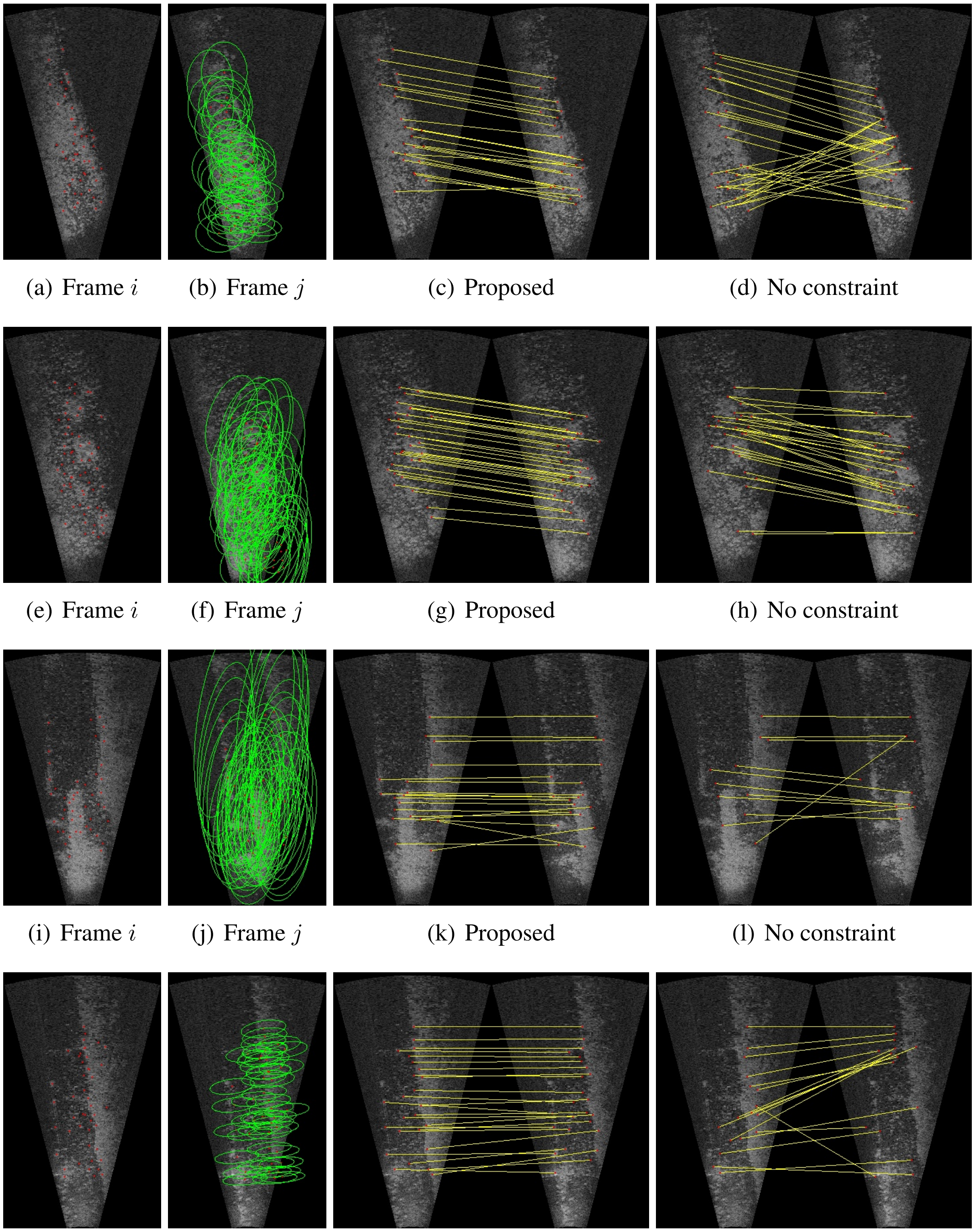
$$\Sigma_j^m = J_{p_i^m}\Sigma_{p_i^m} J_{p_i^m}^T + \sigma_{\phi_m}^2 J_{\phi_m} J_{\phi_m}^T + J_{x_ix_j}\Sigma_{x_ix_j} J_{x_ix_j}^T. \tag{5.20}$$

$\sigma_{\phi_m}^2 = \phi_{max}$ is the covariance of ϕ_m since we have cue of ϕ_m due to the ambiguity. $\Sigma_{x_ix_j}$ is the joint covariance of pose frame x_i and frame x_j available from SLAM. The radius of the meaningful keypoint neighborhood is used to give the covariance of the detected feature point $\Sigma_{p_m^i}$.

$$(p_m^j - \mu_j)^T(\Sigma_m^j)^{-1}(p_m^j - \mu_j) = \kappa^2, \tag{5.21}$$

where κ^2 follows a χ_2^2 distribution.

A set of example matching results is given in Fig. 5.7 and it is also compared to nearest neighborhood matching. A-KAZE detection and speeded up robust features (SURF) feature descriptors are used in the implementation. It can be seen that our matching approach results in many more correct associations. While some outliers exist, even in the improved matching approach, the ratio of inliers is high enough to be effective in a structure from motion framework. In the implementation of the system an initial structure from motion is run to enable the identification of gross outliers that can be excluded from a second structure from motion pass. Features are only included in the structure from motion problem when they are found in all frames in the clique.



(a) Frame i (b) Frame j (c) Proposed (d) No constraint

(e) Frame i (f) Frame j (g) Proposed (h) No constraint

(i) Frame i (j) Frame j (k) Proposed (l) No constraint

(m) Frame i (n) Frame j (o) Proposed (p) No constraint

Figure 5.7: Sonar feature matching using pose constraints. The first column gives sonar image i with detected feature points marked in red. The second column gives sonar frame j. The expected correspondent location of the feature points in frame j is marked in red on frame j. The searching area in frame j is marked in green. The third column gives feature matching results using a pose constraint. The fourth column gives feature matching results using the nearest neighboring matching.

