# Active Visual SLAM with Exploration for Autonomous Underwater Navigation

by

Ayoung Kim

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Mechanical Engineering)
in The University of Michigan
2012

Doctoral Committee:

Assistant Professor Ryan M. Eustice, Chair
Professor Huei Peng
Professor Jing Sun
Assistant Professor Silvio Savarese

# Report Documentation Page

*Form Approved*
*OMB No. 0704-0188*



| 1. REPORT DATE | 2. REPORT TYPE | 3. DATES COVERED |
|---|---|---|
| **2012** | | **00-00-2012 to 00-00-2012** |

| 4. TITLE AND SUBTITLE | 5a. CONTRACT NUMBER |
|---|---|
| **Active Visual SLAM with Exploration for Autonomous Underwater Navigation** | 5b. GRANT NUMBER |
| | 5c. PROGRAM ELEMENT NUMBER |
| 6. AUTHOR(S) | 5d. PROJECT NUMBER |
| | 5e. TASK NUMBER |
| | 5f. WORK UNIT NUMBER |
| 7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of Michigan,Ann Arbor,MI,48109** | 8. PERFORMING ORGANIZATION REPORT NUMBER |
| 9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) | 10. SPONSOR/MONITOR'S ACRONYM(S) |
| | 11. SPONSOR/MONITOR'S REPORT NUMBER(S) |

12. DISTRIBUTION/AVAILABILITY STATEMENT
**Approved for public release; distribution unlimited**

13. SUPPLEMENTARY NOTES

14. ABSTRACT

**One of the major challenges in the field of underwater robotics is the opacity of the water medium to radio frequency transmission modes, which precludes the use of a global positioning system (GPS) and high speed radio communication in underwater navigation and mapping applications. One approach to underwater robotics that overcomes this limitation is vision-based simultaneous localization and mapping (SLAM), a framework that enables a robot to localize itself, while simultaneously building a map of an unknown environment. The SLAM algorithm provides a probabilistic map that contains the estimated state of the system, including a map of the environment and the pose of the robot. Because the quality of vision-based navigation varies spatially within the environment the performance of visual SLAM strongly depends on the path and motion that the robot follows. While traditionally treated as two separate problems, SLAM and path planning are indeed interrelated: the performance of SLAM depends significantly on the environment and motion; however, control of the robot motion fully depends on the information from SLAM. Therefore, an integrated SLAM control scheme is needed?one that can direct motion for better localization and mapping, and thereby provide more accurate state information back to the controller. This thesis develops perception-driven control, an integrated SLAM and path planning framework that improves the performance of visual SLAM in an informative and efficient way by jointly considering the reward predicted by a candidate camera measurement, along with its likelihood of success based upon visual saliency. The proposed control architecture identifies highly informative candidate locations for SLAM loop-closure that are also visually distinctive, such that a camera-derived pose-constraint is probable. Results are shown for autonomous underwater hull inspection experiments using the Bluefin Robotics Hovering Autonomous Underwater Vehicle (HAUV).**

15. SUBJECT TERMS

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON |
|---|---|---|---|---|---|
| a. REPORT<br>**unclassified** | b. ABSTRACT<br>**unclassified** | c. THIS PAGE<br>**unclassified** | **Same as Report (SAR)** | **168** | |

**Standard Form 298 (Rev. 8-98)**
Prescribed by ANSI Std Z39-18

# ACKNOWLEDGEMENTS

There have been so many incredible moments that still remain vivid in my memory. I would like to thank all of the people who made this possible.

First of all, and foremost, I greatly appreciate the detailed and brilliant advice from my research advisor ,Professor Ryan Eustice, that enabled me to complete this 5 year program. Ive been climbing up this mountain to reach up to this point with his advice as stepping stones for all those years. He always has been supportive and willing to discuss on issues teaching me the view that I need as a researcher.

I would like to thank you, Edward Mahony, with whom we completed building two AUVs. I also thank all of the PeRL members, Gaurav, Nick, Jeff, Paul, Steve, and Ryan Wolcott, who have been open for valuable discussion and questions. Chatting with them enabled me to escape from long, unsolved mysteries.

It was so great for me to meet the brilliant group doing the hull inspection project. I thank Dr. Michael Kaess, Hordur Johannsson, Dr. Brendan Englot, Professor Franz Hover, and Professor John Leonard. The project would not have been successful without help from Bluefin robotics, Dr. Jerome Vaganay and Kimberly Shurn.

I would have not reached this level of academic achievement without support from my family. Even though they are far away, they have continually encouraged me and greatly supported me. The many days and nights Google chatting with my Dad and doing Facetime with my Mom will never be forgotten. I also would like to thank my two roommates, Jinyoung and Minyoung, and Jinyoungs two lovely dogs, Sam and Teddy, for the uncountable moments of giving their emotional support. I was so lucky to have them in my life. During this time while we all are far away from home, we have lived like a family.

## Funding

This work is supported through grants from the Office of Naval Research (ONR) (Award #N00014-07-1-0791 and #N00014-12-10092).

# TABLE OF CONTENTS

**ACKNOWLEDGEMENTS** . . . . . ii

**LIST OF FIGURES** . . . . . vi

**LIST OF TABLES** . . . . . viii

**LIST OF APPENDICES** . . . . . ix

**ABSTRACT** . . . . . x

**CHAPTER**

**I. Introduction** . . . . . 1

1.1 Literature Review on Computer Vision . . . . . 4
1.1.1 Structure from Motion . . . . . 4
1.1.2 Visual Odometry . . . . . 6
1.1.3 Next-Best-View . . . . . 6
1.1.4 Place Recognition . . . . . 7
1.1.5 Challenges in Underwater Vision . . . . . 8
1.2 Literature Review on Underwater Navigation . . . . . 9
1.2.1 Beacon-based Positioning Systems . . . . . 9
1.2.2 Doppler velocity log (DVL) . . . . . 11
1.3 Literature Review on SLAM . . . . . 12
1.3.1 Large Scale SLAM . . . . . 13
1.3.2 Reliable Data Association . . . . . 15
1.3.3 Underwater Visual SLAM . . . . . 16
1.4 Literature Review on Path Planning and Visual Servoing . . . . . 17
1.4.1 Path Planning . . . . . 17
1.4.2 Visual Servoing . . . . . 18
1.5 Thesis Outline . . . . . 18

**II. Real-time Pose-graph Visual SLAM** . . . . . 19

2.1 In-water Hull Inspection using an AUV . . . . . 19

2.1.1 Challenges and Approaches . . . 20
2.2 System Overview . . . 21
2.3 Two-view Camera Registration Engine . . . 24
2.3.1 Feature Extraction . . . 25
2.3.2 Pose-Constrained Correspondence Search . . . 25
2.3.3 Geometric Model Selection . . . 26
2.3.4 Two-view Bundle Adjustment . . . 29
2.4 Implementation and Results . . . 33
2.5 Conclusion . . . 35

**III. Visual Saliency for SLAM** . . . 37

3.1 Introduction . . . 37
3.1.1 Motivation . . . 37
3.1.2 Review on Saliency and Bag-of-Words . . . 40
3.1.3 Overview of Our Approach . . . 41
3.1.4 BoW Vocabulary Generation . . . 41
3.2 Saliency . . . 47
3.2.1 Local Saliency . . . 47
3.2.2 Global Saliency . . . 48
3.3 Saliency-informed Visual SLAM . . . 52
3.3.1 Salient Keyframe Selection . . . 53
3.3.2 Saliency Incorporated Link Hypothesis . . . 55
3.4 Saliency Results . . . 57
3.4.1 Local and Global Saliency Maps . . . 57
3.4.2 Saliency-informed SLAM . . . 61
3.5 Conclusion . . . 67

**IV. Perception-driven Navigation** . . . 68

4.1 Related Work . . . 69
4.2 Motivation . . . 74
4.2.1 Conventional Preplanned Survey Design . . . 74
4.2.2 Revisit Planning / Control for Loop-closures . . . 77
4.3 Perception-driven Navigation . . . 79
4.3.1 Waypoint Generation . . . 81
4.3.2 Path Generation . . . 86
4.3.3 Reward for a Path . . . 89
4.4 Results . . . 102
4.4.1 Simulation Setup . . . 102
4.4.2 PDN Parameters . . . 105
4.4.3 PDN with Synthetic Saliency Map . . . 106
4.4.4 PDN with Real Image Data . . . 111
4.5 Conclusion . . . 114

**V. Conclusion** . . . . . 115

5.1 Contributions . . . . . 115
5.2 Future Work . . . . . 116

**APPENDICES** . . . . . 118

**BIBLIOGRAPHY** . . . . . 140

# LIST OF FIGURES

**Figure**


1.1 Examples of AUV exploration . . . 2
1.2 Sample images for feature-poor and feature-rich hulls . . . 3
1.3 Structure from motion examples . . . 5
1.4 Challenges in underwater images . . . 10
1.5 Typical underwater beacon acoustic positioning systems . . . 11
1.6 Photo of DVL mounted on two underwater vehicles . . . 12
1.7 Large scale SLAM . . . 14
1.8 Examples of underwater visual SLAM . . . 16
2.1 Illustration of the in-water hull inspection application using an AUV . . . 20
2.2 5-DOF camera measurement model . . . 22
2.3 Depiction of the pose-graph SLAM constraint graph . . . 23
2.4 Block-diagram of the core image registration engine . . . 24
2.5 Two factors in successful registration: prior uncertainty and saliency . . . 27
2.6 Accuracy in inliers selection for homography versus essential matrix . . . 28
2.7 Geometric Information Criterion (GIC) on synthetic data . . . 30
2.8 Geometric Information Criterion (GIC) on real data . . . 31
2.9 Illustration of sparse bundle adjustment (SBA) . . . 32
2.10 Experimental setup for *USS Saratoga* hull inspection survey using the HAUV . . . 33
2.11 SLAM result from the *USS Saratoga* . . . 34
2.12 Photomosaic results on the *USS Saratoga* . . . 35
2.13 Uncertainty versus path length plot . . . 36
3.1 Local and global saliency maps on *R/V Oceanus* . . . 38
3.2 Depiction of vocabulary construction and saliency computation . . . 42
3.3 Illustration of SIFT and SURF descriptors . . . 45
3.4 Effect of pre-blurring versus scale-forced SURF . . . 46
3.5 Online vocabulary size over the course of a hull inspection mission . . . 47
3.6 Local saliency example for color and grayscale ship hull images . . . 49
3.7 Global saliency example for underwater and indoor images . . . 52
3.8 Local saliency versus navigation prior uncertainty . . . 54
3.9 Illustration of link proposal using saliency . . . 56
3.10 Experimental setup for saliency evaluation . . . 57
3.11 Local and global saliency maps on the *USCGC Venturous* . . . 58

3.12 Local and global saliency maps on the *SS Curtiss* . . . 60
3.13 Exhaustive SLAM for the *SS Curtiss* . . . 63
3.14 Saliency-informed SLAM result for the *SS Curtiss* . . . 64
3.15 Statistics on saliency-informed SLAM for the *SS Curtiss* . . . 66
4.1 Related work to perception-driven navigation (PDN) . . . 69
4.2 A typical underwater grid pattern mission for area coverage . . . 75
4.3 Calculation of camera sensor overlap ratio . . . 76
4.4 Illustration of PDN . . . 77
4.5 Two mission profiles in used in PDN's evaluation: camera and sonar . . . 78
4.6 Illustration of PDN's flow diagram . . . 80
4.7 Online clustering for a densely-spaced camera mission . . . 84
4.8 Online clustering for a sparsely-sampled sonar mission . . . 85
4.9 Waypoint selection for two typical mission profiles (camera and sonar) . . . 86
4.10 Point-to-point path planning for camera and sonar missions . . . 89
4.11 Empirical probability of link success for visual SLAM surveys on three different vessels . . . 91
4.12 Construction of empirical probability of successful link, $P_L$ . . . 92
4.13 Robot pose uncertainty propagation . . . 95
4.14 Robot pose uncertainty from revisiting versus exploration . . . 99
4.15 Comparison of PDN's robot uncertainty calculation relative to previous work . . . 100
4.16 Simulation setup for PDN evaluation . . . 103
4.17 Pose-uncertainty-only PDN with synthetic saliency distribution . . . 104
4.18 Target coverage area calculation . . . 106
4.19 PDN results for evenly distributed saliency map . . . 107
4.20 PDN results for biased saliency maps . . . 108
4.21 PDN-aided SLAM trajectory behavior with respect to $\alpha$ . . . 110
4.22 PDN for saliency-informed SLAM on a sonar mission . . . 112
4.23 PDN for saliency-informed SLAM on a camera mission . . . 113
A.1 Hull inspection using sonar and camera . . . 120
A.2 Two different camera configurations for the HAUV . . . 121
A.3 Real-time SLAM software architecture . . . 123
A.4 Local, vehicle and sensor coordinate frames . . . 124
A.5 5-DOF camera measurement model . . . 126
B.1 Camera uncertainty model . . . 130
B.2 Example trajectory design paths . . . 132
B.3 CRLB based contour plot . . . 133
C.1 2D Covariance propagation example of three poses . . . 136

# LIST OF TABLES

**Table**


1.1 Feature detectors and descriptors . . . . . 7
1.2 600 kHz DVL single-ping precision of bottom-track velocity . . . . . 12
3.1 Improvement summary of using saliency-informed simultaneous localization and mapping (SLAM) . . . . . 65
3.2 Link proposal statistics for saliency-informed SLAM . . . . . 67
4.1 Summary of related works to PDN . . . . . 73
A.1 Specifications of major HAUV components . . . . . 122

# LIST OF APPENDICES

**Appendix**


A. Implementation Details of In-water Hull Inspection . . . . . 119
A.1. Hovering Autonomous Underwater Vehicle . . . . . 120
A.2. SLAM Software Architecture . . . . . 123
A.3. Robot / Sensor Coordinate Frames . . . . . 124
A.4. 5-DOF Camera Measurement . . . . . 125
B. Survey Design using the Cramer Rao Lower Bound (CRLB) . . . . . 128
B.1. Cramer Rao Lower Bound . . . . . 128
B.2. Modeling of the Camera Measurement . . . . . 129
B.3. Fisher Information Matrix . . . . . 130
B.4. Effect of the Design Parameters . . . . . 131
B.5. Conclusion . . . . . 133
C. Monotonicity of Covariance Propagation . . . . . 135

# ABSTRACT

One of the major challenges in the field of underwater robotics is the opacity of the water medium to radio frequency transmission modes, which precludes the use of a global positioning system (GPS) and high speed radio communication in underwater navigation and mapping applications. One approach to underwater robotics that overcomes this limitation is vision-based simultaneous localization and mapping (SLAM), a framework that enables a robot to localize itself, while simultaneously building a map of an unknown environment. The SLAM algorithm provides a probabilistic map that contains the estimated state of the system, including a map of the environment and the pose of the robot.

Because the quality of vision-based navigation varies spatially within the environment, the performance of visual SLAM strongly depends on the path and motion that the robot follows. While traditionally treated as two separate problems, SLAM and path planning are indeed interrelated: the performance of SLAM depends significantly on the environment and motion; however, control of the robot motion fully depends on the information from SLAM. Therefore, an integrated SLAM control scheme is needed—one that can direct motion for better localization and mapping, and thereby provide more accurate state information back to the controller.

This thesis develops perception-driven control, an integrated SLAM and path planning framework that improves the performance of visual SLAM in an informative and efficient way by jointly considering the reward predicted by a candidate camera measurement, along with its likelihood of success based upon visual saliency. The proposed control architecture identifies highly informative candidate locations for SLAM loop-closure that are also visually distinctive, such that a camera-derived pose-constraint is probable. Results are shown for autonomous underwater hull inspection experiments using the Bluefin Robotics Hovering Autonomous Underwater Vehicle (HAUV).

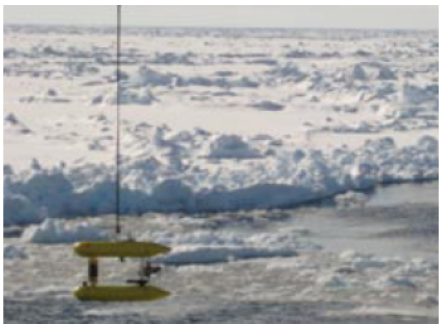
# CHAPTER I

# Introduction

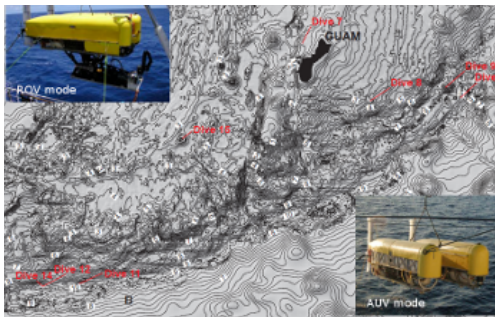
Autonomous underwater vehicles (AUVs) have played an important role in scientific research due to their ability to collect data in underwater environments that are either inaccessible or too dangerous for humans to explore as shown in Figure 1.1. Some notable examples include AUV arctic exploration under the polar ice in search of hydrothermal vents (Kunz et al., 2009) and the exploration of the Mariana Trench using a hybrid autonomous underwater vehicle (AUV) / remotely operated vehicle (ROV) to reach the ocean's deepest depth (Bowen et al., 2009). Other than explorations, underwater structures such as dams, ship hulls, harbors and pipelines also need to be periodically inspected for assessment, maintenance and security reasons. Autonomous vehicles have the potential for better coverage efficiency, improved survey precision and overall reduced need for human intervention. Ridao et al. (2010) reported on automated dam inspection using an autonomous underwater vehicle, focusing on mapping the in-water portion of the dam. Pipeline, cable and in-water supporting structure inspection and maintenance has been studied by Bradbeer et al. (1997), Paim et al. (2005), Curti et al. (2005), Gustafson et al. (2011), Maki et al. (2012). Similar to other underwater structures, in-water hull inspection using autonomous systems has been studied since it was identified as a core technology in 1992 within the Naval community (Bohlander et al., 1992). Effort in this area has resulted in the development of a number of autonomous hull inspection platforms such as those reported by Trimble and Belcher (2002), Menegaldo et al. (2009), Negahdaripour and Firoozfam (2006), Vaganay et al. (2009), and Hover et al. (2012).

In comparison to terrestrial navigation, underwater navigation is challenging because the opacity of water to electromagnetic waves precludes the use of the global positioning system (GPS) and other high speed radio communication. Due to the lack of accurate position information from GPS, traditional underwater navigation methods (e.g., Doppler velocity log (DVL) and long-baseline (LBL) systems) have been used to solve the navigation problem using acoustic signals. Both DVL (Brokloff, 1994) and LBL systems (Austin

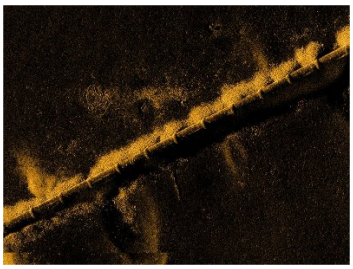
**Figure 1.1** Examples of AUV exploration. (a) Arctic exploration (Kunz et al., 2009). (b) Mariana Trench exploration (Bowen et al., 2009). (c) Autonomous pipeline inspection (Gustafson et al., 2011). (d) Autonomous dam inspection (Ridao et al., 2010). (e) Autonomous hull inspection (Hover et al., 2012).



(a) Arctic exploration




(b) Mariana Trench exploration



(c) Pipeline inspection


(d) Dam inspection

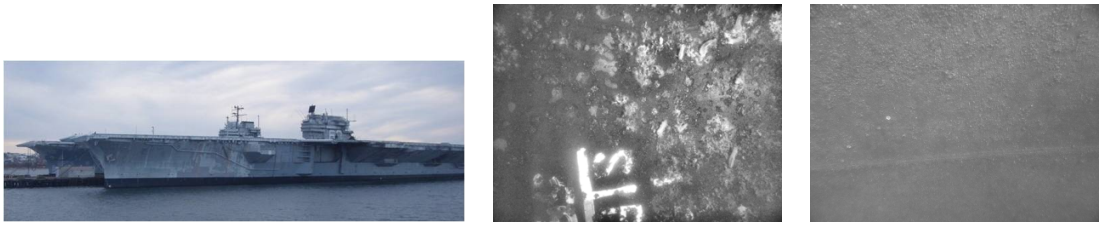
(e) Hull inspection

et al., 1984) have drawbacks; specifically LBL systems require infrastructure and DVL-only navigation exhibits monotonically increasing uncertainty.

Traditionally, navigation limitations can be overcome by using a simultaneous localization and mapping (SLAM) algorithm to fuse sensor measurements derived from vision and/or sonar. Similar to human navigation, in which sight confirms our position when we recognize a previously visited scene, visual measurements significantly reduce the uncertainty when a site is revisited and recognized (i.e., loop-closing). The advantage of visual SLAM arises from these loop-closure camera measurements, which add independent constraints to the pose-graph (will be discussed in §II), and greatly reduce position uncertainty as compared to pure odometry (dead reckoning).