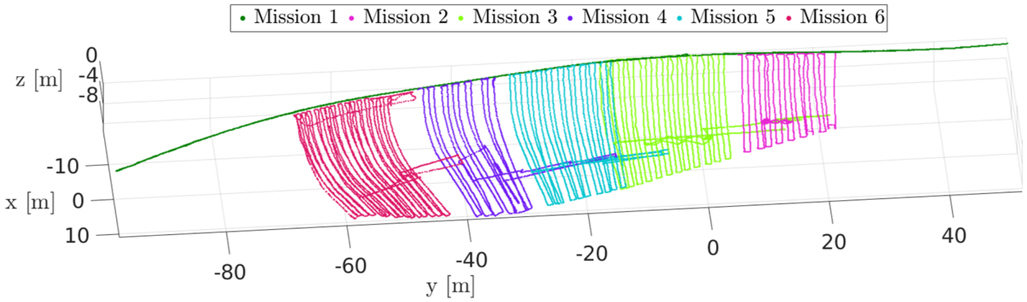


Figure 5.8: SLAM missions collected in the inspection of the SS-Curtiss in 2014. This figure depicts six missions in a common reference frame.

Table 5.1: Trajectory Statistics.

Mission	1	2	3	4	5	6
length in X axis [m]	25.17	15.9	27.40	26.54	26.62	23.54
length in Y axis [m]	149.79	16.46	25.47	32.07	12.79	27.68
length in Z axis [m]	0.1173	8.767	9.51	9.66	9.53	9.21
Num. of nodes	2240	2210	4389	2987	2597	5440

5.4 Experiment

In this section, we evaluate our proposed algorithm on real-world data collected in a ship hull inspection application. The dataset used in our experiment comes from the inspection of the US-Curtiss (Fig. 1.7) in 2014. As depicted in Fig. 5.8, the data consists of a set of trajectories around different portions of the ship hull (focused on areas that exclude the rudder and screws). Information on the statistics of each trajectory is given in Table 5.1.

We evaluate the loop closure clique proposal approach by analyzing the success rate of all proposals. If within a hypothesis proposal clique, sufficient feature chains are built and the structure from motion optimization converges, we mark the proposal as 1 (true positive). Otherwise, if insufficient feature chains are detected or the structure from motion is under constrained due to a singular motion pattern, we mark it as 0 (false positive). The saliency-aware information gain L^s is used to rank the proposals. A Precision-Recall (PR) curve over all the hypothesis proposals in **Mission 2** is given in Fig. 5.9. The results show that the saliency aware hypothesis proposal method is able to effectively suppress proposals in the low saliency area, greatly increasing the number of true positives.

We provide quantitative and qualitative results evaluating the localization accuracy of the proposed SLAM algorithm as compared to dead reckoning localization results. Since the hovering autonomous underwater vehicle (HAUV) is operating in a GPS-denied environment

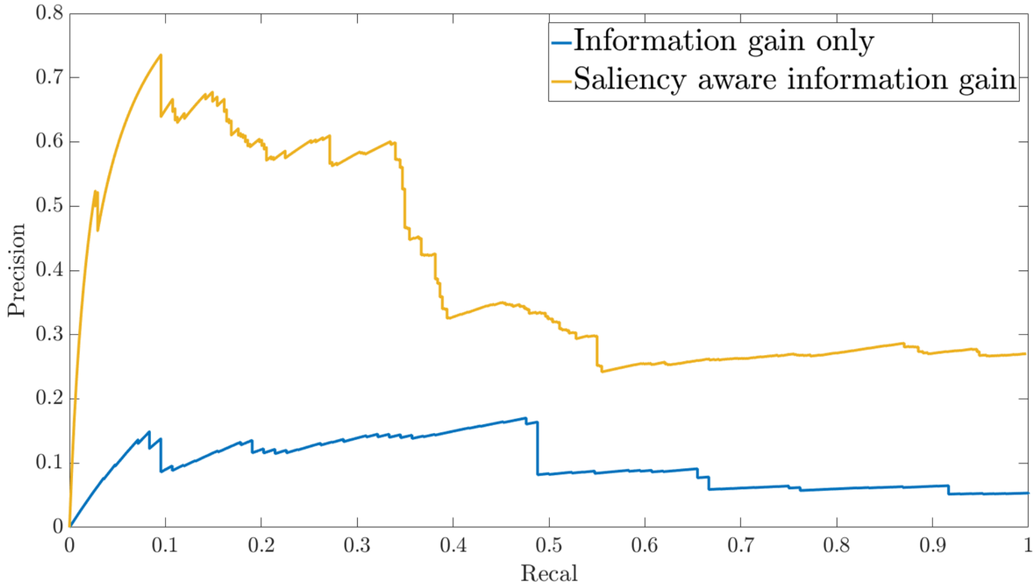


Figure 5.9: Precision-Recall (PR) curve of hypothesis proposal using saliency aware information gain. The hypothesized proposal clique is ranked by saliency and information gain and the PR curve is calculated based on success rate. We compare the proposed method with a more traditional information-gain only score.

with no external reference set up, there is no ground truth available for the localization accuracy. To evaluate the performance, we use a relative ground truth generated from an off-line visual-based bundle adjustment framework using camera images, which is reported by Ozog et al. in [83]. Table 5.2 depicts SLAM localization accuracy with respect to ground truth positions compared with dead-reckoning trajectories. To account for the variability introduced by the randomization of the sample-based initial proposal selection method introduced in Section 5.3.3, we conduct 20 trial runs for each mission. The mean and standard deviation of the absolute error in each trial are reported.

The incorporation of the sonar structure from motion constraint is able to improve the localization accuracy in most of the mission trials correcting the drift error. The improvement becomes more obvious when the relative error is higher, mainly due to the limited resolution of the sonar image with respect to the corresponding physical sensor footprint. Failure cases exist where the sonar constraint is not able to improve the localization accuracy. For example, in **Mission 1**, the vehicle is performing a surface mission at the water line of the ship hull. The small true relative motions are overwhelmed by the noise of the pose constraints provided by the sonar. This is an example of how, given the properties of acoustic imaging, sonar constraints are only useful for recovering gross relative motion.

The SLAM trajectories for each method are given in Fig. 5.10 with sonar constraints

Table 5.2: SLAM estimation accuracy with respect to camera-image-based offline bundle adjustment for each mission. In this table we compare the localization accuracy of our proposed algorithm (Prop.) and the dead reckoning (DR). For the proposed method, the means (μ) and standard deviations (σ) of absolute error over 20 runs for each mission are given ($\mu[\sigma]$). The smallest error for each mission in each axis is in bold.

Mission		1	2	3	4	5	6
Error in X [m]	**Prop.**	**0.65[0.16]**	**0.39[0.13]**	**1.29[0.36]**	**0.39[0.15]**	**0.62[0.30]**	**2.19[0.25]**
	DR	0.79	0.52	1.64	0.68	1.06	2.44
Error in Y [m]	**Prop.**	**0.66[0.02]**	**0.74[0.11]**	0.57[0.10]	**1.49[0.27]**	**0.61[0.14]**	**1.26[0.31]**
	DR	0.70	0.90	**0.52**	1.64	0.62	1.70
Error in Z [m]	**Prop.**	1.30[0.24]	**0.42[0.07]**	0.97[0.09]	0.42[0.09]	**0.50[0.08]**	**2.08[0.23]**
	DR	**0.99**	0.50	**0.88**	**0.29**	0.57	2.57

depicted in Fig. 5.11. It can be seen that the benefit of the FLS local structure from motion front end becomes more obvious as the accumulated error increases. It should be noted that the corrections applied by the sonar are more modest than typical SLAM corrections using optical image approaches. This is mainly due to the much lower spatial resolution of the sonar, and this finding highlights that if water clarity permits, optical constraints should be leveraged with a high priority. Ultimately, we envision that the system should be able to make use of both constraints within different parts of the same mission depending on the

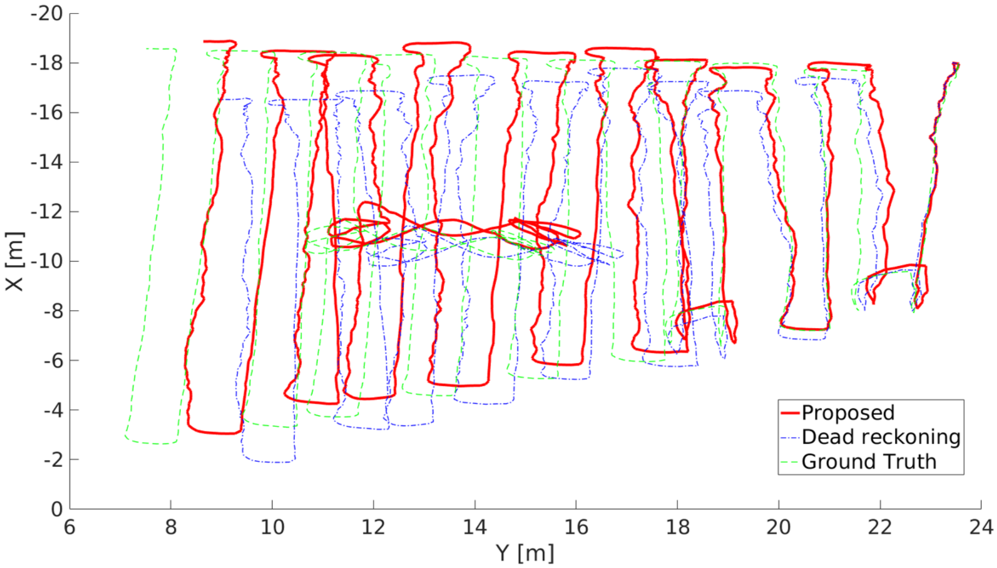


Figure 5.10: Trajectory estimation result of Mission 2. As the drift accumulates, the constraints from the sonar local structure from motion are able to correct the trajectory towards the ground truth.

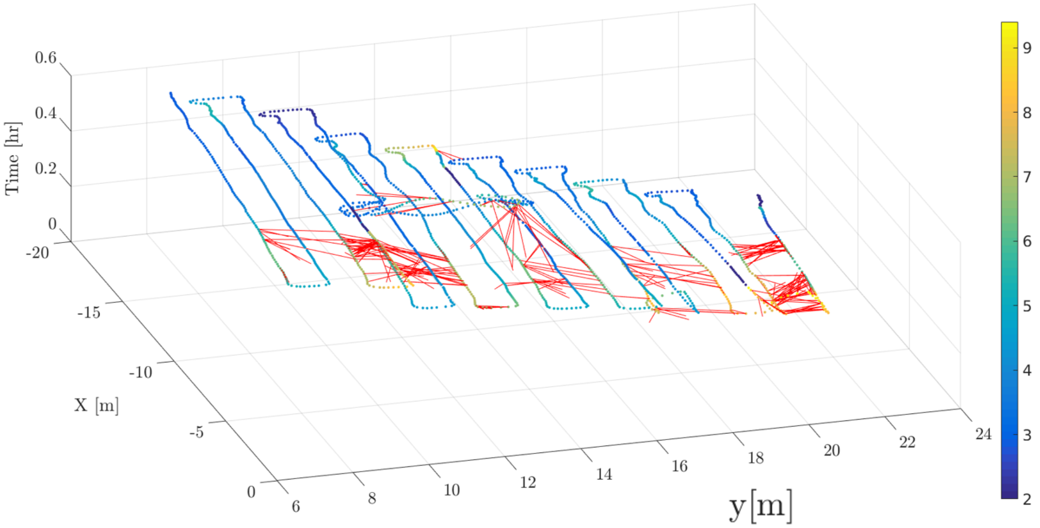


Figure 5.11: FLS structure from motion constraints added to SLAM graph. The results are shown in a time elevation map. Nodes in the trajectory are colored by the saliency score associated with the sonar frames.

environment, relative hull position, and imaging geometry.