Despite this major contribution in reducing uncertainty, visual measurements may not be uniformly available in an underwater environment where the spatial feature distribution varies greatly (Figure 1.2). This indicates that successful measurements strongly depend upon two factors: (*i*) the saliency of visual features and (*ii*) their spatial distribution as seen by the robot. The first factor, saliency, is an image measurement that represents distinguishability of a visual feature (§III). The second factor, the observed spatial distribution,

**Figure 1.2** Sample images for feature-poor and feature-rich hulls. Underwater images of a ship hull show diverse underwater feature distributions ranging from feature-rich regions (b) to feature-less regions (c). The information on this feature distribution is not known a priori, and significantly influences the success of camera measurements.



(a) *USS Saratoga* (b) Feature-rich image (c) Feature-less image

is mainly determined by the environment and egomotion of the robot (e.g., path and gaze). A similar interrelation has been proposed by Sim and Roy (2005) who found that changes in trajectory can result in better navigation. However, we should note that these two factors should be considered simultaneously for better navigation results, especially for underwater images where the possibility of making a valid registration may not be as uniform as in the terrestrial environment. Based upon this motivation, this thesis's goal has been to develop a control scheme for better navigation by providing trajectory perturbations to improve pose observability under the consideration of a visual saliency map.

Although SLAM has been developed in many different contexts and demonstrated successfully in many real-world implementations, there still remains an analytical gap between our understanding of navigation performance and its relation to the trajectory of the robot. Because a trajectory that leads a robot in the direction of informative and likely loop-closing will significantly reduce navigation uncertainty, the motion of the robot is critical to the navigation performance. This thesis addresses this gap by focusing on the development of a vision-based pose-graph SLAM algorithm that couples perception with control in a direct way to improve navigation and control performance simultaneously.

Aiming toward this goal, the expected contributions of this thesis are:

- Real-time visual SLAM is developed and successfully applied to a real-world AUV ship hull inspection application (Chapter II). Specifically, a monocular camera image registration engine is developed and integrated into a real-time SLAM implementation.

- Developed two novel measures of visual saliency that improve underwater visual SLAM (Chapter II). Local saliency measures texture richness of a scene and aids keyframe selection. Global saliency detects rarity of a scene (Chapter III).

- A novel solution for concurrent SLAM and planning for the robotic area coverage problem, called perception-driven navigation (PDN) (Chapter IV) is developed. PDN is an integrated navigation algorithm that automatically achieves efficient target area coverage while maintaining good visual SLAM navigation performance. PDN provides an intelligent and fully autonomous online control scheme for efficient bounded-error area coverage that strikes a balance between revisit and exploration actions in a decision theoretic way.

The following sections will review the relevant literature related to vision-based SLAM. Since vision-based underwater navigation is closely related to computer vision, especially in such areas as structure-from-motion and visual odometry, the literature in this area will be reviewed first. Subsequently, literature on traditional underwater acoustic navigation methods, followed by both filter-based and optimization-based SLAM, will be reviewed. Finally, areas relevant to perception driven navigation will be reviewed, such as path planning and visual servoing.

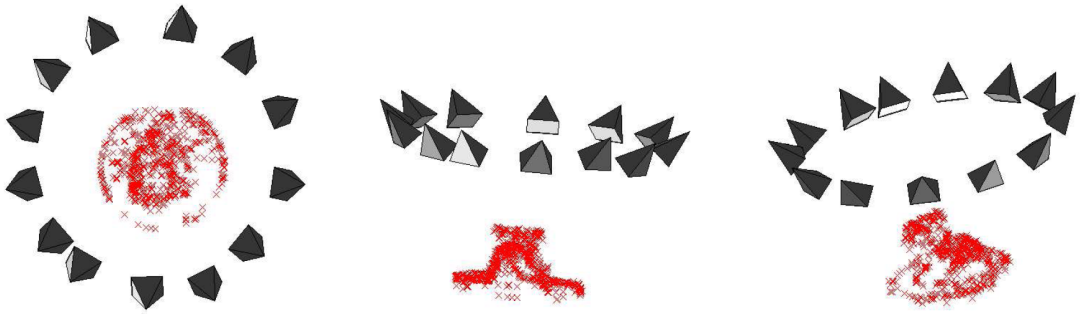
## 1.1 Literature Review on Computer Vision

Visual perception is one of the main components in this thesis; thus this section briefly reviews computer vision as it pertains to visual SLAM. The two major vision-based navigation approaches are: (*i*) filter-based navigation and (*ii*) optimization-based navigation. Both approaches are closely related to structure-from-motion (SFM) and visual odometry (VO) in computer vision. For both navigation techniques, data association (i.e., detecting revisited areas) is one of the main issues. Thus, this section will also review research on place recognition and its close relation to data association. Although most algorithms in this section apply to general images, we should note that underwater imaging has its own unique challenges as compared to terrestrial imaging due to the exponential attenuation of light underwater (Duntley, 1963).

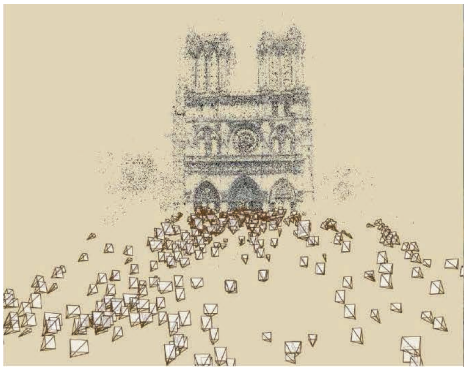
### 1.1.1 Structure from Motion

Multiple images of an object taken from varying viewpoints (due to either camera or object motion) can reveal the structure of the object, while simultaneously providing the estimated motion of the camera(s) (Figure 1.3(a)). Using images collected from a single-camera undergoing motion, structure-from-motion (SFM) employs optimization techniques to reconstruct the structure. One popular such method is bundle adjustment (Triggs et al., 2000), which solves for both known and unknown camera internal param-

**Figure 1.3** Structure from motion examples. (a) With unknown camera calibration parameters in an unordered dataset, Brown and Lowe (2005) solve for the camera motion and reconstructed structure. (b) An implementation of SFM, called photo tourism, that compiles the photos of world famous structures from the Internet to optimize the virtual structure. (c) Another large scale SFM implementation presents a landmark recognition engine using Internet images.



(a) Structure from motion using bundle adjustment (Brown and Lowe, 2005).



(b) Photo tourism. Reconstruction of Notre Dame de Paris (Snavely et al., 2006).

(c) Web-scale landmark recognition engine (Zheng et al., 2009).

eters for an unordered data set (Brown and Lowe, 2005), where the main focus is on the accuracy and consistency of the reconstructed structure and camera trajectory. One exciting example of an SFM project, called Photo tourism (Snavely et al., 2006), involves constructing a virtual structure of world famous structures, such as Notre Dame Cathedral in Paris and the Great Wall of China (Figure 1.3(b)), using photographs compiled from the Internet. Given the thousands of images available, this project solves a large sparse bundle adjustment (SBA) optimization problem, yielding a virtual 3D reconstructed structure that is globally consistent. More recent work by Zheng et al. (2009) develops a web-scale landmark recognition engine. This globally-wide project builds 5,312 landmarks from worldwide cities including sample landmarks shown in Figure 1.3(c), using efficient modeling and clustering algorithms on Internet images.

Although most SFM approaches are considered to be off-line, real-time performance

has been reported in several studies, including the work of real-time SFM by Tomasi et al. (1995), Molton et al. (2004) using small planar patches, and Mouragnon et al. (2009) using incremental bundle adjustment. Another real-time implementation, Parallel Tracking and Mapping (PTAM), has been reported by Klein and Murray (2007) with parallelized tracking and mapping processes for a small, indoor environment. More recently, Strasdat et al. (2012) presented a thorough comparison between filter-based SLAM and keyframe-based bundle adjustment, analyzing their similarities and relative advantages over a sequence of sequential image frames. Since estimated motion can be achieved as a by-product of SFM, this method has been implemented to estimate motion together with the reconstructed structure. So far, implementations have mostly been in terrestrial environments; however, recently cases of successful underwater SFM have been reported. One example of an underwater implementation is by Pizarro et al. (2009) who applied bundle adjustment to reconstruct a large-scale area and to estimate camera motion. Another example is that of Johnson-Roberson et al. (2010) who have shown impressive kilometer-scale reconstructions of benthic environments. Similarly, image mosaicing has been used to detect motion in a sequence of underwater images (Gracias et al., 2003).

### 1.1.2 Visual Odometry

Visual odometry (VO) relies on the visual information from one or more cameras to estimate odometry information (Nister et al., 2004). VO works as an odometry sensor, which can create the reconstructed structure as a by-product. Several successful implementations of VO include rough terrain (Konolige et al., 2007), Mars exploration (Maimone et al., 2007), underwater localization (Corke et al., 2007), and ground vehicles (Nister et al., 2006). Other recent developments include local optimization (Zhang and Shan, 2001), optical flow separation (Kaess et al., 2009), robust VO (Campbell et al., 2005), and combinations with other sensors (Oskiper et al., 2007) to estimate motion from an image sequence. Unlike SFM, VO must provide real-time performance when used in combination with other navigation methods because it is mostly used as an odometry sensor.

### 1.1.3 Next-Best-View

Next-Best-View (NBV) is the problem of visual sensor planning for automated model acquisition, first introduced by Connolly (1985). This view planning algorithm in computer vision chooses the next best viewpoint given a set of collected images. The objective is to reconstruct a 3D model efficiently and accurately through a planned sensor trajectory. NBV is closely related to the active perception problem (Bajcsy, 1988) with different approaches

**Table 1.1** Feature detectors and descriptors. The second column shows the functionality as a detector/descriptor, while the third column indicates S (scale), R (rotation), I (illumination) and A (affine) invariant characteristic. Note that affine invariance is used as an indicator of view-point change invariance as per Mikolajczyk et al. (2005).

| **Algorithm** | **Detector/Descriptor** | **Invariance** |
|---|---|---|
| Scale Invariant Feature Transform (SIFT) (Lowe, 2004) | Both | S,R,I |
| Speeded Up Robust Features (SURF) (Bay et al., 2006) | Both | S,R,I |
| Zernike Moments (Khotanzad and Hong, 1990) | Descriptor | S,R |
| Kanade-Lucas-Tomasi (KLT) (Shi and Tomasi, 1994) | Detector | A |
| Maximally Stable Extremal-region Detector (MSER) (Matas et al., 2004) | Detector | A |
| Harris (Harris and Stephens, 1988) | Detector | R |
| Harris-Affine (Mikolajczyk and Schmid, 2004) | Detector | A |
| Hessian-Affine (Mikolajczyk and Schmid, 2004) | Detector | A |
| Harris-Laplace (Mikolajczyk and Schmid, 2004) | Detector | A |
| Hessian-Laplace (Mikolajczyk and Schmid, 2004) | Detector | A |
| Edge Based Region detector (EBR) (Tuytelaars and Van Gool, 2004) | Detector | A |
| Intensity Extrema-Based Region detector (IBR) (Tuytelaars and Van Gool, 2004) | Detector | A |
| Features from Accelerated Segment Test (FAST) (Rosten et al., 2010) | Detector | A |
| Fern (Ozuysal et al., 2010) | Detector | I,A |
| Oriented FAST and Rotated BRIEF (ORB) (Rublee et al., 2011) | Both | R,I,A |

presented to plan for the optimal viewpoint sequence. Pito (1999) suggested a planning strategy that minimizes the unseen volume. Reed and Allen (2000) build a target model while building the 3D construction to constrain the problem. Whereas these two methods aim to reduce the unseen portion of the target model, an information-based approach is presented in the work by Li and Liu (2005). They compute the information gain for the next measurement based on entropy to determine the optimal next-view.

### 1.1.4 Place Recognition

Place recognition is primarily concerned with describing images and categorizing them, so that new images can be sorted into either existing categories (e.g., places or views) or new categories. This ability to determine if a scene has been previously visited is an important factor in robot navigation because navigation uncertainty drops significantly when a robot can recognize that it has visited the same place before (i.e., loop-closing). To recognize a previously visited place, images are represented using descriptors and sorted into different categories. One way to represent images is using a feature detector and descriptor. Various feature descriptors are presented in Table 1.1, which shows that each possesses dif-

ferent invariance properties[1]. Among these descriptors, SIFT and SURF are known to be robust to many different properties and are, therefore, popular, but are computationally expensive. Although recent results from FAST (Rosten et al., 2010), Fern detector (Ozuysal et al., 2010) and ORB (Rublee et al., 2011) prove their speed and performance with terrestrial images, this thesis chooses SIFT and SURF detector/descriptors for their robustness. These Difference of Gaussian (DoG)-based feature detector/descriptors are remarkably robust to noise and, thus, proper for underwater images that encompass backscattering. Another approach to describing images is to use a bag-of-words representation. Originally developed for text-based applications, expansion of this approach to images were found in Leung and Malik (2001), Sivic and Zisserman (2003), and Csurka et al. (2004). Using a bag-of-words representation discards geometric relationships in the image, but enables a faster search. Other popular ways to represent images include reducing the dimensionality of measurement vectors by using principal component analysis (PCA) (Chen and Wang, 2006; Dudek and Jugessur, 2000).

Once the set of images are represented by descriptors, a new input image needs to be sorted into an existing group or registered as a new group via a learning process. In this step, different machine learning techniques are used to classify (when we know the number of classes) or to cluster (when the number of classes are unknown). Example implementations of these techniques include Support vector machines (SVMs) (Luo et al., 2007; Pronobis et al., 2010), k-nearest neighbors (k-NNs), Hidden Markov models (HMMs) (Li and Kosecka, 2006) and various combinations of these techniques (Escolano et al., 2007).

The sorting process involves global / local aspects of place recognition: the process globally finds the correct match in the data set based on local descriptors. The saliency of the local descriptor implies the potential for the descriptor to be found in global matching. Thus, defining a metric for the saliency of this local aspect, and using it in a perception-driven control scheme, is of interest to this thesis.

### 1.1.5 Challenges in Underwater Vision

Because vision is the main focus of this research, its limitations and challenges in underwater perception need to be examined. Underwater imaging possesses more challenges than terrestrial imaging because of power limitations and the exponential attenuation of light (Singh et al., 2004; Jaffe et al., 2001; Duntley, 1963). Most of all, AUV missions are limited in terms of power since the vehicle runs on batteries for a mission (Bradley et al., 2001). Due to this limitation, the light source is typically strobed synchronously with the

[1]Mikolajczyk et al. (2005) shows that image distortion caused by view-point change can be modeled by an affine transformation.

camera frame rate rather than continuously illuminated as in video streaming.

Challenges come from the absorption property of water, which is wavelength-dependent and causes red to be more severely attenuated than green (Figure 1.4(a)–(b)) (Duntley, 1963). Light conditions affect scene appearance; thus, image feature descriptors should be invariant and robust to handle this effect. Secondly, the environment itself may also not have uniform spatial distribution of texture-rich scenes, as is more typical in terrestrial environments. Sample images showing this varying feature distribution underwater are depicted in Figure 1.2. These images illustrate that underwater feature distribution may play a critical factor in producing meaningful loop-closure events. Moreover, even with a feature-rich distribution, scene visibility highly depends on the transparency of water.

Images may need enhancement techniques such as dehazing (Shwartz et al., 2006; Fattal, 2008; Carlevaris-Bianco et al., 2010) or histogram equalization (Zuiderveld, 1994), before using typical feature detectors, which were introduced in §1.1.4. Figure 1.4(c) and Figure 1.4(d) show the work of dehazing by Carlevaris-Bianco et al. (2010), where the dehazed image reveals greater image detail than the original. Similarly, another enhancement technique, contrast limited adaptive histogram equalization by Zuiderveld (1994), has been applied to Figure 1.4(e) and Figure 1.4(f) to improve the image visibility. Lastly, underwater images are likely to be noisy from the effect of backscattering, and this necessitates the use of a robust motion estimation algorithm to reject false positives properly. The sample images in Figure 1.4(g) and Figure 1.4(h) are obtained from the same place and have large overlap but the image contents are highly corrupted by the noise. All of these limitations need to be considered in underwater visual SLAM.

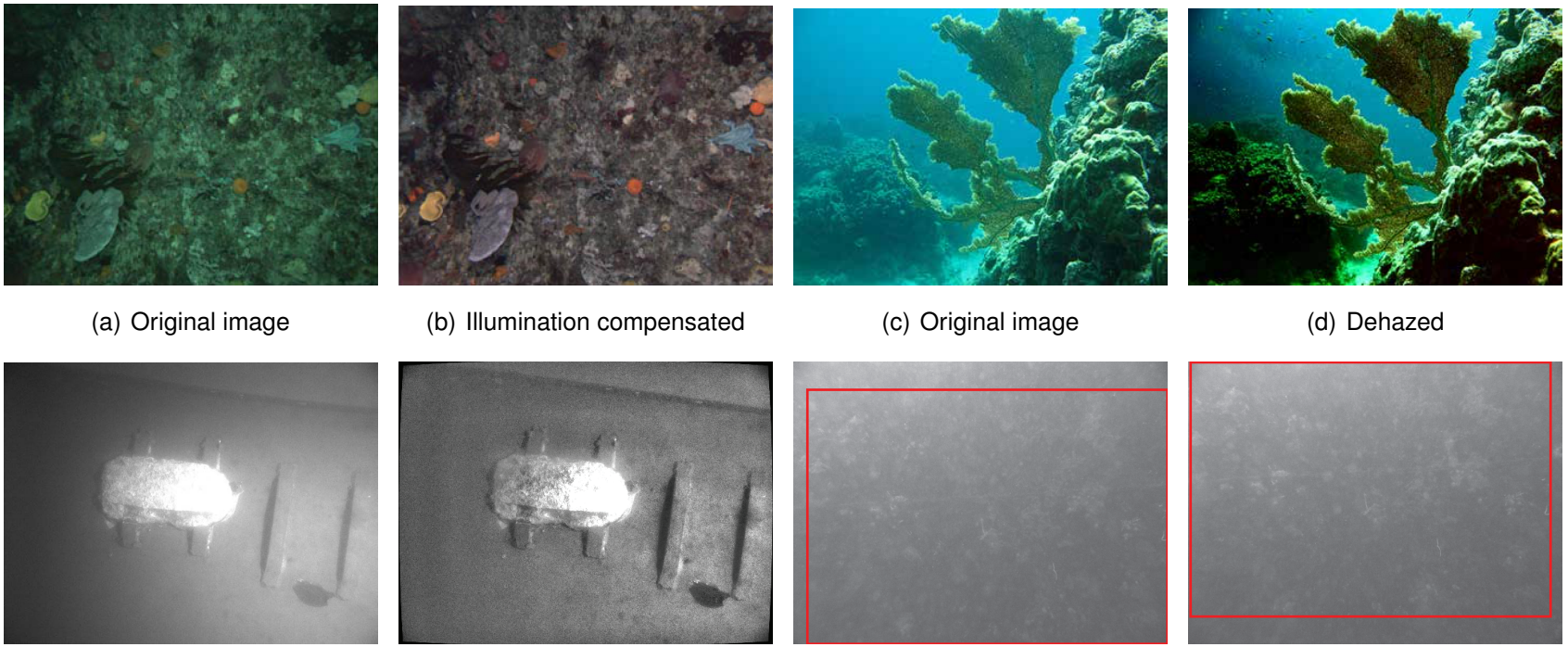
## 1.2 Literature Review on Underwater Navigation

To overcome limitations of electromagnetic signal transmittance in water, traditional underwater navigation methods have used acoustic devices and systems because acoustic energy propagates further with lower attenuation (Urick, 1983). Two such popular navigation systems are DVL and LBL systems (Whitcomb et al., 1999), which will be reviewed in this section.

### 1.2.1 Beacon-based Positioning Systems

Acoustic beacon positioning systems fall into one of three types: long-baseline (LBL), short-baseline (SBL), and ultra short baseline (USBL) (Figure 1.5). LBL and SBL measure range by calculating the time of flight (TOF), whereas USBL measures both range and bearing from the TOF and phase. Among these systems, however, LBL is known to be the

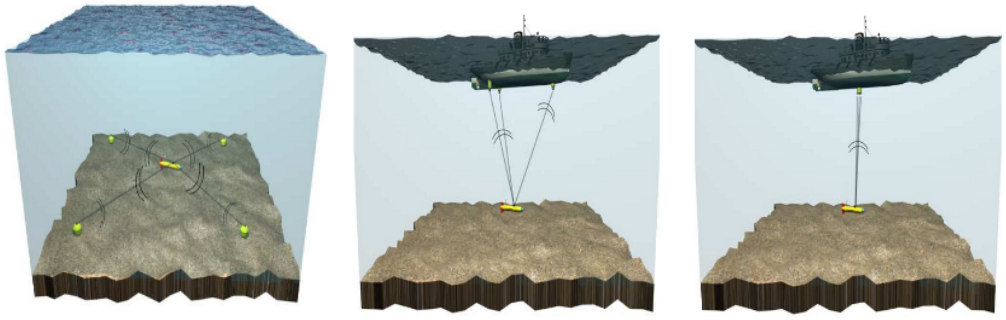
**Figure 1.4** Challenges in underwater images. The top two figures ((a) and (b)) show the effect of illumination compensation for underwater images (Johnson-Roberson et al., 2010). The images (c) and (d) are the dehazing work by Carlevaris-Bianco et al. (2010). The images in the last row are the result of applying contrast limited adaptive histogram equalization (Zuiderveld, 1994) to underwater ship hull images. Images (g) and (h) are sample underwater images with noise, and are taken from the same place where the red box indicates their overlap.



(a) Original image (b) Illumination compensated (c) Original image (d) Dehazed

(e) Original image (f) Histogram equalized (g) Backscatter noise (h) Backscatter noise

**Figure 1.5** Typical underwater beacon acoustic positioning systems (Alcocer et al., 2006). Three types of underwater acoustic positioning systems (LBL, SBL, and USBL) are illustrated.



(a) LBL (b) SBL (c) USBL

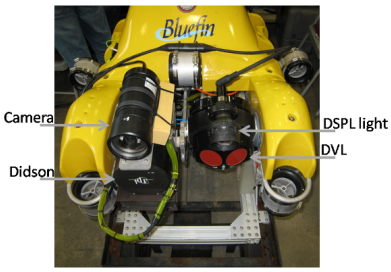
most accurate positioning system (Austin et al., 1984) for deep ocean operations using a seafloor installed beacon network. This navigation method requires a network of acoustic transponders to first be deployed and calibrated in the area where the survey is planned. Using each beacon's range measurement, a robot triangulates its position in the underwater space by measuring the time of travel between the beacons and the robot. The main advantage of the system is that errors are globally bounded, regardless of the run time and the scale of the survey. However, the system requires the additional procedure of beacon installation and calibration ahead of time.

### 1.2.2 Doppler velocity log (DVL)

The advent of the DVL is relatively recent in underwater navigation (Brumley et al., 1987; Brokloff, 1994). This system uses separate acoustic beams to measure velocity using the Doppler effect. A typical DVL configuration uses four beams in a Janus configuration to measure the Doppler shift and calculate motion with respect to the seafloor (Figure 1.6). By transforming the beam measurements, a three-dimensional body-frame velocity with respect to the seafloor can be obtained with high accuracy of a few cm/s (Table 1.2) (Teledyne RD Instruments, 2008). Navigation using a DVL requires integrating velocity over time to compute the position of the vehicle; thus, even with highly accurate velocity sensor measurements, integrated pose uncertainty will monotonically increase and exhibit unbounded error growth for a long mission. Therefore, in practice, errors should be bounded using a combination of sensors.

**Figure 1.6** Photo of DVL mounted on two underwater vehicles. (a) University of Michigan AUV testbed. (b) Hovering Autonomous Underwater Vehicle (HAUV) used for in-water ship hull inspection. (c) RDI ADCP

(a) DVL on UMich AUV




(b) DVL on HAUV

(c) RDI DVL

**Table 1.2** 600 kHz DVL single-ping precision of bottom-track velocity

| | |
|---|---|
| Maximum altitude | 65 m |
| Minimum altitude | 0.5 m |
| Velocity range | $\pm$ 5 m/ s |
| Long term accuracy | $\pm$ 0.4 % $\pm$ 0.2 cm/ s |
| Precision at 1 m/s | $\pm$ 1.2 cm/s |
| Resolution | 0.1 cm/s (default), 0.01mm/s (selectable) |
| Ping rate | 4–7 Hz typical |

## 1.3 Literature Review on SLAM

A robot has to solve two essential problems in navigation, namely localization (knowing where it is) and mapping (building a map of its environment). One novel solution to these two interwoven problems is SLAM, which estimates locations and maps simultaneously based on motion models and sensor measurements. Stemming from the early seminal work of Smith et al. (1988), which addressed how uncertainty propagates through stochastic spatial constraints, SLAM has been developed in various areas and will be described below. While SLAM can be categorized in different ways, one method is to create branches of the SLAM framework on the basis of the various sensor combinations that it uses. For instance, the recent rise in the popularity of cameras or laser sensors represents a typical sensor selection in SLAM. Such implementations emphasize two main issues: *(i)* the ability to handle large scale maps (scalability) and *(ii)* reliable data association. This section reviews SLAM algorithms relevant for pose-graph visual SLAM for underwater environments, focusing on the advances in these two main areas.

### 1.3.1 Large Scale SLAM

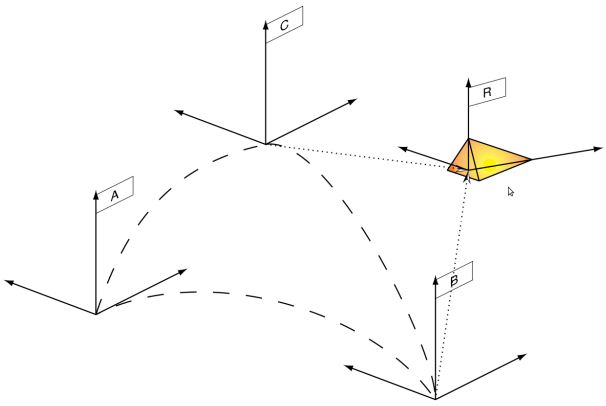
The main challenge for managing large scale maps lies in the computational cost. Traditionally, the extended Kalman filter (EKF) in SLAM has been widely used as a standard nonlinear state estimator for its simplicity. However, this approach is limited for large scale maps given that its computational cost is $O(n^2)$ for a state vector with size $n$, where $n$ is the number of landmarks contained in the map. Thus, the cost rises as the dimension of the state vector increases whenever more landmarks are added to the map. To address the quadratically increasing computational cost of the EKF, several approaches have been developed, such as *(i)* submaps and *(ii)* information filters.

The first method for achieving a large scale map involves the use of submapping techniques, i.e., building local maps and then stitching the local maps together to build larger maps. This type of submap technique was originally proposed by Leonard and Feder (2001) to reduce the computational cost by decoupling the larger maps into submaps. This work was further developed into the *Atlas* framework by Bosse et al. (2004) where each submap is represented as a coordinate system in a global graph (Figure 1.7(a)). For examples of underwater implementations, submap approaches using vision (Pizarro et al., 2009) and bathymetry maps from acoustic sensors (Roman and Singh, 2005) have been successfully demonstrated. All of these mentioned approaches, however, require the additional work of local map management.

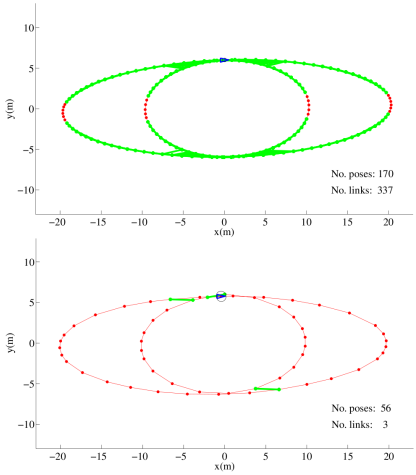
Another approach for mapping large maps is to use a sparse extended information filter (SEIF) (Thrun et al., 2004), which stems from an extended information filter (EIF) (Mutambara, 1999)—a dual form of the EKF. Thrun et al. (2004) showed that the information matrix is approximately sparse for landmark-based SLAM, and that it can be made exactly sparse under certain assumptions, allowing for its scalability to large areas. Due to the approximated sparsity (Figure. 1.7(c)) of the SLAM information matrix, a SEIF can avoid the quadratic cost associated with the EKF. This insight was further developed into an exactly sparse delayed-state filter (ESDF) by Eustice et al. (2006b), who proved that the information matrix is exactly sparse without approximation by keeping all delayed poses. This information filter approach has been further improved to select only informative measurements (Ila et al., 2010), which significantly reduces computational cost by maintaining only informative links ( Figure 1.7(b)).

Finally, another method is called optimization-based SLAM. Unlike the filter-based approach, building a map and localizing the vehicle position can be considered to be an optimization problem. In a filter-based SLAM framework, both the extended Kalman filter (EKF) and the extended information filter (EIF) are based on single-point linearization. Because a linear filter has limitations due to linearization errors, an alternative solution

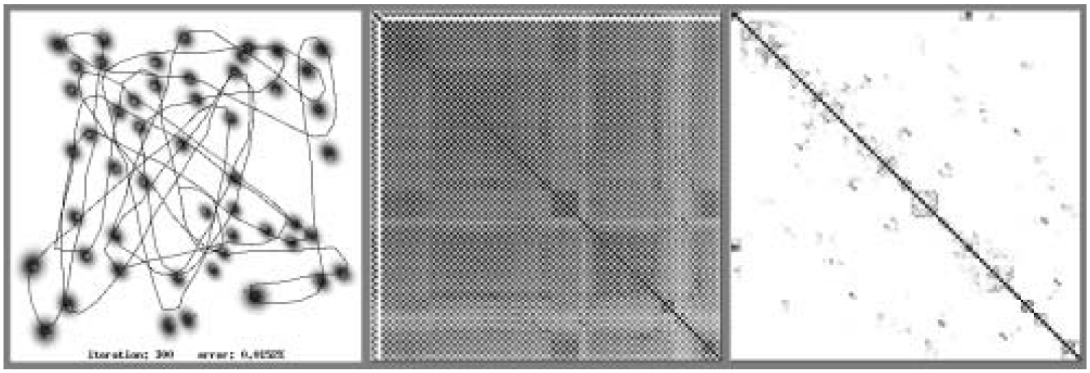
**Figure 1.7** Large scale SLAM. Submap and information form approaches for large-scale SLAM.




(a) Atlas framework (Bosse et al., 2004)




(b) Compact SLAM (Ila et al., 2010)



(c) Information matrix in sparse extended information filter (SEIF) (Thrun et al., 2004)

to navigation has been proposed by optimizing all sensor constraints to find a global minimum. This use of incremental optimization to solve the SLAM problem has been the focus of several studies. For instance, an on-line nonlinear optimization, a variant of Stochastic Gradient Descent, has been used to solve the SLAM problem (Olson et al., 2007; Grisetti et al., 2008). Considering the SLAM problem as a graph, Konolige (2005) addressed the issue using an optimization technique that involved variable reduction. A similar approach developed by Kaess (2008) used smoothing rather than filtering, which yielded a successful implementation of a method called iSAM (incremental smoothing and mapping). The iSAM approach uses nonlinear optimization (Gauss Newton or the Levenberg-Marquardt) and solves SLAM as a least-squares problem. Although these optimization techniques linearize the objective function to find the solution, the iteration step prevents the algorithm from the incorrect linearization. Another approach, Frame SLAM (Konolige and Agrawal,

2008), uses submaps in the optimization, which estimates motion in small maps and optimizes from a global aspect.

### 1.3.2 Reliable Data Association

Data association is the problem of establishing correspondence between measurements and map elements. Given a new measurement, data association involves solving the problem of how a robot can determine the correct match to an existing part of the map when they are separated in both time and space. Data association is known to be one of the hardest, but most important problems in SLAM. When either measurement or estimation is inaccurate (e.g., an incorrect correspondence has been established in data association), the map and the position estimate will be distorted by the bad match.

Efforts to solve loop-closing problems have been reported in (Bosse and Zlot, 2008; Neira and Tardos, 2001; Kuipers and Beeson, 2002). For example, Bosse and Zlot (2008) developed a two-level loop-closing algorithm based on the *Atlas* framework, while Neira and Tardos (2001) introduced a way to measure joint compatibility, which resulted in optimal data association performance. Kuipers and Beeson (2002) use a machine learning technique, bootstrap learning, to solve the place recognition problem together with a topological map, called the Spatial Semantic Hierarchy (SSH). Recently, vision information has been exploited in data association by providing an independent loop-closing ability based on appearance, which will be presented in the following section.

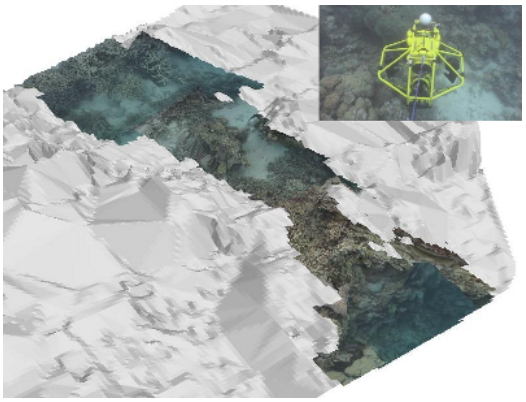
#### Appearance-based SLAM

Applying place recognition to robot navigation is a relatively recent approach, yielding robust and successful results. Ho and Newman (2007) used a similarity matrix based on scene similarity, which showed a way to use appearance in loop-closing. A related development in appearance-only SLAM by Cummins and Newman (2009), called Fast Appearance-Based Mapping (FAB-MAP), shows an impressive result by mapping a large-scale area in real-time. This similarity-based approach solely depends on the visual information, i.e., the appearance of the map using a bag-of-words representation. However, for an underwater environment, the possibility of successful visual place recognition may not be as uniformly high as that for terrestrial images and, therefore, should take into account geometric uncertainty when hypothesizing candidate places. Another point to note about FAB-MAP is that it builds its vocabulary offline using a training dataset, which could be a problem when the trained vocabulary is not representative of the current application (e.g., training terrestrial images for an underwater application). There are appearance-based

SLAM implementations that maintain the vocabulary online, such as those by Angeli et al. (2008) and Kawewong et al. (2010). In these implementations, they initiate the vocabulary from an empty set and, then, build as the mission proceeds. For example, Kawewong et al. (2010) presented the Position Invariant Robust Feature (PIRF) based navigation algorithm using online vocabulary, and showed comparable performance to FAB-MAP.

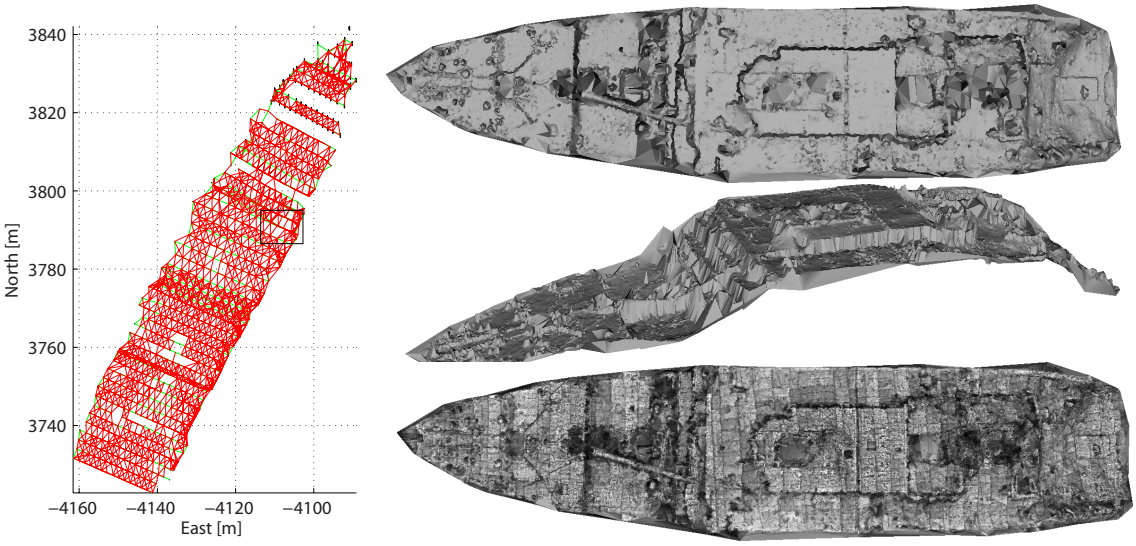
### 1.3.3 Underwater Visual SLAM

**Figure 1.8** Examples of underwater visual SLAM.



(a) Mapping of the Great Barrier reef (Williams and Mahon, 2004)

(b) Underwater cave exploration (Fairfield et al., 2008)




(c) Mapping of the RMS Titanic (Eustice et al., 2006b)

Because this thesis focuses on underwater robotics applications, this section briefly reviews the SLAM framework in an underwater environment, especially when a system uses vision. For underwater environments, where GPS is unavailable, SLAM should be able

to cope with monotonically growing navigation drift. However, underwater visual SLAM is more challenging than terrestrial navigation in several respects. First, light conditions are limited. Because ambient sunlight is not a reliable light source in this environment, robots must carry their own light source. The light conditions also limit the motion of the robot, requiring it to be on the order of several meters away from the scene being imaged. Another challenge in underwater navigation is that the scene's feature distribution is highly amorphous in underwater areas. When performing SLAM on land, we expect man-made environments with relatively feature-rich distributions. However, in an underwater environment, such uniform feature distribution is often absent; thus SLAM implementations in this domain require more robust handling of camera derived constraints. Despite these challenges, several successful implementations can be found in the underwater visual SLAM literature (Figure 1.8): e.g., in research on entropy SLAM (Saez et al., 2006), mapping of the Great Barrier reef (Williams and Mahon, 2004), and mapping the Titanic using visually augmented navigation (VAN) (Eustice et al., 2006b). More recently, Medagoda et al. (2011) propose a method that improves mid-water column navigation also using visual SLAM.

## 1.4 Literature Review on Path Planning and Visual Servoing

Perception driven navigation is very similar to path planning and visual servoing in many ways. In this section, these relevant areas are reviewed in the context of their relation to perception driven navigation.

### 1.4.1 Path Planning

Path planning can be performed given knowledge of the environment and state information of the robot. Path planning algorithms aim to achieve an optimal trajectory for the robot within a set of specified constraints, such as obstacles avoidance and the need for energy efficiency. In this section, both deterministic and probabilistic path planning for general platforms will be reviewed, followed by a discussion of path planning for the underwater environment.

Traditional path planning solves for the optimal path when the state and control sequences are deterministic (Yang and Brock, 2006; Carroll et al., 1992). Stochastic search in path planning has been considered from the perspective of uncertainty in a control sequence (LaValle and Kuffner, 1999; Kavraki et al., 1996; Kaelbling et al., 1995). This has been further developed in the work of Prentice and Roy (2009), which considers the uncertainty of the states in the information domain.

Typical path planning algorithms used in the underwater domain are A* (Carroll et al., 1992), potential fields (Warren, 1990), and partially observable Markov decision processes (POMDP) (Kurniawati et al., 2008; Saigol et al., 2009). However, unlike terrestrial vehicle path planning, underwater path planning includes different aspects of the environment, including water currents (Kruger et al., 2007) and ocean convection (Willcox et al., 1996). Another area that may be related to path planning is survey design, where the design is often conducted prior to the mission and, thus, can be considered as off-line path planning.

### 1.4.2 Visual Servoing

Visual servoing aims to find the proper control action based on vision sensor information (Chaumette and Hutchinson, 2006). This vision information comes from either single or multiple cameras mounted to the robot. From this vision information, visual servoing solves for the current configuration of the robot and uses it to generate a control scheme. Applications of visual servoing include manipulator end effector control using cameras (Kragic and Christensen, 2002), and tracking an object in a ground/underwater environment (Rife and Rock, 2001). Visual servoing or vision-based robot control is very similar to perception driven navigation in that it generates a control sequence based on vision input. However, the underlying objective function is different: perception driven navigation aims to achieve better navigation performance by solving the control-navigation problem, while visual servoing seeks the instant feedback control without evaluating the overall navigation results.

## 1.5 Thesis Outline

This thesis develops active visual SLAM for large-scale autonomous underwater navigation. Chapter II presents conventional visual SLAM with preplanned trajectory. Chapter III introduces two novel measures of visual saliency, which capture image texture-richness and rarity. These saliency metrics are then used in SLAM in an active way to improve SLAM performance. Lastly, Chapter IV introduces PDN—algorithm designed to solve the integrated SLAM navigation and area coverage problem.

# CHAPTER II

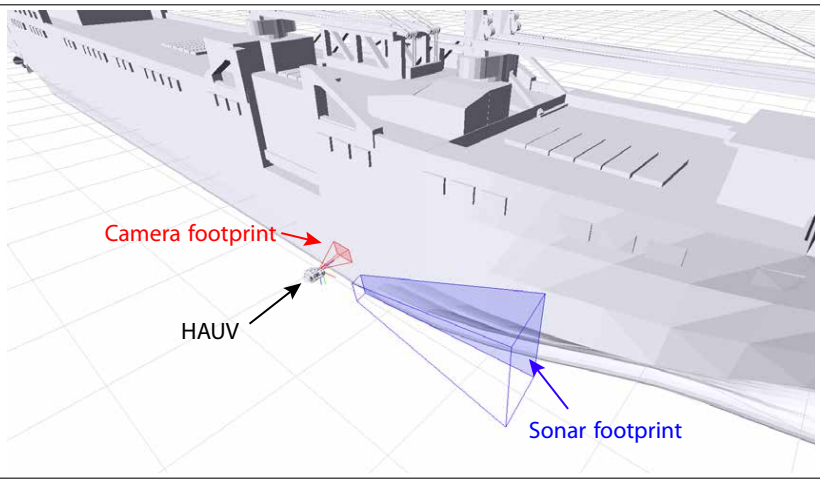
# Real-time Pose-graph Visual SLAM

This chapter presents an extension of visually augmented navigation (VAN) (Eustice et al., 2006b) to include geometric model selection for robust image registration, mainly focusing on pose-graph visual SLAM. Originally, the VAN algorithm used only 3D (essential matrix) registration models to estimate motion from camera measurements. However, when the scene is planar, a 2D (homography) model can solve the motion estimation problem in a more robust and simpler way; thus, we introduce a model selection framework to determine the appropriate model. This extension was motivated from a ship hull inspection application, where the nature of the structure (the hull) is mostly locally planar, but also includes highly three dimensional structures (e.g, bilge keel, shaft, screw). This structural characteristic motivated us to develop a robust image registration engine capable of dealing with the structure variance. In this chapter, we start from the essential background of our pose-graph simultaneous localization and mapping (SLAM) algorithm and then introduce the camera engine, including its model selection framework. Real-world results from ship hull inspection are presented.