5.5 Conclusion

In this chapter, we proposed a real-time acoustic SLAM system, which makes use of measurements from FLS images for underwater robot navigation. We have demonstrated a complete end-to-end acoustic SLAM system, particularly addressing the practical challenges of using FLS features, including data association, key frame selection and outlier rejection. The proposed method is evaluated on a real-world dataset collected for a ship hull inspection application. Experimental results indicate that our approach is able to efficiently propose information rich cliques of sonar frames and make use of them in a real-time SLAM system to correct the robot's position.

CHAPTER 6

Conclusion and Future Work

This thesis proposes advanced feature extraction and utilization approaches for both optical and acoustic sensors in underwater autonomous navigation systems, within a target application of ship hull inspection using a hovering autonomous underwater vehicle (HAUV). We have shown several contributions that use this information to improve underwater vehicle navigation, in both known and unknown environments. In this chapter, we succinctly summarize the contributions proposed in this thesis and provide a discussion on additional directions to further pursue our work.

6.1 Contributions

The contributions of this thesis include:

1. A high-level feature detection and description approach for robust registration between optical images against dramatic appearance changes. We efficiently detect visually salient patches through image segmentation and local region contrast evaluation. We achieve robust image registration results using model selection on matched salient patches. We compare our approach to a traditional feature approach using scale-invariant feature transform (SIFT) features in a random sample consensus (RANSAC) matching framework, and show our proposed method could achieve promising performance despite low feature density and appearance changes when the benchmark method underperforms;

2. Image-level descriptive features leveraging convolutional neural network (CNN) techniques for Forward-looking Sonar (FLS) images. We trained high dimensional features highly related to vehicle poses using a CNN over labeled data from an offline bundle adjustment (BA) algorithm. The trained features provide a robust and efficient similarity measurement under high-noise-level and weak-feature-texture

imaging conditions. The place recognition performance using the proposed feature achieved promising accuracy compared to traditional hand-crafted features. We have also integrated the proposed feature in a multi-session simultaneous localization and mapping (SLAM) registration framework with other sensor measurements and showed that our proposed feature is capable of improving the system's robustness against optical visual aliasing;

3. A general pre-processing procedure to utilize high-dimensional features in real-time Bayesian filtering frameworks. We reduce the curse of dimensionality of observation in recursive Bayesian estimations and provide a quantification to black-box feature uncertainty using data perturbation techniques. We have shown that the proposed approach can be applied in a real-time localization framework using high-dimensional sonar image features. In addition to acoustics, we have also applied the proposed approach on highly-abstracted visual features, which illustrates the generalization of the proposed approach; and finally,

4. A novel pose-graph SLAM algorithm using FLS images to provide pose constraints for vehicle ego-motion estimation. We leverage local bundle adjustment of FLS images to provide more robust constraints over vehicle poses compared to two-view registration. We have also proposed a sonar image feature matching algorithm using pose priors achievable from a SLAM system. We evaluated the proposed algorithm on real-world data collected in ship hull inspection missions, and showed the capability of the method in drift correction for autonomous underwater vehicles (AUVs).

6.2 Future Work

This thesis provides the foundation to many potential future work directions in underwater multi-modality navigation systems. In this section, we provide a brief discussion of some of the interesting potential directions:

Multi-robot navigation systems

Due to restriction of battery life, the applications of AUVs are limited to relatively small operation areas and mission times. Multi-robot cooperation systems could relax this restriction and increase the utility of AUVs. The proposed multi-session registration approach sets the scene for future multi-robot applications of the localization techniques.

In both Chapter 2 and Chapter 3, we show improvement in multi-session SLAM registration using optical camera images and FLS images. Although all the experiment settings considered in this document are single-vehicle systems, the proposed features and registration approaches can be extended to multi-vehicle navigation systems. The challenges of both heterogeneous platforms and sensors would need to be addressed to make this a reality.

Improved sonar local bundle adjustment with other sensors and information

In Chapter 5, we develop a novel pipeline of using FLS images in a SLAM front-end. A local bundle adjustment on FLS imagery provided the pose constraints. However, the high ambiguity of some motion patterns between sonar frames limited the effectiveness of the constraints in all cases. Given information from other sensors, the local bundle adjustment can be improved. Possible sources include distance measurements from Doppler velocity logs (DVLs), local scene structure reconstruction from stereo cameras or a CAD model of the ship hull. Difficulties of this potential extension will lie in the extrinsic calibration among different sensors.

Key-point feature learning in sonar images

In Chapter 5, we have shown that the sonar feature matching approach based on pose prior constraint and sonar field-of-view (FOV) could improve matching robustness. In the experimental implementation put forward in Chapter 5, Accelerated-KAZE (A-KAZE) is used in sonar feature matching. Although it has been discussed in [89] that this feature extractor presents better performance in sonar imagery compared to other widely-used visual features, they are not particularly designed for sonar image geometry.

Leveraging the work in Chapter 3, CNNs have demonstrated great power in learning a feature representation in FLS images. By harnessing such an approach on key-point feature representation learning, feature matching could be improved.

APPENDIX A

Epipolar geometry and Fundamental matrix

An illustration of geometry relationship between two pinhole cameras looking at one scene point is shown in Figure A.1. The image points P_L and P_R are sharing the same plane defined by scene point P and the camera projection centers O_L and O_R.

This planar constraint decreases the searching complexity for the corresponding point searching. Given one point in the left image P_L, the epipolar plane is defined by P_L, O_L and E_L. The corresponding point can only be found on the epipolar line on the right image, which is the line of intersection of the epipolar plane and the right image plane.

A mathematical description of the above constraint can be concluded by Fundamental matrix F:

$$X_L' F X_R = 0$$

$$F = K_L^{-T} R[t]_\times K_R^{-1}$$

X_L and X_R are homogeneous coordinates of a pair of corresponding points,and R and t are the rotation and transformation parameters between two camera coordinate systems.

In feature point matching, a fundamental matrix is often used to validate the matching pairs. However, in most of the cases, R and t are unknown. The most common practice is using RANSAC to estimate a best fundamental matrix. Matching pairs are randomly sampled to estimate candidate fundamental matrices and the one that fits the majority of matching pairs will be used. A large number of matching pairs and a sufficient ratio of true matching pairs are needed to guaranteed the validity of this method.

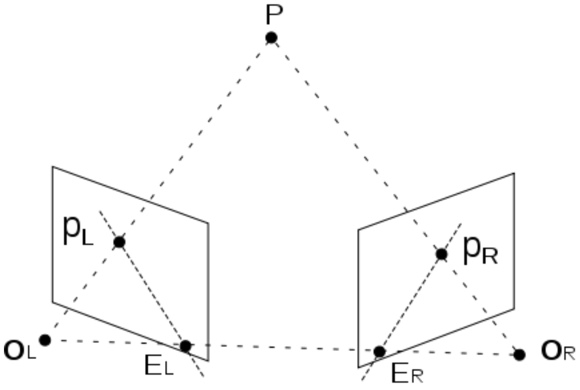


Figure A.1: An illustration of epipolar geometry relationship. The image points P_L and P_R share the same plane defined by scene point P and two camera projection centers O_L and O_R.

.

APPENDIX B

Support Vector Machines

Support vector machines (SVMs) are supervised learning models with associated learning algorithms that analyze data and recognize patterns, used for classification and regression analysis. Given a set of training examples, each labeled as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier.

As shown in Figure B.1, the main task of an SVM training algorithm is to estimate a boundary that separates the two categories into different sides in a feature space. SVMs can have varied forms depending on the feature space where its boundary is detected. Given a set of training samples:

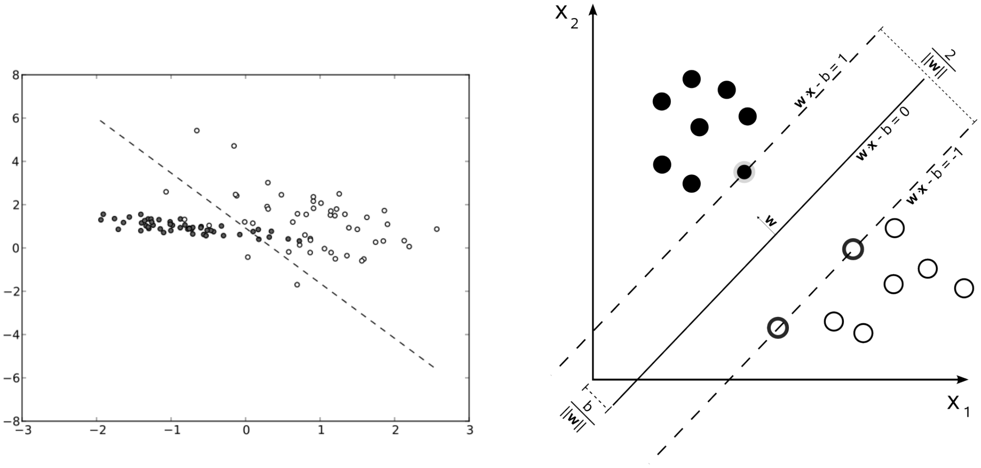
$$D = \{(x_i, y_i) | x_i \in R^n, y_i \in \{1, -1\}\}_{i=1}^{m}$$

The most standard linear SVMs separate the samples in their vector space R^n. If the training samples are linearly separable, a boundary can be found with the form: $\mathbf{wx} - b = 0$, so that:

$$\mathbf{w}x_i - b >= 0, y_i = 1;$$

$$\mathbf{w}x_i - b <= 0, y_i = -1;$$

When the training data is separable, an infinite number of boundary lines exist that hold the above constraint. The SVM training algorithm is a solution to the optimization problem that maximizes the margin of boundary to training samples, which is defined in Figure B.1(b). The definition of boundary margin indicates that only a small set of samples is related to the final decision of boundary; those samples are referred to as support vectors. This property determines that SVMs are less sensitive to the total amount of training samples and more robust to the problem of overfitting.



(a) SVM training

(b) Margin definition of SVM boundary

Figure B.1: The main task of an SVM training algorithm is to estimate a boundary that separates two different categories of samples in a feature space.

BIBLIOGRAPHY

[1] Stanford university CS231n: Convolutional neural networks for visual recognition. http://cs231n.github.io/.

[2] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL http://tensorflow.org/. Software available from tensorflow.org.

[3] R. Achanta, F. Estrada, P. Wils, and S. Süsstrunk. Salient region detection and segmentation. In *Computer Vision Systems*, pages 66–75. Springer, 2008.

[4] P. F. Alcantarilla and T. Solutions. Fast explicit diffusion for accelerated features in nonlinear scale spaces. *IEEE Trans. Patt. Anal. Mach. Intell*, 34(7):1281–1298, 2011.

[5] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pages 73–80, San Francisco, CA, USA, June 2010.

[6] B. Anderson and J. Crowell. Workhorse auv–a cost-sensible new autonomous underwater vehicle for surveys/soundings, search & rescue, and research. In *OCEANS, 2005. Proceedings of MTS/IEEE*, pages 1–6. IEEE, 2005.

[7] I. Arasaratnam and S. Haykin. Cubature kalman filters. *IEEE Transactions on automatic control*, 54(6):1254–1269, 2009.