## 2.1 In-water Hull Inspection using an AUV

This section briefly introduces an overview of the in-water ship hull inspection application before describing the SLAM implementation. Many underwater structures such as dams, ship hulls, harbors and pipelines need to be periodically inspected for assessment, maintenance and security reasons. Among these, our interest is in autonomous underwater hull inspection, which seeks to map and inspect the below-water portion of a ship *in situ* while in port or at sea. Typical methods for port security and ship hull inspection require either deploying human divers (Mittleman and Wyman, 1980; Mittleman and Swan, 1993), using trained marine mammals (Olds, 2003), or piloting a ROV (Lynn and Bohlander,

**Figure 2.1** Illustration of the in-water hull inspection application using an AUV. Depicted is the Hovering Autonomous Underwater Vehicle (HAUV) performing inspection on a hull using camera and sonar sensor modalities. The individual sensor footprints are plotted with red (camera) and blue (sonar) cones.




1999; Carvalho et al., 2003; Negahdaripour and Firoozfam, 2006). Autonomous vehicles have the potential for better coverage efficiency, improved survey precision and overall reduced need for human intervention. As early as 1992, there was an identified need within the Naval community for developing such systems (Bohlander et al., 1992). In recent times, effort in this area has resulted in the development of a number of automated hull inspection platforms (Harris and Slate, 1999; Trimble and Belcher, 2002; Vaganay et al., 2006; Menegaldo et al., 2008).

### 2.1.1 Challenges and Approaches

In the absence of global positioning system (GPS), underwater navigation feedback in this context is typically performed using an inertial measurement unit (IMU) or DVL derived odometry (Menegaldo et al., 2009; Vaganay et al., 2006), and/or acoustic beacon time-of-flight ranging (Trimble and Belcher, 2002; Desert Star Systems, 2002). The main difficulties of these traditional navigation approaches are that they either suffer from unbounded drift (e.g., odometry), or they require external infrastructure that needs to be set up and calibrated (e.g., acoustic beacons). Both of these scenarios tend to vitiate the "turn-key" automation capability that is desirable in autonomous hull inspection.

The basic idea behind our autonomous hull inspection navigation approach is to use camera-drived constraints to correct the dead-reckoned (DR) navigation drift. In this project, we have used two imaging sensor modalities, a sonar and a camera, on an autonomous underwater vehicle (AUV) for the inspection and navigation task. As presented in works by

Walter et al. (2008) and Johannsson et al. (2010), an imaging sonar provides a promising imaging quality regardless of the water clarity. An optical camera, on the other hand, can provide far more detail than sonar when water clarity allows. As can be seen in Figure 2.1, the sonar has a larger sensor footprint coverage than the camera. Typical inspection missions aim to achieve 100% sensor coverage with sonar which yields mission profiles with too large of track-line spacing for the camera to have cross-track image overlap. In this type of mission profile, however, the camera provides useful navigation constraints wherever the AUV traverses back over its path. In this thesis, we focus only on the camera modality aspect of SLAM constraints. For readers interested in details of the hull inspection project, more information can be found in Appendix Chapter A.

## 2.2 System Overview

For this SLAM problem, we employ a pose-graph SLAM representation of the environment and, therefore, augment our state description to include a collection of historical vehicle poses sampled at regular spatial intervals throughout the environment. For the coordinate frame definition, see Appendix §A.3.

### State Representation

We estimate the vehicle's full degree of freedom (DOF) pose, $\mathbf{x} = [x, y, z, \phi, \theta, \psi]^\top$, where the pose (position and Euler attitude) is defined in a local-level Cartesian frame. The augmented state representation is expressed as follows for $n$ keyframes

$$X = \left[\mathbf{x}_1^\top, \cdots, \mathbf{x}_i^\top, \cdots, \mathbf{x}_n^\top\right]^\top,$$

with each of the pose samples, $\mathbf{x}_i$, corresponding to the time instance $t_i$ of a keyframe stored by our visual perception process.

For the process and measurement models, we assume standard Gaussian models with independent control $\mathbf{u}_i$ and measurement noise, $\mathbf{v}_i \sim \mathcal{N}(0, \Sigma_i)$ and $\mathbf{w}_k \sim \mathcal{N}(0, \Lambda_k)$, respectively, where

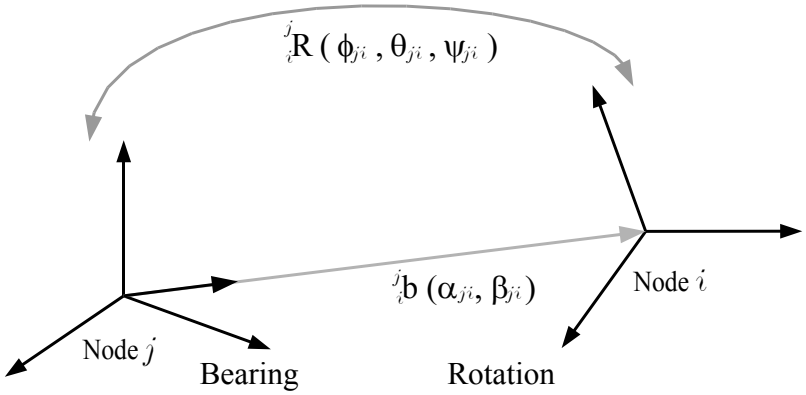
$$\mathbf{x}_i = f(\mathbf{x}_{i-1}, \mathbf{u}_i) + \mathbf{v}_i, \tag{2.1}$$

is a stochastic state transition model linking two sequential poses and

$$\mathbf{z}_{ij}^k = h(\mathbf{x}_i, \mathbf{x}_j) + \mathbf{w}_k, \tag{2.2}$$

is a stochastic measurement model between two nodes $i$ and $j$ with measurement index $k$.

**Figure 2.2** 5-DOF camera measurement model. The camera measurement is a bearing-only relative-pose between node $i$ and node $j$ that consists of azimuth, elevation and Euler angles.




| Bearing | | Rotation | |
|---|---|---|---|
| $\alpha_{ji}$ | azimuth | $\phi_{ji}$ | Euler roll |
| $\beta_{ji}$ | elevation | $\theta_{ji}$ | Euler pitch |
| | | $\psi_{ji}$ | Euler yaw |

### Camera Constraints

Throughout this thesis, keyframe registration is the main form of SLAM measurement adding non-sequential constraints to the graph. The camera measurement is defined as a pairwise registration of keyframes acquired by the calibrated monocular vision system, which results in relative-pose constraints modulo scale between historical pose samples in $X$. Here, the 5-DOF camera measurement $\mathbf{z}^{k}_{ij}$ between poses $\mathbf{x}_i$ and $\mathbf{x}_j$ is modeled as an observation of the azimuth $\alpha_{ji}$ and elevation angle $\beta_{ji}$ of the baseline direction of motion, and the relative Euler orientation $\phi_{ji}, \theta_{ji}, \psi_{ji}$ between the two camera poses (Eustice et al., 2008, 2006b). The detailed computation is provided in Appendix §A.4.

$$\mathbf{z}_{ij} = h_{\mathsf{5dof}}(\mathbf{x}_i, \mathbf{x}_j) = \Big[\alpha_{ji}, \beta_{ji}, \phi_{ji}, \theta_{ji}, \psi_{ji}\Big]^{\top}, \tag{2.3}$$

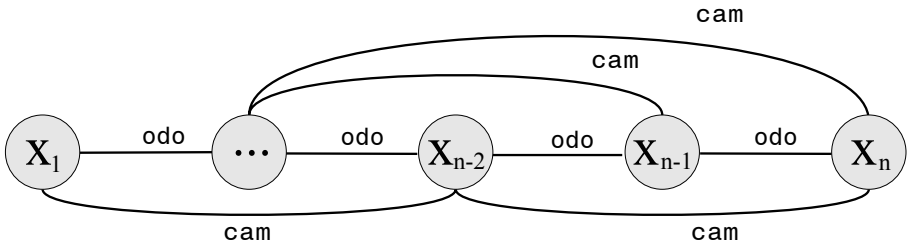
,where the Jacobian of $\mathbf{z}_{ij}$ with respect to $X$ is sparse,

$$\mathrm{H}_{\mathbf{x}} = \Big[0 \quad \cdots \quad \tfrac{\partial \mathbf{h}_{\mathsf{5dof}}}{\partial \mathbf{x}_i} \quad \cdots \quad 0 \quad \tfrac{\partial \mathbf{h}_{\mathsf{5dof}}}{\partial \mathbf{x}_j} \quad \cdots \quad 0\Big]. \tag{2.4}$$

### SLAM Back-end

Many inference algorithms exist to solve the pose-graph SLAM problem (Lu and Milios, 1997; Konolige, 2004; Eustice et al., 2006a; Olson et al., 2007; Konolige and Agrawal, 2008; Grisetti et al., 2008; Kaess et al., 2012, 2008). In this thesis, incremental smoothing and mapping (iSAM) by Kaess et al. (2012) is used as the SLAM back end. In this framework, a collection of poses are estimated given odometry constraints (`odo`) and camera

**Figure 2.3** Depiction of the pose-graph SLAM constraint graph. Odometry constraints (odo) are sequential whereas camera constraints (cam) can be either sequential or non-sequential.




constraints (cam) between poses in the graph.

To use iSAM as the back-end server, we employ the open-source iSAM algorithm due to its efficiency for real-time implementation and covariance recovery (Kaess et al., 2010, 2008; Kaess and Dellaert, 2009). iSAM solves for the maximum *a posteriori* (MAP) estimate of the joint probability distribution

$$p(X, Z, U) \propto p(\mathbf{x}_0) \prod_{i=1}^{n} p(\mathbf{x}_i|\mathbf{x}_{i-1}, \mathbf{u}_i) \prod_{k=1}^{m} p(\mathbf{z}_{ij}^k|\mathbf{x}_i, \mathbf{x}_j), \tag{2.5}$$

where $X$ is the augmented state vector with poses of cardinality $n$, $Z$ is the set of measurements of cardinality $m$, and $U$ is the set of control (i.e., odometry) inputs of cardinality $n-1$.

Under the above assumptions, we can formulate the maximization of (2.5) as

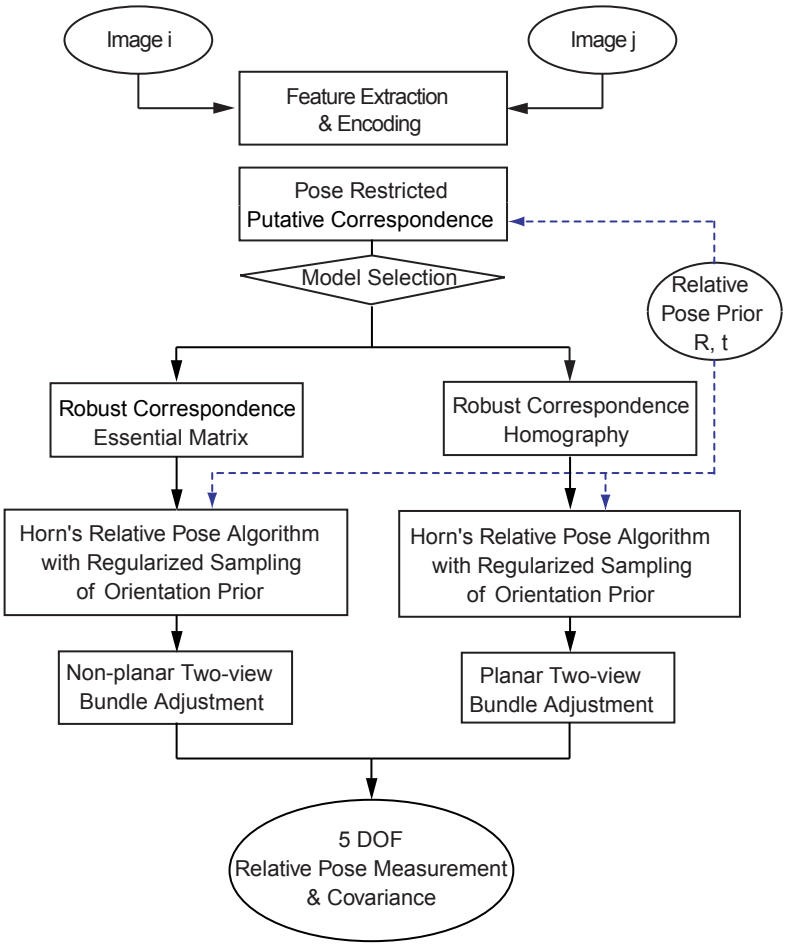
$$\hat{X} = \arg\max_X p(Z, U|X)p(X) \tag{2.6}$$

$$= \arg\min_X \sum_{i=1}^{n-1} ||f(\mathbf{x}_{i-1}, \mathbf{u}_i) - \mathbf{x}_i||^2_{\Sigma_i} + \sum_{k=1}^{m} ||h(\mathbf{x}_i, \mathbf{x}_j) - \mathbf{z}_{ij}^k||^2_{\Lambda_i} + ||\mathbf{x}_0 - \hat{\mathbf{x}}_0||^2_{\Sigma_0}, \tag{2.7}$$

where $\hat{X}$ is the MAP estimate of the pose-graph state. The iSAM algorithm uses nonlinear optimization to solve SLAM as an incremental least-squares problem. Although this optimization technique linearizes the objective function to find the solution, each batch step allows for the relinearization of the original set of nonlinear constraints, which aids in convergence. Based upon this formulation, iSAM also provides efficient marginal covariance recovery via dynamic programming (Kaess and Dellaert, 2009), which is useful during data association and link hypothesis (Section 3.3.2).

## 2.3 Two-view Camera Registration Engine

**Figure 2.4** Block-diagram of the core image registration engine. Given a proposed candidate image pair, features are extracted (§2.3.1) and putative correspondences are found (§2.3.2). A robust geometric model selection framework (§2.3.3) then determines whether an Essential matrix or homography registration model best applies. The result is checked against the SLAM pose prior using a Mahalanobis distance gate for outlier rejection. Finally, the resulting inlier correspondences are fed to a two-view bundle adjustment (§2.3.4) to determine the optimal 5-DOF relative-pose constraint.




This section discusses the core of our visual SLAM perception engine, namely the camera registration engine. Given image features and a proposed overlapping image pair, the process of pairwise motion estimation involves four main steps: 1) image feature extraction; 2) establishing inlier correspondences; 3) fitting a geometric registration model using a robust estimation framework; and 4) optimizing the camera relative-pose constraint. A block-diagram of the overall image registration algorithm is depicted in Figure 2.4.

### 2.3.1 Feature Extraction

The first step in the two-view camera engine is to extract robust feature points for putative matching. Images are first radially undistorted and contrast corrected using contrast-limited adaptive histogram equalization (Zuiderveld, 1994). This enhances the visual detail and aids in feature extraction (Figure 2.5(b)). We use a combination of Scale Invariant Feature Transform (SIFT) (Lowe, 2004) and Speeded Up Robust Features (SURF) (Bay et al., 2008) feature detectors/descriptors. SURF and SIFT exhibit scale, rotation, and illumination invariance. In our experience, we have found SIFT's gradient-based histogram descriptors to be more discriminatory for putative matching on underwater hull imagery, while SURF's Haar wavelet-based descriptors are better for our bag-of-words saliency calculation (to be discussed in §III). Graphics Processing Unit (GPU) based implementations (Wu, 2007; Furgale and Tong, 2010) are available to speed up the feature extraction to $\sim$10 Hz SIFT and $\sim$30 Hz SURF on 1 Mpx imagery (i.e., $1360 \times 1024$).

### 2.3.2 Pose-Constrained Correspondence Search

Once the features have been extracted, we use our SLAM pose prior knowledge to guide the pairwise putative matching. This matching is formulated in terms of a probabilistically-driven pose-constrained correspondence search (PCCS) (Eustice et al., 2008; Carlevaris-Bianco and Eustice, 2011). PCCS allows us to spatially restrict the putative correspondence search region and, thereby, register what would otherwise be feature-sparse underwater imagery (e.g., Figure 2.5(c) top).

The PCCS search bound is formed by assuming two projective camera models,

$$\mathrm{P} = \mathrm{K}[\mathrm{I} \,|\, \mathbf{0}] \text{ and } \mathrm{P}' = \mathrm{K}[\mathrm{R} \,|\, \mathbf{t}], \tag{2.8}$$

where $\mathrm{R}$ and $\mathbf{t}$ represent the relative-pose rotation and translation between the two cameras, and $\mathrm{K}$ is the camera calibration matrix. With scene depth $Z$ coarsely measured from DVL range measurements, we can use the infinite homography ($\mathrm{H}_\infty = \mathrm{KRK}^{-1}$) to map a point, $\mathbf{u}$, in one image to its projection in the paired image, $\mathbf{u}'$, via

$$\mathbf{u}' = \frac{\mathrm{H}_\infty \mathbf{u} + \mathrm{K}\mathbf{t}/Z}{\mathbf{H}_\infty^{3\top} \mathbf{u} + t_z/Z}, \tag{2.9}$$

where $\mathbf{H}_\infty^{3\top}$ refers to the third row of $\mathrm{H}_\infty$, and $t_z$ is the third element of $\mathbf{t}$. Because the relative-pose prior and the scene depth are known with uncertainty, we can use (2.9) to obtain a first-order covariance of the projected point. This resulting covariance yields an

elliptical search region oriented along the epipolar line (Figure 2.5(c)).

Two cases are interesting to note here and are illustrated in Figure 2.5. Figure 2.5 shows a depiction of the underwater image registration process as applied to non-salient (top) and salient (bottom) hull imagery. The top set of images are for a non-salient image pair with a strong SLAM pose prior. The bottom set of images are for a visually salient image pair, but with a weak SLAM pose prior. Raw images (Figure 2.5(a)) are first radially undistorted and histogram equalized (Figure 2.5(b)) before extracting features. A pose-constrained correspondence search (Figure 2.5(c)) using the SLAM pose prior is then applied to guide putative matching. The resulting inlier correspondences (Figure 2.5(d)) and motion model are found from a robust estimation geometric model selection framework. Finally, these inliers are fed to a two-view bundle adjustment to determine the maximum likelihood estimate (MLE) 5-DOF camera pose constraint (Figure 2.5(e)).

In cases where we have a strong prior on the relative vehicle motion, for example sequential imagery with odometry or when the pose-graph estimate is well-constrained, the probabilistic search region provides a tight bound for putative matching search (e.g., Figure 2.5(c) top). Due to this tight bound, we can often match what would be otherwise non-salient imagery (e.g., Figure 2.5(d) top). On the other hand, when we have a weak pose prior, for example poor odometry or closing large loops, then the PCCS search constraint will be uninformative (e.g., Figure 2.5(c) bottom). Nevertheless, if the hull imagery is sufficiently salient, then images may be matched using purely appearance-based means (e.g., Figure 2.5(d) bottom). This indicates that image saliency plays a major role in determining successful camera measurements and could be exploited if quantified. This observation forms the basis of our image saliency framework discussed in Chapter III.

### 2.3.3 Geometric Model Selection

Once we have established putative correspondences, we can then refine these to determine an inlier correspondence set via a random sample consensus (RANSAC) model-selection framework (Kim and Eustice, 2009). One may expect that a locally planar structure assumption is adequate on the open areas of the hull; however, this assumption is not everywhere true because some portions of the hull are highly three-dimensional (e.g., bilge keel, shaft, screw). To accommodate for this structure variability, we employ a geometric model-selection framework to automatically choose the proper registration model, either homography or essential matrix, when robustly determining the inlier set.

There are several reported algorithms in the literature aimed at overcoming registration ambiguity by using a model-selection framework to automatically determine the appropriate registration model. Akaike Information Criterion (AIC) (Akaike, 1974), Geometric

**Figure 2.5** Two factors in successful registration: prior uncertainty and saliency. The top set of images are for a non-salient image pair with a strong SLAM pose prior. The bottom set of images are for a visually salient image pair, but with a weak SLAM pose prior. A pose-constrained correspondence search (PCCS) (c) using the relative-pose prior is applied to guide putative matching. Note that this search constraint is very strong in the top set of images, but very weak for the bottom set.