[8] M. Aykin and S. Negahdaripour. On feature extraction and region matching for forward scan sonar imaging. In *Oceans, 2012*, pages 1–9, Oct 2012. doi: 10.1109/OCEANS.2012.6404983.

[9] M. D. Aykin and S. Negahdaripour. On feature matching and image registration for two-dimensional forward-scan sonar imaging. *Journal of Field Robotics*, 30(4): 602–623, 2013.

[10] A. Bachrach, S. Prentice, R. He, P. Henry, A. S. Huang, M. Krainin, D. Maturana, D. Fox, and N. Roy. Estimation, planning, and mapping for autonomous flight

using an rgb-d camera in gps-denied environments. *Int. J. Rob. Res.*, 31(11):1320–1343, Sept. 2012. ISSN 0278-3649. doi: 10.1177/0278364912455256. URL http://dx.doi.org/10.1177/0278364912455256.

[11] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*, 2015.

[12] T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (SLAM): Part II. *IEEE Robot. Autom. Mag.*, 13(3):108–117, 2006.

[13] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *Proc. European Conf. Comput. Vis.*, pages 404–417. Springer, 2006.

[14] T. Bengtsson, P. Bickel, B. Li, et al. Curse-of-dimensionality revisited: Collapse of the particle filter in very large scale systems. In *Probability and statistics: Essays in honor of David A. Freedman*, pages 316–334. Institute of Mathematical Statistics, 2008.

[15] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, June 2010. Oral Presentation.

[16] C. M. Bishop. Training with Noise is Equivalent to Tikhonov Regularization. *Neural Computation*, 7(1):108–116, jan 1995. ISSN 0899-7667.

[17] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

[18] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua. Brief: Computing a local binary descriptor very fast. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(7):1281–1298, 2012.

[19] N. Carlevaris-Bianco and R. M. Eustice. Multi-view registration for feature-poor underwater imagery. In *Proc. IEEE Int. Conf. Robot. and Automation*, pages 423–430, Shanghai, China, May 2011.

[20] S. M. Chaves, R. W. Wolcott, and R. M. Eustice. Neec research: Toward gps-denied landing of unmanned aerial vehicles on ships at sea. *Naval Engineers Journal*, 127 (1):23–35, 2015.

[21] K. Chen and L. Liu. Geometric data perturbation for privacy preserving outsourced data mining. *Knowledge and Information Systems*, 29(3):657–695, 2011.

[22] Z. Chen, O. Lam, A. Jacobson, and M. Milford. Convolutional neural network-based place recognition. *CoRR*, abs/1411.1509, 2014. URL http://arxiv.org/abs/1411.1509.

[23] R. Collobert and J. Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM, 2008.

[24] M. Cummins and P. Newman. Fab-map: Probabilistic localization and mapping in the space of appearance. *Int. J. Robot. Res.*, 27(6):647–665, 2008.

[25] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, volume 1, pages 886–893, San Diego, CA, USA, June 2005.

[26] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(6):1052–1067, 2007.

[27] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255, June 2009. doi: 10.1109/CVPR.2009.5206848.

[28] J. Duchi, P. L. Bartlett, and M. J. Wainwright. Randomized smoothing for stochastic optimization. *URL http://arxiv. org/abs/1103.4296*, 2011.

[29] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: Part I. *IEEE Robot. Autom. Mag.*, 13(2):99–110, 2006.

[30] C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Theoretical Computer Science*, 9(3-4):211–407, 2013.

[31] G. A. Einicke and L. B. White. Robust extended kalman filtering. *IEEE Transactions on Signal Processing*, 47(9):2596–2599, Sep 1999. ISSN 1053-587X. doi: 10.1109/78.782219.

[32] R. M. Eustice, O. Pizarro, and H. Singh. Visually augmented navigation for autonomous underwater vehicles. *IEEE J. Ocean. Eng.*, 33(2):103–122, 2008.

[33] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *Int. J. Comput. Vis.*, 59(2):167–181, 2004.

[34] Y. Gal and Z. Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. *arXiv preprint arXiv:1506.02142*, 2015.

[35] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[36] R. Gomez-Ojeda, M. Lopez-Antequera, N. Petkov, and J. G. Jiménez. Training a convolutional neural network for appearance-invariant place recognition. *CoRR*, abs/1505.07428, 2015. URL http://arxiv.org/abs/1505.07428.

[37] R. Hadsell, A. Erkan, P. Sermanet, M. Scoffier, U. Muller, and Y. LeCun. Deep belief net learning in a long-range vision system for autonomous off-road driving. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 628–633, Sept 2008. doi: 10.1109/IROS.2008.4651217.

[38] A. Handa, T. Whelan, J. McDonald, and A. Davison. A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In *IEEE Intl. Conf. on Robotics and Automation, ICRA*, Hong Kong, China, May 2014.

[39] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. In *Advances in neural information processing systems*, pages 545–552, 2006.

[40] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2003.

[41] J. Hassal and K. Zaveri. *Acoustic noise measurements*. Brüel & Kjaer, 1979.

[42] G. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[43] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *ArXiv preprint*, arXiv:1207.0580, 2012.

[44] J. Hosang, M. Omran, R. Benenson, and B. Schiele. Taking a deeper look at pedestrians. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4073–4082, 2015.

[45] F. Hover, R. Eustice, A. Kim, B. Englot, H. Johannsson, M. Kaess, and J. Leonard. Advanced perception, navigation and planning for autonomous in-water ship hull inspection. *Intl. J. of Robotics Research, IJRR*, 31(12):1445–1464, Oct. 2012.

[46] T. A. Huang and M. Kaess. Towards acoustic structure from motion for imaging sonar. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 758–765, Sept 2015. doi: 10.1109/IROS.2015.7353457.