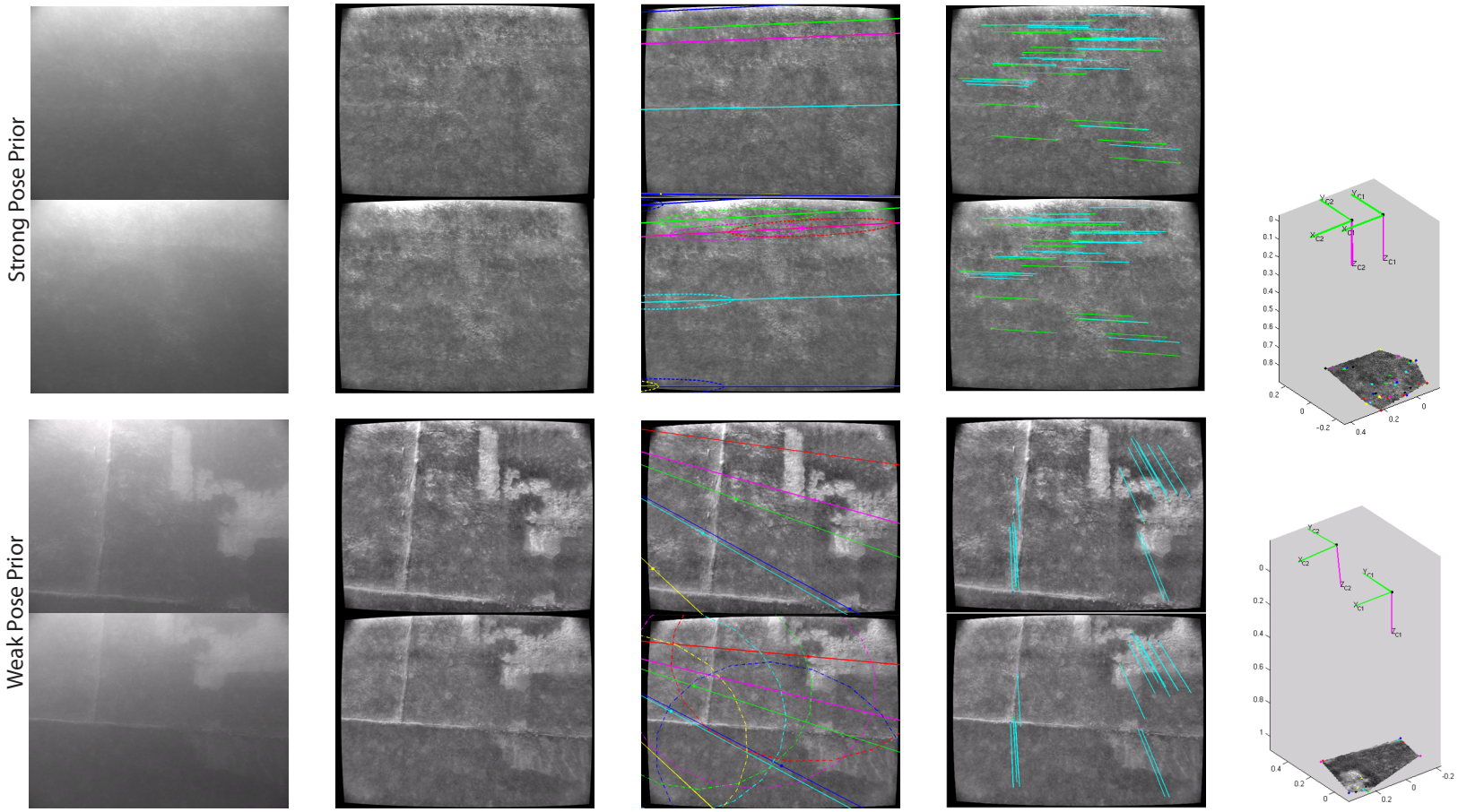
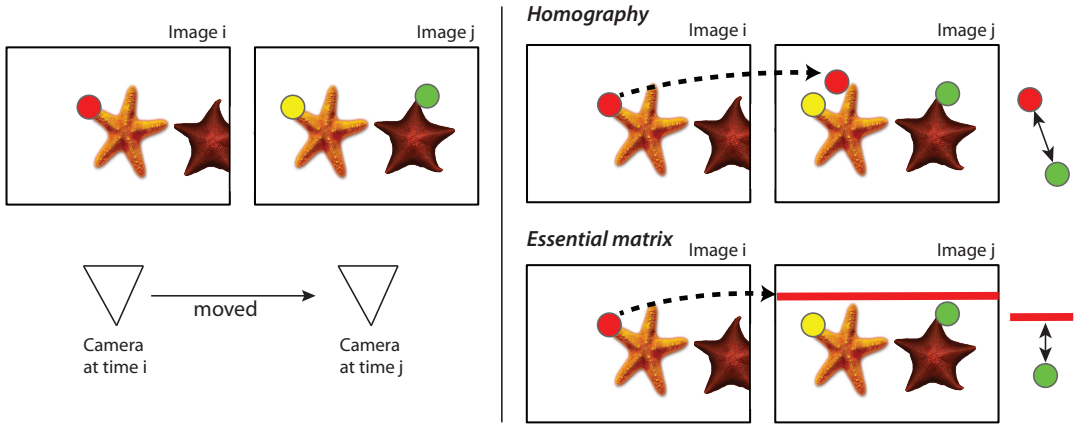


**Figure 2.6** Accuracy in inliers selection for homography versus essential matrix. Illustration of RANSAC accuracy in inlier selection for two different models. For the homography, the error measured between the green and red feature would result in a large point-to-point distance, and hardly register as a correspondence. The red feature shows small point-to-point distance to the yellow feature using the homography. On the other hand, the essential matrix maps a red feature to a line (not a point) and error is measured as the point-to-line orthogonal distance. In this case, the distance to the red line from the green point can result in a smaller error and be falsely registered as a correspondence.




AIC (Kanatani and Kanazawa, 1999), Robust AIC (Torr, 1998) and Geometric Information Criterion (GIC) (Torr, 1999) all aim to determine the inherent geometric complexity of the image scene; see Gheissari and Bab-Hadiashar (2005) for a side-by-side evaluation. These algorithms determine the correct registration model by measuring the summation of the reprojection error along with a model complexity penalty, and choose the best model as the one with the smallest error.

In this work, we have adopted Torr's GIC (Torr, 1999),

$$\text{GIC} = \sum_{i}^{N} \frac{e_i^2}{\sigma^2} + \lambda_1 N d + \lambda_2 P. \tag{2.10}$$

Here, $e_i$ is the $i^{th}$ correspondence's reprojection error, $\sigma^2$ is the feature extraction variance, $N$ is the number of correspondences, $d$ is the dimension of the registration manifold (2 for homography and 3 for the essential matrix), $P$ is the number of parameters of the registration model (8 for homography and 5 for essential matrix) and $\lambda_1 = 2$ and $\lambda_2 = 4$ are penalty weights, as recommended in the literature.

For planar structures, the homography model is preferable. Given $N$ correspondences, the total number of variables in the two-view bundle adjustment is $2N + 8$ for the homography model, and $3N + 5$ for the essential matrix. Moreover, the homography model

provides greater accuracy in inlier correspondence establishment than the essential matrix does since the homography model is a point-to-point metric whereas the essential matrix is only a point to epipolar line mapping as can be seen in Figure 2.6. The homography model is, however, not feasible for non-planar structure, which is why we want to retain the ability to use the essential matrix for when we encounter complex structure on the hull. Figure 2.7 and Figure 2.8 demonstrates GIC's utility in automatic model fitting.

GIC score calculation on both synthetic and real images are shown in Figure 2.7 and Figure 2.8. We computed the score in the graph of Figure 2.7(d), by gradually changing the complexity of the structure from planar (Figure 2.7(a)) to non-planar (Figure 2.7(b)). Two sample synthetic images (Figure 2.7(c)) depict the projected scenes. The graph of the GIC score versus structure complexity (Figure 2.7(d)) shows that as we increase the non-planar nature of the structure, that essential matrix ($\mathsf{GIC}_\mathsf{E}$) eventually becomes a better model choice than the homography ($\mathsf{GIC}_\mathsf{H}$). This is because the reprojection error stays the same with the essential matrix model while the homography reveals increasing error as the structure becomes more complex.

For two sample pairs of real hull imagery, as shown in Figure 2.8, the top sequence of images show where GIC has automatically determined an essential matrix model as being correct, whereas for the bottom set of images GIC correctly chooses a homography. In both examples, the left and middle columns show two overlapping image pairs while the right column shows their inlier correspondences. Yellow dots indicate correspondences that satisfy the epipolar geometry constraint, while red circles indicate correspondences that are consistent with the homography constraint.

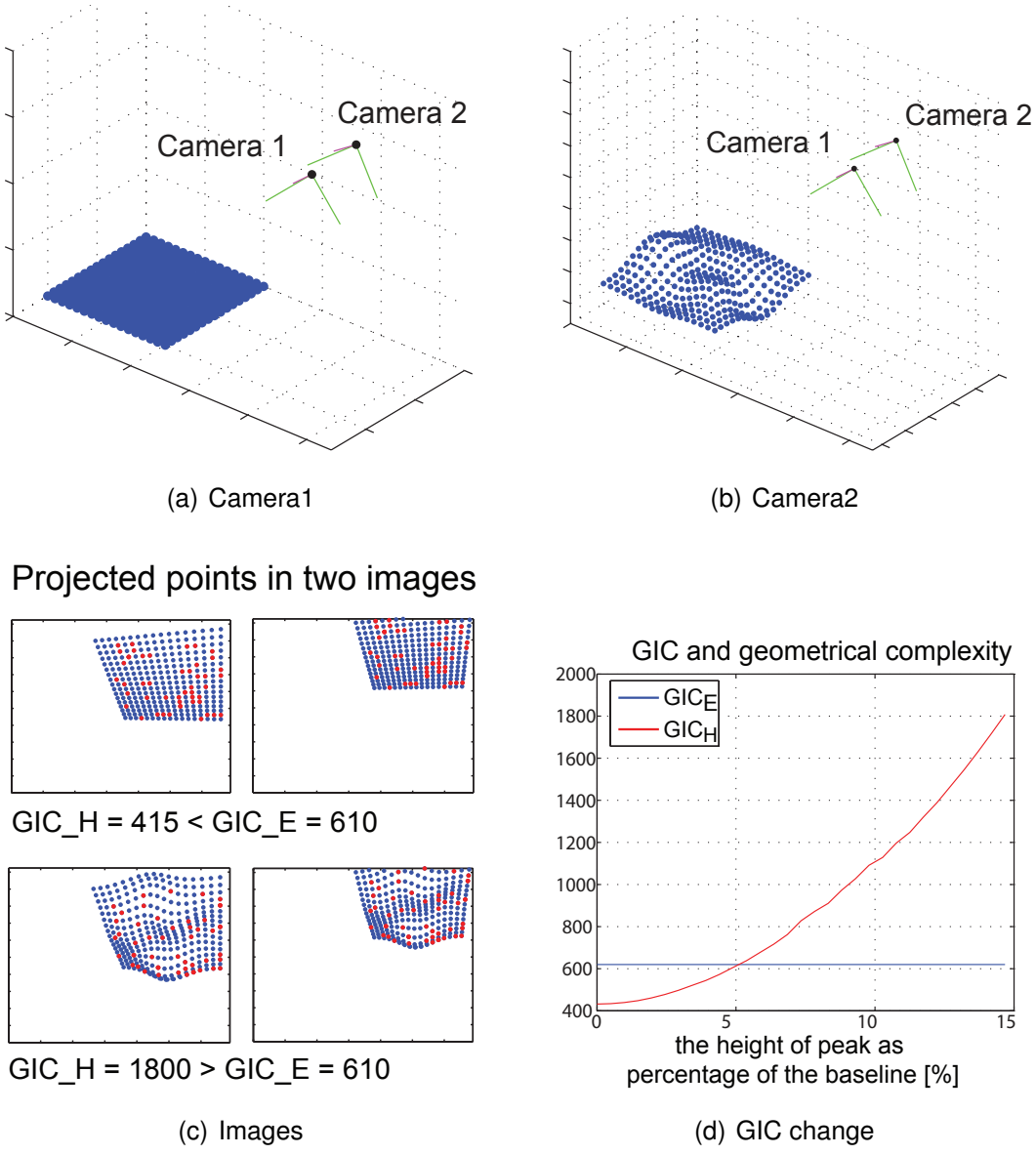
### 2.3.4 Two-view Bundle Adjustment

After selecting the proper model via geometric model selection, the model and set of correspondences are optimized within a two-view bundle adjustment. Horn's relative-pose algorithm (Horn, 1991) is applied prior to the bundle adjustment to refine the inliers set and provide an initial guess of the relative-pose. Then, a two-view bundle adjustment estimates the optimal relative-pose constraint between the two cameras.

#### 2.3.4.1 Essential matrix

When the model selection criteria results in choosing the essential matrix, the cost function is chosen such that it minimizes the reprojection error in both images by optimizing camera relative-pose (i.e., $\mathrm{R}, \mathbf{t}$) and the triangulated 3D structure points (i.e., $\mathbf{X}_n$) associated with the inlier image correspondences (Hartley and Zisserman, 2000):

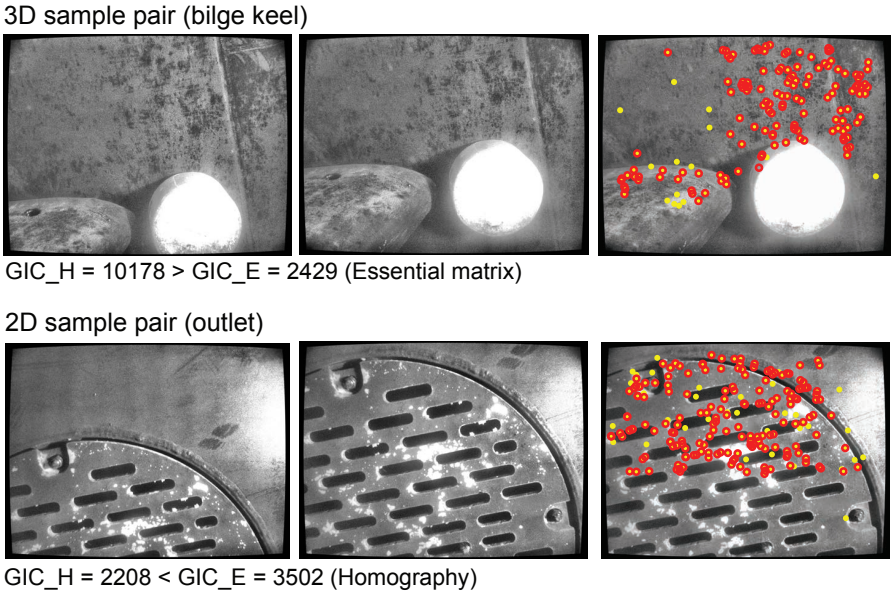
**Figure 2.7** GIC on synthetic data. Depiction of GIC's utility for automatic model selection as applied to a synthetic dataset. The blue dots represent the projected structure points in each image, while the red dots indicate the randomly sampled true correspondences as seen from both cameras. We vary the scene structure from being planar (a) to fully 3D (b). Two sample images are shown in (c) with their GIC score. The change of GIC score with respect to the complexity of structure is shown in (d)




(a) Camera1 (b) Camera2

(c) Images (d) GIC change

**Figure 2.8** GIC on real data. Depiction of GIC's utility for automatic model selection as applied to real hull data. Yellow dots indicate correspondences that satisfy the epipolar geometry constraint, while red circles are those which are consistent with the homography constraint.




$$\min_{\mathrm{R},\mathbf{t},\mathbf{X}_n} f(\mathrm{R},\mathbf{t},\mathbf{X}_n) = \sum_{n=1}^{N} \mathbf{e}_n^{\top}\mathbf{e}_n. \tag{2.11}$$

The cost function $f(\,\cdot\,)$ is the sum squared reprojection error taken over all $N$ correspondences. The camera projective matrices are $\mathrm{P}_i = \mathrm{K}[\mathrm{I}\,|\,\mathbf{0}]$ and $\mathrm{P}_j = \mathrm{K}[\mathrm{R}\,|\,\mathbf{t}]$ where the camera internal parameters, K, are the same for both cameras and known from calibration. Finally, the $n^{th}$ pixel reprojection error is defined as

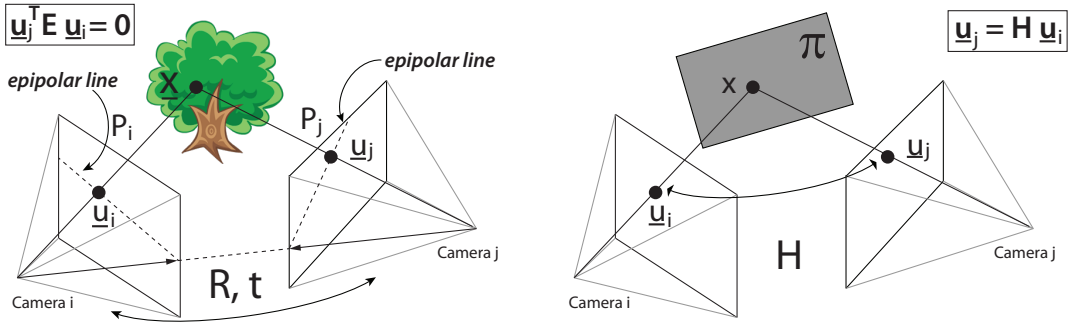
$$\mathbf{e}_n = \begin{bmatrix} \underline{\mathbf{u}}_{i_n} - \mathrm{P}_i\underline{\mathbf{X}}_n \\ \underline{\mathbf{u}}_{j_n} - \mathrm{P}_j\underline{\mathbf{X}}_n \end{bmatrix}, \tag{2.12}$$

where $\underline{\mathbf{u}}_{i_n}$ and $\underline{\mathbf{u}}_{j_n}$ are the homogeneous coordinate pixel locations for cameras $i$ and $j$, respectively, and $\underline{\mathbf{X}}_n$ is the homogeneous coordinate triangulated 3D structure point.

#### 2.3.4.2 Homography

When the model selection criteria results in choosing the homography, the objective function is chosen to minimize the sum squared reprojection error using a plane induced homography model (Hartley and Zisserman, 2000). The optimization is performed over the homography parameters (R, $\mathbf{t}$, $\mathbf{n}$, $d$) and the optimal pixel correspondences $\hat{\mathbf{u}}_{i_n}$, which satisfy the homography mapping exactly (i.e., $\underline{\hat{\mathbf{u}}}_{j_n} = \mathrm{H}\underline{\hat{\mathbf{u}}}_{i_n}$ where $\underline{\hat{\mathbf{u}}}_{i_n}$ is the homogeneous

**Figure 2.9** Illustration of sparse bundle adjustment (SBA). (a) When model selection chooses essential matrix as the proper model, SBA solves for the relative-pose, $(\mathrm{R}, \mathbf{t})$, and the 3D structure, $\mathbf{X}$, by minimizing reprojection error (2.11). $\underline{\mathbf{u}}_i$ and $\underline{\mathbf{u}}_j$ are pixel points in images where $\mathrm{P}_i$ and $\mathrm{P}_j$ are the projection matrices. (b) For homography, the point is constrained on a plane $(\pi)$, and SBA solves for the optimal reprojected points $(\underline{\hat{\mathbf{u}}}_i)$, plane parameters $(\mathbf{n}, d)$, and relative pose $(\mathrm{R}, \mathbf{t})$ as in (2.14)




(a) Essential matrix (b) Homography

coordinate version of $\hat{\mathbf{u}}_{i_n}$).

The plane induced homography, H, is written in terms of the world plane, $\pi = [\mathbf{n}^\top, d]^\top$, and camera relative-pose, R and $\mathbf{t}$, as

$$\mathrm{H} = \mathrm{K}(\mathrm{R} - \mathbf{t}\mathbf{n}^\top/d)\mathrm{K}^{-1}. \tag{2.13}$$

Here, the world plane normal, $\mathbf{n}$, and orthogonal distance, $d$, are expressed in camera $i$'s reference frame. Given the $N$ correspondences between the two images, the optimization solves for the optimal value of homography parameters together with the ideal image points,

$$\min_{\mathrm{R},\mathbf{t},\mathbf{n},d,\underline{\hat{\mathbf{u}}}_{i_n}} f(\mathrm{R}, \mathbf{t}, \mathbf{n}, d, \underline{\hat{\mathbf{u}}}_{i_n}) = \sum_{n=1}^{N} \mathbf{e}_n^\top \mathbf{e}_n, \tag{2.14}$$

where

$$\mathbf{e}_n = \begin{bmatrix} \underline{\mathbf{u}}_{i_n} - \underline{\hat{\mathbf{u}}}_{i_n} \\ \underline{\mathbf{u}}_{j_n} - \mathrm{H}\underline{\hat{\mathbf{u}}}_{i_n} \end{bmatrix}, \tag{2.15}$$

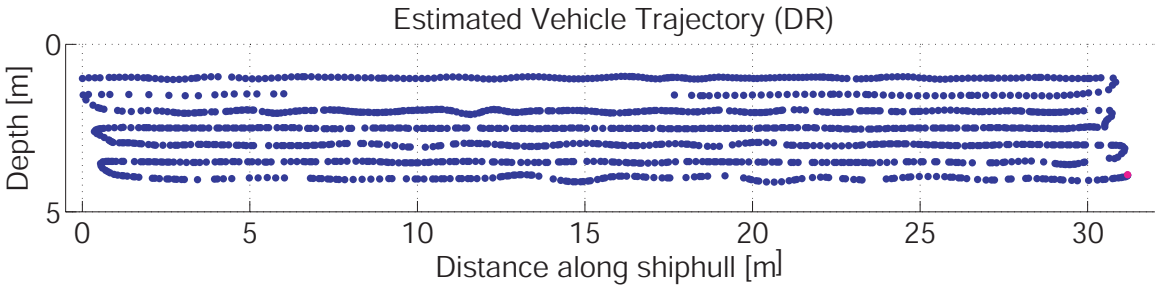
and $\underline{\mathbf{u}}_{i_n}$ and $\underline{\mathbf{u}}_{j_n}$ are defined as before.

Finally, in either case (homography or essential matrix), the two-view bundle adjustment results in a 5-DOF bearing-only camera measurement (2.3), and a first-order estimate of its covariance (Haralick, 1994), which is then published as a constraint.

**Figure 2.10** Experimental setup for *USS Saratoga* hull inspection using the HAUV. We mapped a section of the aircraft carrier following the trajectory as depicted in (c). The mission started on the top left corner toward the bottom right corner following a lawn mower grid pattern. On the second track-line, the vehicle moved without camera measurements, yielding a blank section in the middle of the survey.

(a) *USS Saratoga* aircraft carrier


(b) HAUV




(c) Hull-relative trajectory estimated from odometry

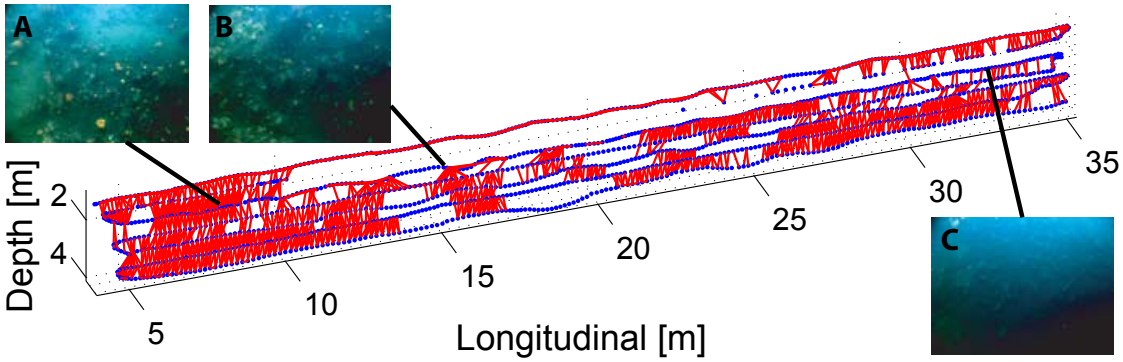
## 2.4 Implementation and Results

In May 2008, over 1,300 underwater images of the aircraft carrier *USS Saratoga* (Figure 2.10) were collected at AUVFest2008 in Newport, Rhode Island.[1] The experiment was done in collaboration with MIT and Bluefin Robotics using the autonomous underwater hull inspection vehicle, the HAUV (Hovering Autonomous Underwater Vehicle) (Vaganay et al., 2005) (Details in Appendix §A.1). The experiment consisted of seven $30\ \mathrm{m}$ legs of a boustrophedon survey, each spaced $0.5\ \mathrm{m}$ apart in depth. The camera and Doppler velocity log (DVL) were mounted on tilt actuators so that they approximately maintained a nadir view to the hull. The standoff position of the robot was controlled at $1.5\ \mathrm{m}$ from the hull throughout the experiment with a horizontal trajectory speed of $0.5\ \mathrm{m/s}$. In this experiment, the incremental smoothing and mapping (iSAM) algorithm was selected as the SLAM back-end.

The final trajectory resulting from visual SLAM with model selection is shown in Figure 2.11. The red lines indicate pose constraints derived from image pairs. The camera was

[1] AUVFest is an Office of Naval Research sponsored AUV field demonstration event.

**Figure 2.11** SLAM result from the *USS Saratoga*. The camera-derived SLAM pose constraints for inspection of the *USS Saratoga* are shown with successful camera links in red. Each vertex represents a node in the pose-graph where the red lines indicate the 5-DOF camera measurements. Images A and B show sample images from the feature-rich region and image C presents an example image from the feature-less region.

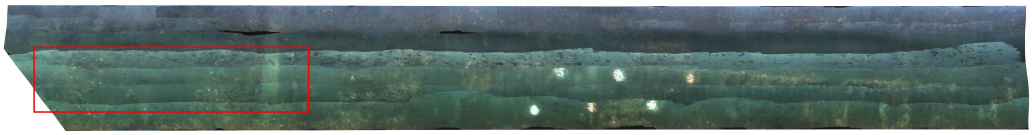



restarted while the robot was returning to the bow in the second leg of the trajectory. This resulted in a blank section in the middle of the second track-line during which the uncertainty ellipsoid inflated as the robot moved without camera measurements over this section. However, once the camera was restarted at the end of the second leg, the SLAM algorithm re-localized the vehicle by adding additional pose constraints to the previous track (first leg). The distribution of visual features strongly affects the camera measurement distribution. As the robot hovers and moves from the feature-less region (C in Figure 2.11 toward stern right) to the feature-rich region (A and B in Figure 2.11 toward bow left), we see a strong increase in the number of camera measurements.

As a by-product of the SLAM trajectory estimation, the 3D structure of the ship hull can be reconstructed using the final trajectory estimate and the pairwise image correspondences. Surface fitting with texture mapping was conducted to generate a 3D photomosaic (Figure 2.12). Note that the major purpose of having the 3D photomosaic is to have qualitative verification of the SLAM result. Because the photomosaic was generated using the final SLAM estimate and back projected image points (without additional blending at the image processing level), the continuity in the image seams between tracks illustrates that the resulting pose-graph is self-consistent. During the mission, six artificial targets, pre-installed on the hull to verify the SLAM navigation performance, are all found as indicated by the six white circles that appear in the texture-mapped reconstruction.

From the SLAM point of view, camera measurements provide a means to reduce the uncertainty in the pose estimation. Without such a sensor, a dead reckoned trajectory will inevitably accumulate pose and map uncertainty as a mission proceeds. Figure 2.13 shows the estimate error in pose uncertainty versus the total path length that the underwater robot

**Figure 2.12** Photomosaic results on the *USS Saratoga*. Triangulated 3D points can be meshed to obtain a smooth surface reconstruction, and texture mapped to create a 3D photomosaic as in (a). The six white dots that appear in (a) are the targets used for visual verification of the hull inspection utility of the algorithm.



(a) Texture mapped reconstruction

(b) Zoomed view of inset

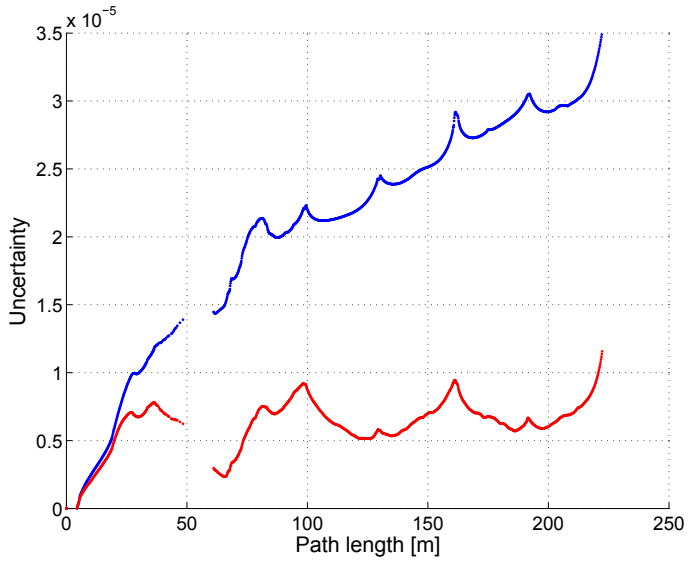
traveled. The uncertainty of the robot pose using DR navigation is represented by the blue line, and shows unbounded and increasing uncertainty along the total path as it localizes itself based upon DVL-based integration. As can be seen from the red lines, camera-constraints significantly reduce the navigation uncertainty, keeping it bounded.

## 2.5 Conclusion

This chapter presented an extension to the VAN algorithm that uses geometrical model selection to provide accurate correspondences in the image-based motion estimation. This extension increases the robustness and accuracy of the camera-derived pose constraint since a large portion of the hull is locally flat. We presented results for mapping a $30\ \mathrm{m}$ by $5\ \mathrm{m}$ section of the hull. While we do not have independent ground-truth to validate our trajectory estimate, we note that the recovered trajectory results in a smooth surface reconstruction, indicating that the recovered poses must be highly self-consistent.

The next step in our research is to automatically determine which areas of the hull are feature-rich and useful for visual SLAM. This information will then be coupled back into the SLAM navigation estimate for an optimal mapping policy. As shown in the current data set, large areas of the hull occupy nearly feature-less regions, while other regions exhibit a high density of features. For these scenarios, it would be ideal to have the vehicle autonomously return to a previously known feature rich area in an intelligent way whenever its pose uncertainty grows too high. If this can be accomplished, the vehicle will be able to

**Figure 2.13** Uncertainty versus path length plot. Whereas the uncertainty of DR navigation (blue line) shows unbounded growth, the uncertainty of SLAM is bounded from camera loop-closures (red line).




maintain an upper bound on its pose uncertainty at any point along the hull, and, thereby, quantitatively guarantee 100% survey coverage for inspection tasks.

# CHAPTER III

# Visual Saliency for SLAM

## 3.1 Introduction

Underwater visual simultaneous localization and mapping (SLAM) is challenging in many aspects (e.g., lack of ambient light, amorphous features, limited field of view). One of the main challenge is that the spatial distribution of visual features is not uniformly distributed within the environment. For example, Figure 3.1(b) depicts a representative underwater visual SLAM result obtained on a ship hull. Here, successful camera-derived measurements (i.e., red links) occur when feature-rich distributions are prevalent. On the other hand, in visually feature-poor regions, the camera produces few, if any, constraints. Thus, the distribution of visual features in the target environment dominates the spatial availability of camera-derived constraints, and hence, the overall precision of our SLAM navigation result. This indicates that visual saliency strongly influences the likelihood of making a successful pairwise camera measurement. When spatially overlapping image pairs fail to contain any locally distinctive textures or features, image registration fails. Hence, having the ability to *quantitatively* evaluate the registration utility of image keyframes would greatly aid underwater visual SLAM.

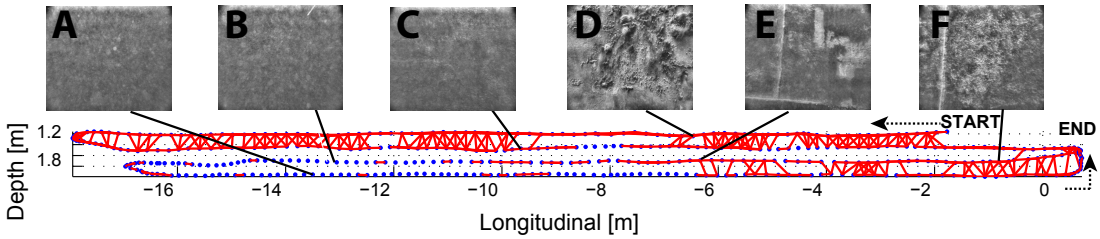
### 3.1.1 Motivation

One of the most important and difficult problems in pose-graph visual SLAM is determining a loop closure event, where the loop-closure in visual SLAM is obtained by recognizing previously viewed scenes. This recognition process involves identifying possible candidate image pairs and attempting to obtain a camera-derived relative-pose constraint. By correctly associating and registering corresponding image pairs, the uncertainty of both the map and the robot pose can be reduced and bounded.

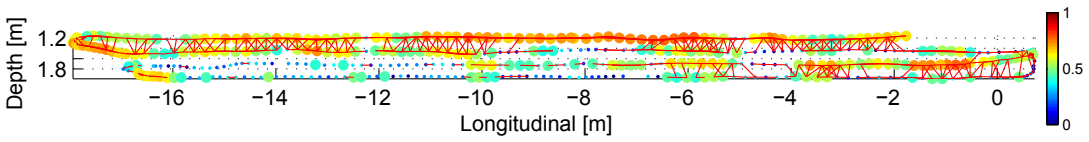
This task necessarily involves choosing optimal loop-closure candidates because (*i*) the

**Figure 3.1** Local and global saliency maps on *R/V Oceanus*. Pictured are the hull inspection SLAM results for a survey of the outboard hull of the *R/V Oceanus* which motivates our development of local and global visual saliency metrics. (a) Depiction of the *R/V Oceanus*' stern with the HAUV in view. (b) SLAM trajectory of the HAUV with successful cross-track camera registrations depicted as red edges. The histogram equalized sample images shown above are indicative of the image feature content within that region of the hull. Note that the density of cross-track links is spatially correlated with image feature content. (c) Our normalized local saliency measure, $S_L$, which spans from 0 to 1, is overlaid on top of the SLAM graph and correlates well with camera link density. (d) Our normalized global saliency measure, $S_G$, which also spans from 0 to 1, is overlaid on top of the SLAM trajectory. For easier visualization, we have enlarged nodes with $S_G > 0.4$. Note that global saliency can be used to identify visually distinct (i.e., rare) scenes with respect to the rest of the hull, such as distinct port openings, weld seams, and hull discolorations.

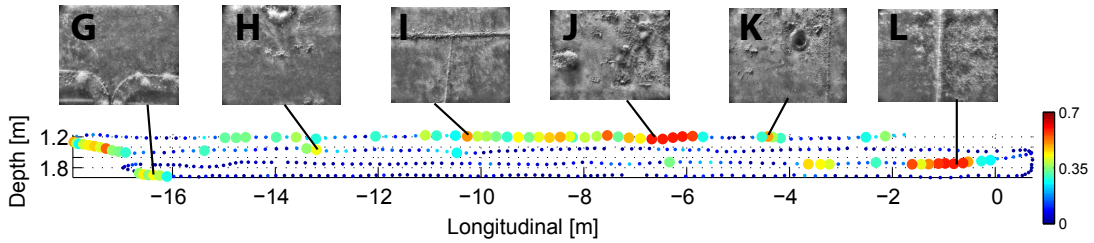

(a) *R/V Oceanus*




(b) SLAM pose-graph result for *R/V Oceanus*




(c) Local saliency map ($S_L$) on *R/V Oceanus*




(d) Global saliency map ($S_G$) on *R/V Oceanus*

cost of estimating the camera-derived relative-pose constraint is computationally expensive (per loop closure candidate) and (*ii*) adding unnecessary/redundant measurements may result in overconfidence (Ila et al., 2010). In this chapter, we focus on pose-graph visual SLAM methods where nodes in the graph correspond to image keyframes and measurements appear as constraints (links[1]) between nodes.

One way to intelligently hypothesize link candidates is to examine the utility of future expected measurements—an elegant method for measuring such uncertainty is to use information gain (Ila et al., 2010). Typically, information gain refers to either Fisher information or mutual information. By definition, the Fisher information is closely related to the Cramer Rao Lower Bound (Bar-Shalom et al., 2001), while mutual information is derived from entropy (Bryson and Sukkarieh, 2005), as defined by Shannon (1948). Example usages of information gain for control can be found in Sim and Roy (2005), Vidal-Calleja et al. (2006), and Bryson and Sukkarieh (2005), where the proposed control schemes evaluate the information gain of possible measurements and lead the robot on trajectories that reduce the overall navigation and map uncertainty. Another application of information gain in SLAM can be found in the task of creating link hypotheses, where only informative links (measurements) are proposed to be added to the pose-graph (Ila et al., 2010).

In the approaches described above, an equal likelihood of measurement availability is assumed. In other words, the information gain measure assesses the geometric value of adding the constraint *without regard to if, in fact, the constraint can be made.* In our scenario, camera-derived measurements may not be uniformly available within the environment due to the fact that the spatial distribution of visual features can vary greatly. For example, Figure 3.1(b) depicts an underwater environment where there is a large variance in feature distribution (the surface of a ship hull). Here, successful camera-derived measurements (red links) occur when feature-rich distributions are available, and in feature-poor regions the camera produces few, if any, constraints.

In this chapter, we develop two novel metrics to measure an image's visual saliency for SLAM. The first is local saliency, which is capable of measuring image registrability correlates well with successful camera links (Figure 3.1(c)). The second is global saliency, which identify visually distinct (i.e., rare) scenes with respect to the rest of the hull (Figure 3.1(d)).

[1]We call the process of hypothesizing possible loop-closure candidates "link proposal", because a measurement will act as a "link" (i.e., constraint) between two nodes in our pose-graph framework.

### 3.1.2 Review on Saliency and Bag-of-Words

The term "saliency" used in this thesis refers to a measure of how distinctive an image is and is related to seminal works by Itti and Koch (2001) and Kadir and Brady (2001). Itti and Koch (2001) showed that the human perception process includes detecting salient regions within a scene and actively controlling our attention to be on the detected region. Kadir and Brady (2001) used entropy as a measure of randomness, and in their work, computed entropy for a local patch and used it in detecting features within an image. If a patch has higher entropy, it likely has more randomness and, thus, is considered a feature. Lee and Song (2010) extended this entropy approach to color images using the Hue Saturation Value (HSV) color-space representation for detecting image features. Similarly, Johnson-Roberson (2010) combined HSV channel entropy with texture entropy from a Gabor filtered image to compute a combined saliency score for color images. This approach was shown to produce usable saliency maps derived from down-looking underwater seafloor imagery; however, its broad application is limited due to its reliance on color imagery (i.e., it is not applicable to grayscale).

As an alternative to the above channel-based methods, several bag-of-words (BoW) saliency representations have recently been explored (Nowak et al., 2006; Marchesotti et al., 2009; Csurka et al., 2004; Toldo et al., 2009). Originally developed for text-based applications, the general bag-of-words approach was first adapted and expanded to images by Leung and Malik (2001); Sivic and Zisserman (2003), and Csurka et al. (2004). In this method, each image is considered to be a document, and each feature descriptor found in the image corresponds to a word in the document. Feature descriptors (Sivic and Zisserman, 2003; Csurka et al., 2004; Nowak et al., 2006), local regions (Fei-Fei and Perona, 2005), or patches (Barnard et al., 2003) can serve as visual words. This representation distills an image to a set of vocabulary words, allowing for aggregate content assessment and enabling faster search. This approach has been successfully applied in diverse applications such as image annotation (Wu et al., 2010), image classification (Lazic and Aarabi, 2007), object recognition (Hu et al., 2009; Larlus et al., 2010) and appearance-based SLAM (Cummins and Newman, 2009; Angeli et al., 2008; Kawewong et al., 2010; Shahbazi and Zhang, 2011).

To develop a measure of visual saliency using a BoW representation, Csurka et al. (2004) explored the use of a BoW image model to selectively extract only "salient" words from an image and referred to them as a bag-of-keypoints. Toldo et al. (2009) explored the use of a histogram of the distribution of words as a global signature of an image, and only salient regions were sampled to solve an object classification problem. In appearance-based SLAM, Fast Appearance-Based Mapping (FAB-MAP) by Cummins and Newman (2009)

uses a model that learns common words during an offline training phase to down-weight common (non-salient) words. Out of the non-model-based approaches, a popular statistics-based approach is called term frequency-inverse document frequency (tf-idf). This statistic is widely used in classification problems (Sivic and Zisserman, 2003; Nister and Stewenius, 2006) due to its simplicity and robustness. It emphasizes rare occurrences resulting in higher tf-idf scores for statistically salient words and typically can be learned online.

In this thesis, we also use a BoW model to measure saliency based on a statistics-based bag-of-words approach to define two visual saliency metrics—local and global saliency. These novel measures will be presented in the following sections.