[47] N. Hurtos, X. Cuf', Y. Petillot, and J. Salvi. Fourier-based registrations for two-dimensional forward-looking sonar image mosaicing. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 5298–5305, Oct 2012. doi: 10.1109/IROS.2012.6385813.

[48] N. Hurtos, S. Nagappa, X. Cufi, Y. Petillot, and J. Salvi. Evaluation of registration methods on two-dimensional forward-looking sonar imagery. In *OCEANS - Bergen, 2013 MTS/IEEE*, pages 1–8, June 2013. doi: 10.1109/OCEANS-Bergen.2013.6608124.

[49] N. Hurts, D. Ribas, X. Cuf, Y. Petillot, and J. Salvi. Fourier-based registration for robust forward-looking sonar mosaicing in low-visibility underwater environments. *Journal of Field Robotics*, 32(1):123–151, 2015. ISSN 1556-4967. doi: 10.1002/rob.21516. URL http://dx.doi.org/10.1002/rob.21516.

[50] V. Ila, J. M. Porta, and J. Andrade-Cetto. Information-based compact pose SLAM. *IEEE Transactions on Robotics*, 26(1):78–93, Feb 2010. ISSN 1552-3098. doi: 10.1109/TRO.2009.2034435.

[51] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[52] H. Jiang, J. Wang, Z. Yuan, T. Liu, N. Zheng, and S. Li. Automatic salient object segmentation based on context and shape prior. In *Proc. British Mach. Vis. Conf.*, volume 3, page 7, Dundee, U.K., Aug. 2011.

[53] H. Johannsson, M. Kaess, B. Englot, F. Hover, and J. J. Leonard. Imaging sonar-aided navigation for autonomous underwater harbor surveillance. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, pages 4396–4403, Taipei, Taiwan, Oct. 2010.

[54] M. Johnson-Roberson, C. Barto, R. Mehta, S. N. Sridhar, K. Rosaen, and R. Vasudevan. Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks? In *IEEE International Conference on Robotics and Automation*, pages 1–8, 2017.

[55] S. J. Julier and J. K. Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, Mar 2004. ISSN 0018-9219. doi: 10.1109/JPROC.2003.823141.

[56] A. T. Kalai and S. Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.

[57] A. Kendall and R. Cipolla. Modelling uncertainty in deep learning for camera relocalization. *CoRR*, abs/1509.05909, 2015. URL http://arxiv.org/abs/1509.05909.

[58] A. Kendall, M. Grimes, and R. Cipolla. Convolutional networks for real-time 6-dof camera relocalization. *arXiv preprint arXiv:1505.07427*, 2015.

[59] A. Kim and R. M. Eustice. Real-time visual SLAM for autonomous underwater hull inspection using visual saliency. *IEEE Trans. Robot.*, 29(3):719–733, 2013.

[60] K. Konolige and M. Agrawal. FrameSLAM: From bundle adjustment to real-time visual mapping. *IEEE Trans. Robot.*, 24(5):1066–1077, 2008.

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

[62] Y. LeCun and C. Cortes. MNIST handwritten digit database. *http://yann.lecun.com/exdb/mnist/*, 2010. URL http://yann.lecun.com/exdb/mnist/.

[63] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998. ISSN 0018-9219. doi: 10.1109/5.726791.

[64] J. J. Leonard and H. F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *Robotics and Automation, IEEE Transactions on*, 7(3):376–382, 1991.

[65] J. Li, R. M. Eustice, and M. Johnson-Roberson. High-level visual features for underwater place recognition. In *Proc. IEEE Int. Conf. Robot. and Automation*, pages 3652–3659, May 2015. doi: 10.1109/ICRA.2015.7139706.

[66] J. Li, R. M. Eustice, and M. Johnson-Roberson. Underwater robot visual place recognition in the presence of dramatic appearance change. In *Proc. IEEE/MTS OCEANS Conf. Exhib.*, pages 1–6, Oct 2015. doi: 10.23919/OCEANS.2015.7404369.

[67] J. Li, P. Ozog, J. Abernethy, R. M. Eustice, and M. Johnson-Roberson. Utilizing high-dimensional features for real-time robotic applications: Reducing the curse of dimensionality for recursive bayesian estimation. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, pages 1230–1237, Oct 2016. doi: 10.1109/IROS.2016.7759205.

[68] J. Li, M. Kaess, R. M. Eustice, and M. Johnson-Roberson. Forward-looking sonar pose-graph SLAM. In *Proc. Int. Symp. Robot. Res.*, 2017. Submitted.

[69] A. Liaw and M. Wiener. Classification and regression by randomforest. *R news*, 2(3): 18–22, 2002.

[70] C. D. Loggins. A comparison of forward-looking sonar design alternatives. In *MTS/IEEE Oceans 2001. An Ocean Odyssey. Conference Proceedings (IEEE Cat. No.01CH37295)*, volume 3, pages 1536–1545 vol.3, 2001. doi: 10.1109/OCEANS.2001.968061.

[71] D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. European Conf. Comput. Vis.*, volume 2, pages 1150–1157, 1999.

[72] D. G. Lowe. Object recognition from local scale-invariant features. In *Proc. European Conf. Comput. Vis.*, volume 2, pages 1150–1157, 1999.

[73] W. Luo, A. G. Schwing, and R. Urtasun. Efficient deep learning for stereo matching. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[74] D. C. Lynn and G. S. Bohlander. Performing ship hull inspections using a remotely operated vehicle. In *OCEANS '99 MTS/IEEE. Riding the Crest into the 21st Century*, volume 2, pages 555–562 vol.2, 1999. doi: 10.1109/OCEANS.1999.804763.

[75] M. J. Milford and G. F. Wyeth. SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights. In *Proc. IEEE Int. Conf. Robot. and Automation*, pages 1643–1649, Saint Paul, MN, USA, May 2012.