### 3.1.3 Overview of Our Approach

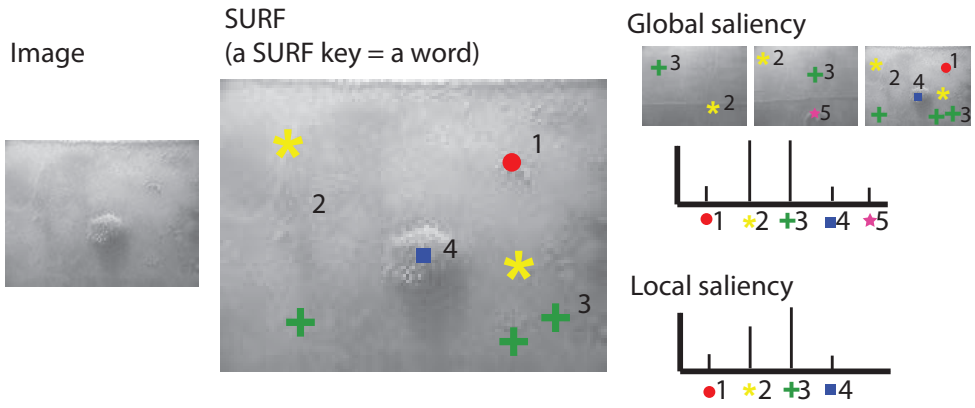
To quantitatively measure the level of visual saliency, we focus on two different measures of saliency: local saliency (i.e., intra-image) and global saliency (i.e., inter-image). Both are computed using a BoW model for image representation. Registrability refers to the intrinsic feature richness of an image (i.e., the amount of texture an image has). The lack of image texture, as in the case of mapping an underwater environment with feature-poor regions (e.g., images $A$ and $B$ in Figure 3.1(b)), prevents image registration from being able to measure the relative-pose constraint. However, texture is not the only factor that defines saliency—an easy counterexample is an image of a checkerboard pattern or a brick wall. Images of these type of scenes have high texture, but likely will fail registration due to spatial aliasing of common features. Thus, we develop local and global saliency as two different measures of image registrability in this section.

A brief illustration of the overall process is depicted in Figure 3.2. We generate a vocabulary online by projecting 128-dimension Speeded Up Robust Features (SURF) keys, developed by Bay et al. (2008), to words using a BoW image model. Given a point of interest, the 128-dimension feature vector is computed from a set of Haar wavelet responses around that point, and is used as a descriptor. Once mapped to a BoW representation, we examine the intra-image histogram of word occurrence for the local saliency measure, and score the saliency level by evaluating its entropy. For global saliency, the inter-image count of word occurrence throughout all previously-seen images is examined. This count is used to compute the global saliency score by measuring inverse document frequency (idf).

### 3.1.4 BoW Vocabulary Generation

Before defining our BoW saliency metric, we first need to outline how we construct our vocabulary. Offline methods for vocabulary generation typically use a clustering algorithm

**Figure 3.2** Depiction of vocabulary construction and saliency computation. SURF keys are extracted from an image and are used to update local and global statistics. Entropy from the local histogram detects intra-image feature richness, while inverse document frequency measures inter-image rarity.




on a representative training dataset. An example of this type of offline method is the FAB-MAP algorithm, which has shown remarkable place recognition results using a pre-trained vocabulary (Cummins and Newman, 2008). Another offline vocabulary building technique by Nister and Stewenius (2006) pre-trains a vocabulary tree to cluster the words with a tree hierarchy for efficient recognition. Other studies have focused on online methods, which incrementally build a vocabulary during the data collection phase (Angeli et al., 2008; Nicosevici and Garcia, 2009; Kawewong et al., 2010; Shahbazi and Zhang, 2011). Position Invariant Robust Feature (PIRF) based navigation by Kawewong et al. (2010) uses this type of online approach, using only consistent Scale Invariant Feature Transform (SIFT) keys to incrementally build the vocabulary. They have showed performance comparable to other state-of-the-art appearance-based SLAM methods. In order to achieve fast and reliable online loop-closure detection, Shahbazi and Zhang (2011) use locality sensitive hashing to build the vocabulary *in situ*. Also, incremental online clustering schemes have been used by Nicosevici and Garcia (2009) to update the vocabulary clusters incrementally.

One advantage to offline methods is that an optimal distribution of vocabulary words (clusters) in a descriptor space can be guaranteed; however, one disadvantage is that the learned vocabulary can fail to represent words collected from totally different datasets as demonstrated by Kawewong et al. (2010). Online construction methods provide flexibility to adapt the vocabulary to incoming data, though equidistant words (clusters) are no longer guaranteed.

Two guidelines underpin our vocabulary building procedure: *(i)* we do not want to assume any prior knowledge of the appearance of the environment, and *(ii)* the vocabulary must be visually representative. With this in mind, we have decided to pursue an online

construction approach that initially starts from an empty vocabulary set, similar to the algorithms by Angeli et al. (2008) and Kawewong et al. (2010). SURF features are extracted from the incoming image and are matched to existing words in the vocabulary based on the Euclidean inner product (SURF descriptors are unit vectors). Whenever the direction cosine is larger than a threshold (0.4—0.6 in our experiments), we augment our vocabulary to contain the new word.

Related to this vocabulary generation, two issues need to be discussed in detail. First, we will discuss the reason for choosing online vocabulary generation instead of using pre-trained model-based, and compare our approach to the state-of-art model-based approach, FAB-MAP. Second, the choice of image feature detector/descriptor will be discussed. Our approach is based on SURF instead of SIFT, though SIFT and SURF are often considered to have similar performance, they yield distinctly different behavior in our saliency application.

**Model-based versus statistics: Why not use FAB-MAP?**

FAB-MAP is a model-based approach to the place recognition problem. It evaluates the probability of the current location in appearance-space given measurements (up to the current time), and selects a location. Using their Bayesian estimation formulation, FAB-MAP learns common objects from their model. It has an ability to learn common objects and assign a small probability to those words. When measurements are full of common objects (e.g, bricks), it is likely that we will find an equally positive match from all images (i.e., we see it everywhere). In other words, FAB-MAP has an ability to diminish the effect of common words in their model, though they are not computing the saliency score directly. Later in FAB-MAP 2.0, a more direct saliency measure appears when they calculate the most useful candidates to improve the recognition speed by pruning out unlikely loop-closure candidates. The saliency score they use is defined below in terms of information,

$$I = -\ln p(z_i|z_{p_i}), \tag{3.1}$$

and indicates the information content of a word ($z_i$) where $z_{p_i}$ is the parent of $z_i$. The conditional probability, $p(z_i|z_{p_i})$, has been trained offline using a Chow-Liu tree (Chow and Liu, 1968) building process. The FAB-MAP authors mention that this information is training set dependent. For example, some features might be very rare in the training set but very common in the actual map.

Regarding this issue, we chose to use a statistics-based approach for the saliency score using online vocabulary construction rather than a pre-trained model-based approach. By

doing so, we could achieve (*i*) online vocabulary building capability and (*ii*) a direct measure of the image saliency level. FAB-MAP's strength is at the modeling of the co-occurrence of words by training in a Chow Liu tree. However, to learn this dependency between words, they need to learn the tree in an offline process, and the online training of this process has been remarked as future work of FAB-MAP. Moreover, there exists several successful online implementations based on statistics (Sivic and Zisserman, 2003; Angeli et al., 2008; Kawewong et al., 2010). Their advantage is that the performance of model-based approach decreases when the environments differ considerably between the trained and actual data (Kawewong et al., 2010). Since our focus is on having a saliency score that is reliable, even for large changes in environment, online performance is essential.

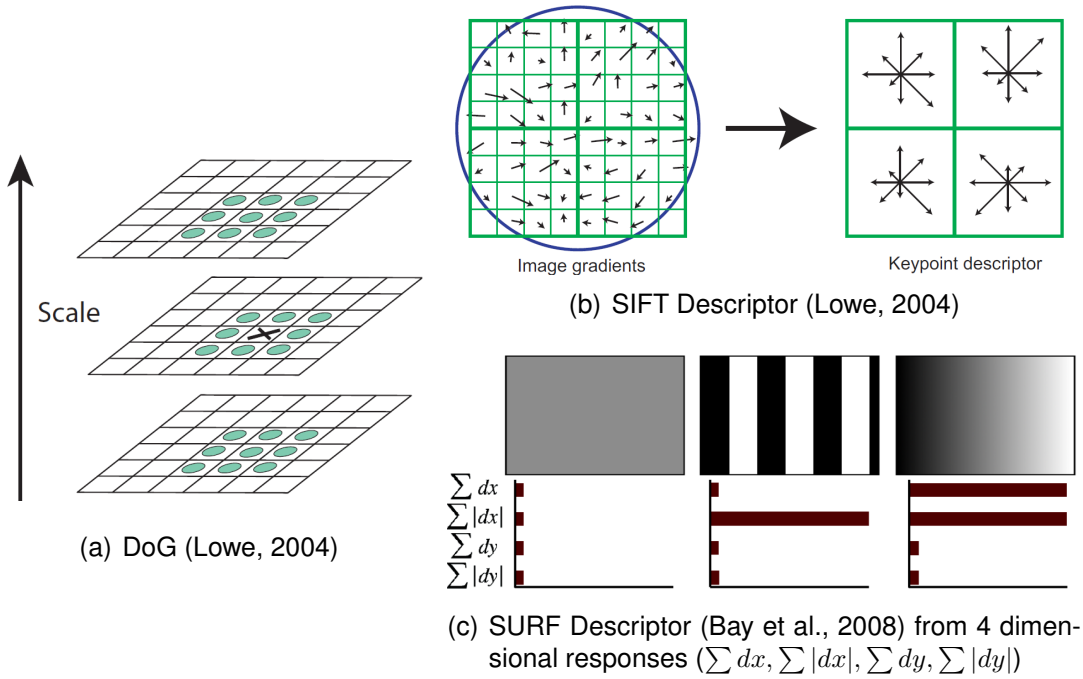
On the second issue, a direct measure of visual saliency, dealing with common words (saliency of words) in model-based approach is incorporated within the Bayesian estimation formulation. In other words, in this type of formulation, no direct measure of saliency is available. Instead, information (3.1) is defined in terms of conditional probability. However, again this conditional probability has been trained offline, which circles back to the discussion of online/offline vocabulary generation.

**SURF versus SIFT: Why not use SIFT?**

The second discussion we continue is about the selection of feature descriptor in vocabulary generation. We evaluated the usage of both 128-dimension SIFT and 128-dimension SURF descriptors and found that SURF features tend to perform better for our saliency calculation. Therefore, we chose to use SURF features in our vocabulary construction. The SIFT descriptor is built by calculating the gradient orientation histogram, whereas the SURF descriptor is built from a set of Haar wavelet responses (Figure 3.3). Due to the noise sensitivity of the gradient orientation calculation, we found that SIFT's descriptor tends to assign two similar texture patches as two distinct words, whereas SURF's wavelet descriptor tends to assign them to the same type of word. (This is similar to what Johnson-Roberson (2010) noted when comparing a Gabor filter for texture detection versus gradient-based methods.)

Finally, one additional point worth noting is that we pre-blur imagery before running SURF. This is done to gently force SURF to return larger scale features. As shown in Figure 3.4, we conducted a test to see the effect of this pre-blurring on underwater imagery. The depicted histogram-equalized sample image is "noisy" due to its accentuation of particles in the water column and the effect of back-scattering. Processing the image at full scale makes the SURF descriptor sensitive to this high-frequency noise and, thus, its descriptors distinctive to each other. While this distinctiveness can be beneficial for putative corre-

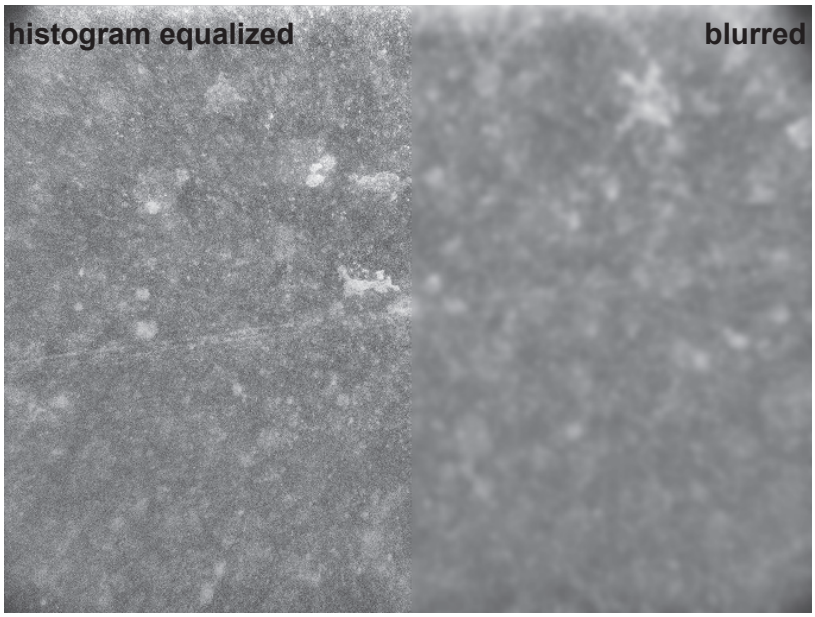
**Figure 3.3** Illustration of SIFT and SURF descriptors. (a) Both SIFT and SURF are based on Difference of Gaussian (DoG) filtering. When describing an image keypoint, (b) SIFT examines the gradient between Gaussian filtered images at different scales, and computes the histogram of binned orientation to build its descriptor. (c) SURF, on the other hand, builds descriptor from Haar wavelet responses. Using a 4x4 array, instead of 2x2 as in the illustration, SIFT computes a histogram over 8 orientations, while SURF computes extended 8 dimensional responses to result in a 128 dimensional vector for both descriptors. These extended responses are computed by separately summing basic 4 dimensional responses for positive and negative values $(\sum_{dx>0} dx, \sum_{dx>0} |dx|, \sum_{dy>0} dy, \sum_{dy>0} |dy|, \sum_{dx<0} dx, \sum_{dx<0} |dx|, \sum_{dy<0} dy, \sum_{dy<0} |dy|)$.



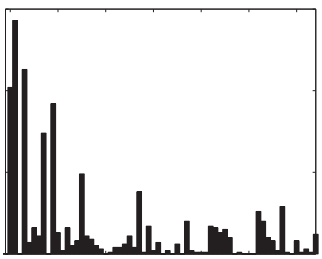

spondence matching, it is detrimental in vocabulary generation for the purpose of saliency detection. When the image contains particles and noise as in the sample image, these distinctive feature keys get mapped to different words, which artificially increases the entropy in our BoW histogram (Figure 3.4(b)). However, this undesirable effect can be reduced by either pre-blurring the image (Figure 3.4(c)), or, by (equivalently) forcing SURF to return larger scale features (Figure 3.4(d)). In practice, we found it easier to use the pre-blurring approach so that we could employ commonly available SURF libraries without the need for any custom modification.

Overall, due to the online vocabulary generation, the resulting vocabulary size will be increasing over the mission time as can be seen in Figure 3.5. However, increase in the vocabulary size slows down and converges over time as the vehicle collects enough representative images within the environment. Because of the pre-blurring and coarse clustering the size of vocabulary is typically small.

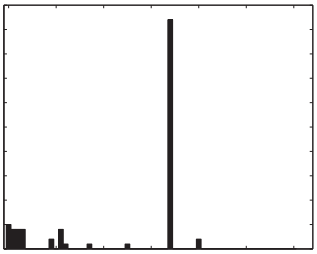
**Figure 3.4** Effect of pre-blurring versus scale-forced SURF. The effect of pre-blurring and scale-forced SURF detection for underwater image saliency are depicted. Image (a) shows the contrast-limited adaptive histogram equalized image on the left half and its blurred version on the right. The BoW histogram showing intra-image word occurrence and its normalized entropy score (i.e., local saliency, $S_L$) are shown for the raw image (b), the blurred image (c), and the scale-forced SURF detection (d). Note that (c) and (d) are comparable.



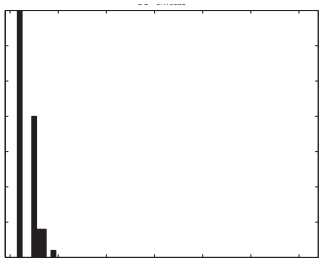

(a) Histogram equalized / blurred image



(b) Raw ($S_L$=0.76)

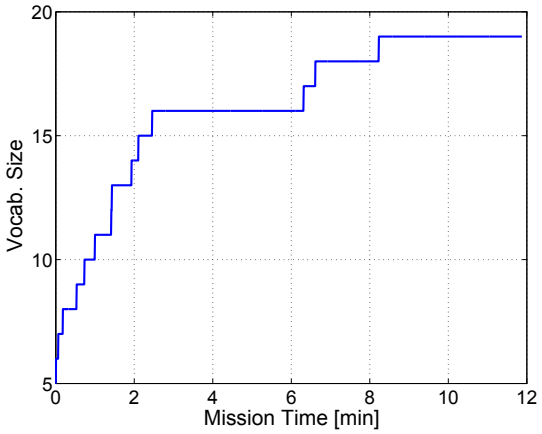


(c) Blur ($S_L$=0.35)

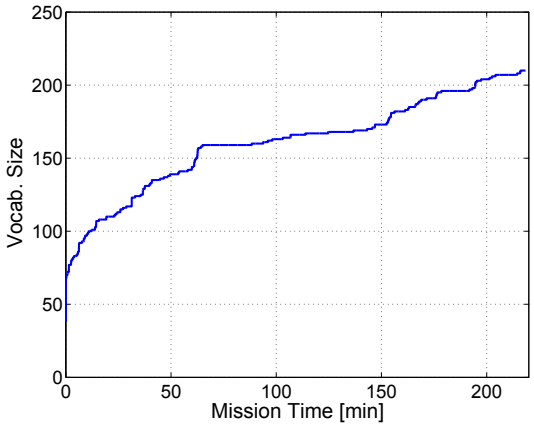


(d) Scale ($S_L$=0.48)

**Figure 3.5** Online vocabulary size over the course of a hull inspection mission. The vocabulary size is plotted for two different vessels versus elapsed mission time in minutes. Because of the pre-blurring and coarse clustering, the resulting vocabulary size is small (19 for *USCGC Venturous* (a) and 210 for *SS Curtiss* (b)).




(a) A small vessel (*USCGC Venturous*)




(b) A large vessel (*SS Curtiss*)

## 3.2 Saliency

Next, we define two different novel measures of saliency based upon this online vocabulary construction—local saliency and global saliency. We also provide a comparison and evaluation on both indoor and underwater images.

### 3.2.1 Local Saliency

One of the original usages of BoW is for texture recognition (Julesz, 1981; Varma and Zisserman, 2009). In these studies, an element of texture, a texton, can be expressed in terms of visual words using a BoW representation. These previous works mainly focused on recognition of texture using texton representation, whereas the local saliency we develop here examines the diversity of the textures to assess image content richness. We define local saliency as an intra-image measure of feature diversity. In defining the diversity, entropy has been widely used as a measure of randomness, though in other works it has been primarily used on image channels. For example, Lee and Song (2010) examined HSV channel entropy to compute a salient region within an image, but not to score the saliency level for the entire image. Similarly, Johnson-Roberson (2010) combined HSV channel entropy with Gabor-filtered intensity to compute image saliency. The limitation of this method, however, is the need for multiple channels to provide reliable performance and, which limits its usage to color images only. In contrast, we assess the diversity of words occurring within image $I_i$ by examining the entropy of its BoW histogram, which works

regardless of color channel availability:

$$H_i = -\sum_{k=1}^{W(t)} p(w_k) \log_2 p(w_k). \tag{3.2}$$

Here, $p(w)$ is the empirical BoW probability distribution within the image computed over the set of vocabulary words $\mathcal{W}(t) = \{w_k\}_{k=1}^{W(t)}$ where $W(t)$ is the size of the vocabulary, which grows with time since we build the vocabulary online. Hence, to normalize our entropy measure, we use the ratio of $H_i$ to the maximum possible entropy to yield a normalized entropy measure $S_{L_i} \in [0, 1]$:[2]

$$S_{L_i} = \frac{H_i}{\log_2 W(t)}. \tag{3.3}$$

This entropy-derived measure captures the diversity of words (descriptors) appearing within an image.

Figure 3.6 shows sample results for color and grayscale underwater hull imagery. For comparison, following Johnson-Roberson (2010), we also compute the hue channel histogram as an alternative measure of saliency. The results show that our normalized BoW entropy score yields comparable results to Johnson-Roberson (2010) in terms of discriminating image saliency for color images, but moreover, our measure works equally well for grayscale imagery too (where no hue channel is available).