[76] M. A. Moline, S. M. Blackwell, C. Von Alt, B. Allen, T. Austin, J. Case, N. Forrester, R. Goldsborough, M. Purcell, and R. Stokey. Remote environmental monitoring units: An autonomous vehicle for characterizing coastal environments. *Journal of Atmospheric and Oceanic Technology*, 22(11):1797–1808, 2005.

[77] T. Naseer, L. Spinello, W. Burgard, and C. Stachniss. Robust visual robot localization across seasons using network flows. In *Proc. AAAI Nat. Conf. Artif. Intell.*, pages 2564–2570, Québec City, Québec, Canada, July 2014. URL http://www.informatik.uni-freiburg.de/~naseer/publications/naseer14aaai.pdf.

[78] S. Negahdaripour. On 3-d motion estimation from feature tracks in 2-d fs sonar video. *Robotics, IEEE Transactions on*, 29(4):1016–1030, 2013.

[79] Y. Nesterov. Smoothing technique and its applications in semidefinite optimization. *Mathematical Programming*, 110(2):245–259, 2007.

[80] P. Neubert, N. Sunderhauf, and P. Protzel. Appearance change prediction for long-term navigation across seasons. In *Mobile Robots (ECMR), 2013 European Conference on*, pages 198–203, 2013.

[81] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *Int. J. Comput. Vis.*, 42(3):145–175, 2001.

[82] P. Ozog and R. M. Eustice. Toward long-term, automated ship hull inspection with visual SLAM, explicit surface optimization, and generic graph-sparsification. In *Proc. IEEE Int. Conf. Robot. and Automation*, pages 3832–3839, Hong Kong, China, June 2014.

[83] P. Ozog and R. M. Eustice. Large-scale model-assisted bundle adjustment using gaussian max-mixtures. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5576–5581, May 2016. doi: 10.1109/ICRA.2016.7487775.

[84] P. Ozog, G. Troni, M. Kaess, R. M. Eustice, and M. Johnson-Roberson. Building 3d mosaics from an autonomous underwater vehicle, Doppler velocity log, and 2d imaging sonar. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1137–1143, Seattle, WA, USA, May 2015.

[85] P. Ozog, M. Johnson-Roberson, and R. M. Eustice. Mapping underwater ship hulls using a model-assisted bundle adjustment framework. *Robotics and Autonomous Systems, Special Issue on Localization and Mapping in Challenging Environments*, 87:329–347, 2017.

[86] F. Perazzi, P. Krahenbuhl, Y. Pritch, and A. Hornung. Saliency filters: Contrast based filtering for salient region detection. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pages 733–740, Providence, RI, USA, June 2012.

[87] H. Rabbani, M. Vafadust, P. Abolmaesumi, and S. Gazor. Speckle noise reduction of medical ultrasound images in complex wavelet domain using mixture priors. *IEEE transactions on biomedical engineering*, 55(9):2152–2160, 2008.

[88] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

[89] 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*, pages 1–6, Oct 2015. doi: 10.23919/OCEANS.2015.7401963.

[90] Sound Metrics Corp. Didson 300 m Imaging Sonar. Specification sheet and documentations Available at "http://www.soundmetrics.com", 2014.

[91] N. Sünderhauf, S. Shirazi, A. Jacobson, F. Dayoub, E. Pepperell, B. Upcroft, and M. Milford. Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free. In *Robotics: Science and Systems*, Auditorium Antonianum, Rome, July 2015. URL http://eprints.qut.edu.au/84931/.

[92] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014.

[93] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust monte carlo localization for mobile robots. *Artificial intelligence*, 128(1):99–141, 2001.

[94] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005. ISBN 0262201623.

[95] S. Wager, S. Wang, and P. Liang. Dropout Training as Adaptive Regularization. In *Proceedings of Neural Information Processing Systems (NIPS)*, 2013.

[96] M. Walter, F. Hover, and J. Leonard. Slam for ship hull inspection using exactly sparse extended information filters. In *2008 IEEE International Conference on Robotics and Automation*, pages 1463–1470, May 2008. doi: 10.1109/ROBOT.2008.4543408.

[97] S. Wang and C. Manning. Fast dropout training. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 118–126, 2013.

[98] M. Welling, M. Rosen-zvi, and G. E. Hinton. Exponential family harmoniums with an application to information retrieval. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1481–1488. MIT Press, 2005.

[99] S. B. Williams, O. R. Pizarro, M. V. Jakuba, C. R. Johnson, N. S. Barrett, R. C. Babcock, G. A. Kendrick, P. D. Steinberg, A. J. Heyward, P. J. Doherty, I. Mahon, M. Johnson-Roberson, D. Steinberg, and A. Friedman. Monitoring of benthic reference sites: Using an autonomous underwater vehicle. *IEEE Robotics Automation Magazine*, 19(1):73–84, March 2012. ISSN 1070-9932. doi: 10.1109/MRA.2011.2181772.

[100] R. W. Wolcott and R. M. Eustice. Visual localization within lidar maps for automated urban driving. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 176–183. IEEE, 2014.

[101] F. Yang, W. Choi, and Y. Lin. Exploit all the layers: Fast and accurate CNN object detector with scale dependent pooling and cascaded rejection classifiers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2129–2137, 2016.

[102] Y. Yang and G. Huang. Acoustic-inertial underwater navigation. In *Proc. of the IEEE International Conference on Robotics and Automation*, Singapore, 2017.

[103] S. Yoon and C. Qiao. Cooperative search and survey using autonomous underwater vehicles (auvs). *IEEE Transactions on Parallel and Distributed Systems*, 22(3): 364–379, 2011.

[104] P. Zarchan. *Progress In Astronautics and Aeronautics: Fundamentals of Kalman Filtering: A Practical Approach*, volume 208. Aiaa, 2005.

[105] Y. Zhang, K. Sohn, R. Villegas, G. Pan, and H. Lee. Improving object detection with deep convolutional networks via bayesian optimization and structured prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 249–258, 2015.