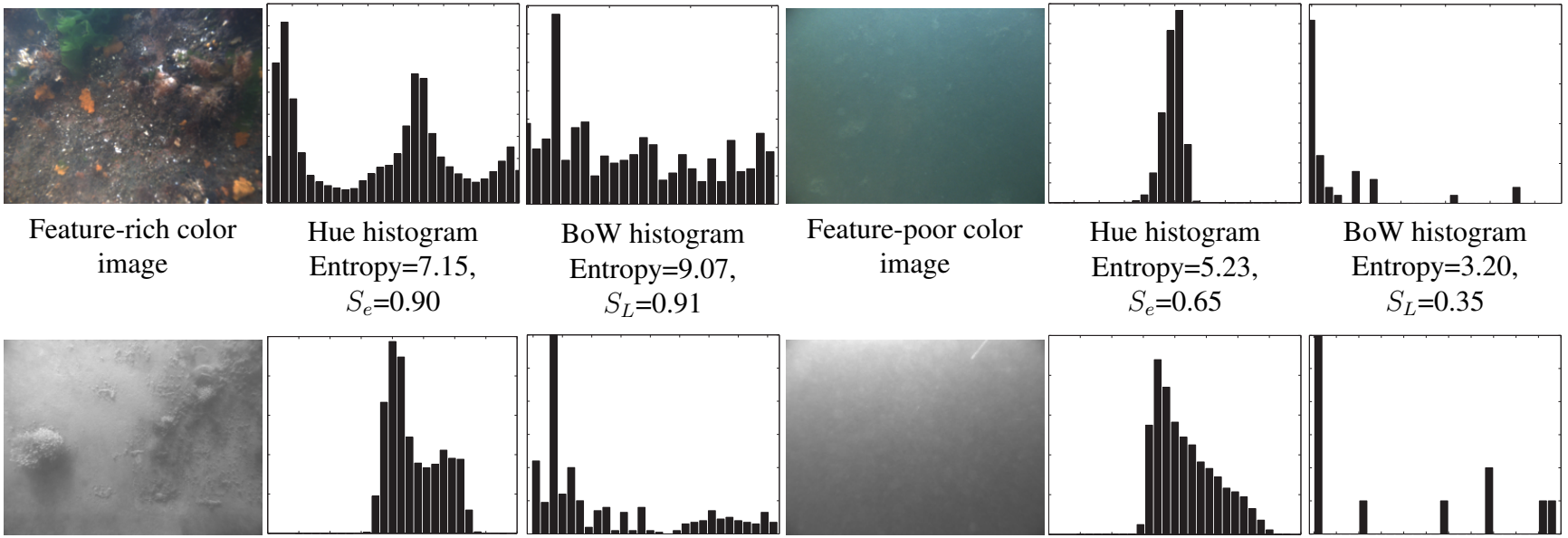
### 3.2.2 Global Saliency

We define global saliency as an inter-image measure of the uniqueness or rarity of features occurring within an image. The purpose of this measure is to identify unique regions of the ship hull that could be useful for guiding where the robot should revisit for attempting large scale loop-closure. In this scenario our SLAM navigation prior will typically be weak and we will, therefore, have to rely upon visual appearance information only for successful pairwise image registration. The sample imagery in Figure 3.1(b) depicts such a case.

To tackle this problem, we were motivated by a metric called term frequency-inverse document frequency (tf-idf), which is a classic and widely used metric in information retrieval (Jones, 1972; Salton and Yang, 1973; Manning et al., 2008). This metric was first adopted in the computer vision community by Sivic and Zisserman (2003) and subsequently shown to produce successful results in image-based classification (Moulin et al., 2010), place recognition (Knopp et al., 2010), scalable recognition using vocabulary tree

[2]The maximum entropy, $\log_2 W$, corresponds to a uniform distribution over a vocabulary of $W$ words.

**Figure 3.6** Local saliency example for color and grayscale ship hull images. Shown are images with varying levels of feature content, where the leftmost plot depicts the source image, the middle plot depicts the image intensity histogram (hue channel for color images and grayscale for monochrome images), and the rightmost plot depicts the BoW histogram. Normalized entropy for Hue and intensity histogram is provided below ($S_e$). For the color images, note that the hue channel histogram and the bag-of-words histogram are both able to distinguish the feature richness of the scene. However, for the grayscale imagery, note that the image intensity histogram fails to detect feature richness, whereas the BoW histogram still works well.



(Nister and Stewenius, 2006) and appearance-based SLAM (Angeli et al., 2008; Kawewong et al., 2010). In a computer vision application, tf-idf for a word $w$ is defined as:

$$t_w = \frac{n_{wd}}{n_d} \log \frac{N}{n_w}, \tag{3.4}$$

where $n_{wd}$ is the number of occurrences of word $w$ in document $d$; $n_d$ is the total number of words in document $d$; $N$ is the total number of documents seen so far; and $n_w$ is the number of documents with the occurrence of word $w$. The tf-idf captures the importance of a word (descriptor) appearing in a document (image) by penalizing common words.

Although tf-idf is prevalent in the text mining literature, its more fundamental form can be defined from inverse document frequency (idf) (Jones, 1972; Salton and Yang, 1973; Robertson, 2004); idf corresponds to the logarithm term in (3.4), and has a higher value for words seen less frequently throughout a history. In other words, we expect a high idf for words (descriptors) that are rare in the dataset. In computer vision, Jegou et al. (2009) used a variation of idf to detect "burstiness" of a scene, noting idf's ability to capture word frequency. Similar use is found in Chum et al. (2008), where the authors used idf as a weighting factor in the definition of their min-Hash similarity metric.

Unlike these previous studies that used idf as a weighting factor for robust matching, we use a sum of idf within an image $I_i$ to score its inter-image rarity:

$$\mathcal{G}_i(t) = \sum_{k \in \mathcal{W}_i} \log_2 \frac{N(t)}{n_{w_k}(t)}. \tag{3.5}$$

Here, $\mathcal{W}_i \subset \mathcal{W}(t)$ represents the subset of vocabulary words occurring within image $I_i$, $n_{w_k}(t)$ is the current number of documents in the database containing word $w_k$, and $N(t)$ is the current number of documents in the database. To guarantee independent sample statistics used in our idf calculation, only spatially distinct (i.e., non-overlapping) images are used to update $n_{w_k}(t)$ and $N(t)$. Since even a common word would be considered "rare" in (3.5) the first time it is observed (i.e., $n_{w_k} = 1$ on first occurrence in the database), $\mathcal{G}_i(t)$ needs to be updated through time. This paper uses an inverted index based update scheme combined with periodic batch updates to maintain $\mathcal{G}(t)$ for all images in the graph. The inverted index scheme (Manning et al., 2008) uses sparse bookkeeping for fast updates on the subset of $\mathcal{G}(t)$ that are impacted when changes in the statistics of $n_{w_k}(t)$ occur, and periodic batch updates that update $\mathcal{G}(t)$ for all nodes in the graph when changes in the number of documents $N(t)$ occur. At worst case, this update is linear in complexity with the number of image nodes. Lastly, as was the case with our local saliency measure, we normalize the rarity measure for image $I_i$ to have a normalized global saliency score

**Algorithm 1** Online vocabulary and saliency calculation.

**Require:** image $I_i$
**Require:** BoW vocabulary $\mathcal{W}(t)$ $\{\emptyset$ on first use$\}$
**Require:** idf statistics $N(t)$, $n_w(t)$
  Preblur and extract SURF features from $I_i$:
  $\mathcal{F}_i \leftarrow [f_1, f_2, \cdots, f_{n_f}]$

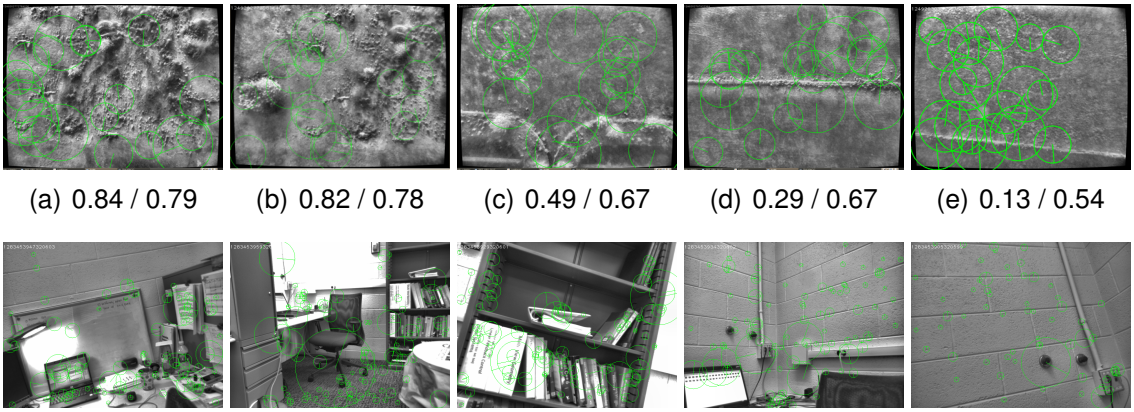
  $\{$compute intra-image BoW statistics$\}$
  initialize BoW histogram: $\mathcal{H}_i \leftarrow \emptyset$
  **for** each feature $f_j \in \mathcal{F}_i$ **do**
    find best vocabulary match $w_k \in \mathcal{W}(t)$
    **if** projection $f_j \cdot w_k >$ threshold **then** $\{$augment vocab.$\}$
      $\mathcal{W}(t) \leftarrow [\mathcal{W}(t), f_j]$, $w_k \leftarrow f_j$, $n_{w_k}(t) \leftarrow 1$
    **end if**
    increment histogram: $\mathcal{H}_i(w_k) \leftarrow \mathcal{H}_i(w_k) + 1$
  **end for**

  $\{$update inter-image idf statistics$\}$
  **if** $I_i$ does not overlap with images already in $N(t)$ **then**
    increment the document database: $N(t) \leftarrow N(t) + 1$
    **for** each $w_k \in \mathcal{W}(t)$ **and** $\mathcal{H}_i(w_k) > 0$ **do**
      increment word occurrence: $n_{w_k}(t) \leftarrow n_{w_k}(t) + 1$
    **end for**
  **end if**

  $\{$local saliency calculation$\}$
  Compute image $I_i$ BoW distribution: $p_i(w) \leftarrow \mathcal{H}_i / n_f$
  Compute image $I_i$ BoW entropy: $H_i \leftarrow$ Eqn (3.2)
  Compute image $I_i$ local saliency: $S_{L_i} \leftarrow$ Eqn (3.3)
  **if** $\mathcal{W}(t)$ was updated **then** $\{$vocab. was augmented$\}$
    Update $S_L$ for all previous images
  **end if**

  $\{$global saliency calculation$\}$
  Compute image $I_i$ rarity: $\mathcal{G}_i(t) \leftarrow$ Eqn (3.5)
  Compute image $I_i$ global saliency: $S_{G_i} \leftarrow$ Eqn (3.6)
  **if** $N(t)$ or $n_w(t)$ were updated **then** $\{$idf statistics changed$\}$
    Update $\mathcal{G}(t)$ for all affected images
    Update maximum rarity $\mathcal{G}_{\mathsf{max}}$
    Update $S_G$ for affected images
  **end if**

**Figure 3.7** Global saliency example for underwater (a)–(e) and indoor (f)–(j) images. Extracted features are marked with green circles. The score underneath each image is $S_G/S_L$ accordingly, where $S_G$ is the global saliency score and $S_L$ is the local saliency score. In both datasets, images are arranged from left to right in order of decreasing global saliency. Note that the global saliency score can be low even for texture rich scenes (e.g., (d), (e), (i) and (j)), indicating that the vocabulary words appearing in those images are common in the environment and, therefore, not visually distinctive in a global sense.



(a) 0.84 / 0.79 (b) 0.82 / 0.78 (c) 0.49 / 0.67 (d) 0.29 / 0.67 (e) 0.13 / 0.54

(f) 0.85 / 0.78 (g) 0.79 / 0.77 (h) 0.60 / 0.73 (i) 0.11 / 0.58 (j) 0.02 / 0.58

$S_{G_i} \in [0, 1]$:

$$S_{G_i}(t) = \frac{\mathcal{G}_i(t)}{\mathcal{G}_{\max}}, \tag{3.6}$$

where the normalizer, $\mathcal{G}_{\max}$, is the maximum summed idf score encountered thus far.

Figure 3.7 shows an example of applying global saliency to categorize sample underwater and indoor office imagery. As can be seen, the global saliency score, $S_G$, fires on the visual rarity of vocabulary words occurring within the image, whereas the local saliency score, $S_L$, fires on vocabulary diversity only. For example, the two rightmost figure columns (i.e., (d),(e) and (i),(j)) show that global saliency can be low even for locally salient imagery. This is because several of the vocabulary words (e.g., bricks, weld lines) occur frequently throughout the environment—lowering their overall idf score.

## 3.3 Saliency-informed Visual SLAM

Now, we have two saliency metrics that we can exploit for better loop-closure detection, which is the most important and difficult problem in SLAM. This section will discuss the improvement that we achieve by using the two saliency metrics in SLAM, which we call saliency-informed SLAM. Using our previously defined local saliency measure, we can improve the performance of visual SLAM in two ways:

1. We can sparsify the pose-graph by retaining only visually salient keyframes;

2. We can make link proposals within the graph more efficient and robust by combining visual saliency with measures of geometric information gain.

In the first step, we can decide whether or not a keyframe should be added at all to the graph by evaluating its local saliency level—this allows us to cull visually homogeneous imagery, which results in a graph that is more sparse and visually informative. This improves the overall efficiency of graph inference and eliminates nodes that would otherwise have low utility in underwater visual perception.

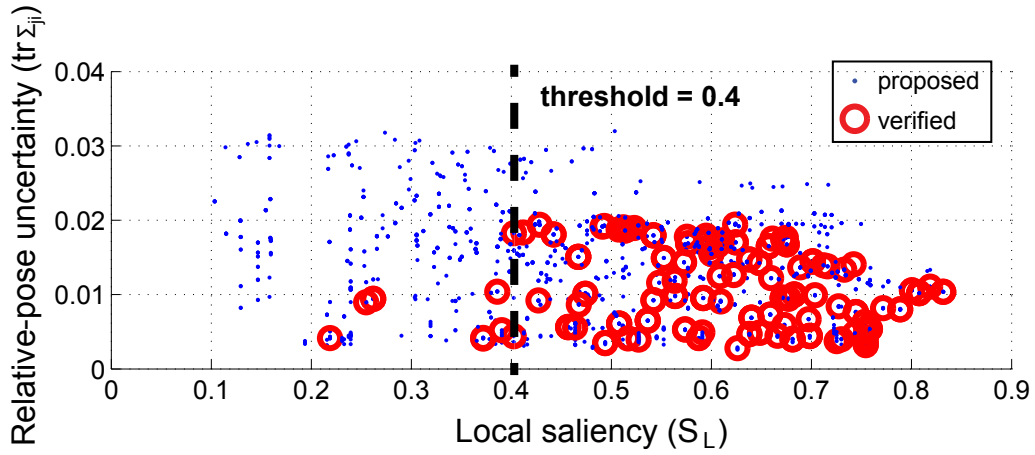
In the second step, we can improve the efficiency of link proposal by making it "salient-aware". For efficient link proposal, Ila et al. (2010) used expected information gain to prioritize which edges to add to the graph—thereby retaining only informative links. However, when considering the case of visual perception, not all camera-derived measurements are equally obtainable. Pairwise registration of low local saliency images will fail unless there is a strong prior to guide the putative correspondence search (e.g., Figure 2.5(c) top), whereas pairwise registration of highly salient image pairs often succeeds even with a weak or uninformative prior (e.g., Figure 2.5(c) bottom). Hence, when evaluating the expected information gain of proposed links, we should take into account their visual saliency, as this is an overall good indicator of whether or not the expected information gain (i.e., image registration) is actually obtainable. By doing so, we can propose the addition of links that are not only geometrically informative, but also visually plausible.

### 3.3.1 Salient Keyframe Selection

During SLAM exploration, image saliency can be used to pre-evaluate whether or not it would be beneficial to add a keyframe to the graph. Naively adding nodes to the graph can introduce a large number of meaningless variables, thereby making SLAM inference computationally expensive. Related to this issue, recent studies have focused on efficient graph building in the name of scalable and lifelong SLAM. To be able to localize and map a large area for a long period of time, an algorithm should manage the number of nodes and measurements that grow over time. Specifically, to enable lifelong mapping, nodes and measurements need to be selectively pruned from the graph in an efficient way. Grisetti et al. (2010) solved this problem at the back-end by enabling coarse structure of the scene using a hierarchy. Another strategy by Kretzschmar et al. (2010) is to prune out less useful nodes and observations by measuring the expected information gain.

When we have a measure of usefulness of the node, we can intelligently choose which set of nodes to include in the graph—only adding keyframes with high local saliency. For

**Figure 3.8** Local saliency versus navigation prior uncertainty. Local saliency of image pairs that resulted in successful pairwise image registration for the *R/V Oceanus* dataset. (a) A scatter plot of relative-pose uncertainty versus local saliency for candidate image pairs satisfying a minimum overlap criteria. Blue dots represent all attempted pairs whereas red circles indicate those that were successfully registered. (b) Tabulated data showing what fraction of failed registrations are pruned and what fraction of successful registrations are retained, when thresholding on different values for the minimum local saliency threshold, $S_L^{\text{min}}$. For example, by using a threshold of $S_L^{\text{min}}$=0.4, we retain 95% of successful registrations, yet are able to prune 32% of failed matches.




(a) Scatter plot of relative-pose uncertainty versus saliency

| $S_L^{\text{min}}$ | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 |
|---|---|---|---|---|---|---|---|
| successful & retained [%] | 100 | 98 | 95 | 82 | 55 | 22 | 5 |
| failed & discarded [%] | 5 | 15 | 32 | 51 | 70 | 88 | 99 |

(b) Effect of thresholding on local saliency

this purpose, we use a minimum threshold on local saliency, $S_L^{\text{min}}$, as a criteria for adding keyframes to the graph. To determine this threshold, we examined the local saliency score of underwater image pairs that resulted in successful pairwise image registration, while simultaneously examining the relative-pose uncertainty associated with their pose-constrained correspondence search (PCCS) search prior. Figure 3.8 displays a scatter plot from this analysis using data from the *R/V Oceanus* dataset (a qualitative result will be provided in Figure 3.11). Plotted as dots are all attempted pairwise image registrations between nodes satisfying a minimum overlap criteria. Out of this set, those pairs that resulted in a successful pairwise image registration are circled. The results show a strong correlation between image registration success and local saliency. For those pairs that fall below a local saliency level of $S_L < 0.4$, we see only a small fraction result in registration success, and for those that do, they have a strong PCCS search prior (i.e., low relative-pose uncertainty). Hence, by discarding images with low local saliency, we see that we can eliminate a large fraction of the failed candidate pairs. In fact, the empirical evidence shows that

we can eliminate 30–68% of the failed attempts by using a minimum saliency threshold somewhere between $S_L^{\mathsf{min}}$=0.4–0.6.

### 3.3.2 Saliency Incorporated Link Hypothesis

One formal approach to hypothesizing link candidates is to examine the utility of future expected measurements—also known as information gain. For example, Ila et al. (2010) use a measure of information gain to add only informative links (i.e., measurements) to the SLAM pose-graph. Other example uses can be found in control (Sim and Roy, 2005; Vidal-Calleja et al., 2006; Bryson and Sukkarieh, 2005), where the control scheme evaluates the information gain of possible future measurements and leads the robot on trajectories that reduce the overall SLAM localization and map uncertainty.

Following Ila et al. (2010), we express the information gain of a measurement update between nodes $i$ and $j$ as

$$\mathcal{I} = H(X) - H(X|\mathbf{z}_{ij}), \tag{3.7}$$

where $H(X)$ and $H(X|\mathbf{z}_{ij})$ are the entropy before and after measurement, $\mathbf{z}_{ij}$, respectively. For a Gaussian distribution, Ila et al. (2010) showed that this calculation simplifies to

$$\mathcal{I} = \frac{1}{2} \ln \frac{|\mathrm{S}|}{|\mathrm{R}|}, \tag{3.8}$$

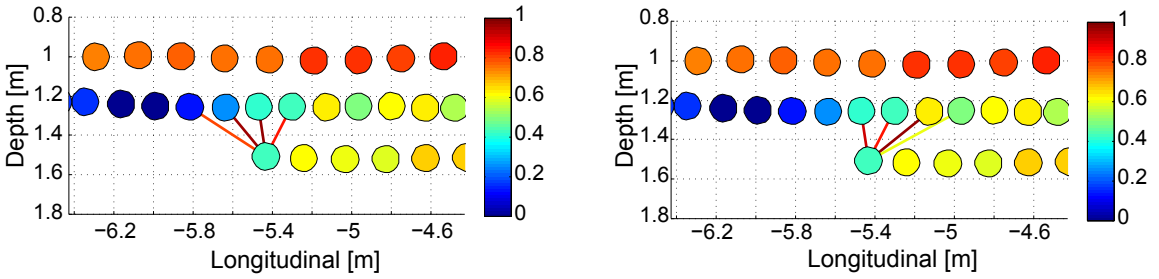
where $\mathrm{R}$ and $\mathrm{S}$ are the measurement and innovation covariance, respectively. In the case of our 5-DOF camera observation model (2.3), calculation of the innovation covariance becomes

$$\mathrm{S} = \mathrm{R} + \begin{bmatrix} \mathrm{H}_i & \mathrm{H}_j \end{bmatrix} \begin{bmatrix} \Sigma_{ii} & \Sigma_{ij} \\ \Sigma_{ji} & \Sigma_{jj} \end{bmatrix} \begin{bmatrix} \mathrm{H}_i & \mathrm{H}_j \end{bmatrix}^\top, \tag{3.9}$$

where $\mathrm{H}_i$ and $\mathrm{H}_j$ are the non-zero blocks of (2.3)'s Jacobian and $\left[\begin{smallmatrix} \Sigma_{ii} & \Sigma_{ij} \\ \Sigma_{ji} & \Sigma_{jj} \end{smallmatrix}\right]$ is the marginal joint covariance between nodes $i$ and $j$, which is efficiently recoverable within iSAM (Kaess and Dellaert, 2009). The utility of evaluating information gain (3.8) is that it can be used to assess which edges are the most informative to add to the pose-graph—before actually attempting image registration.

In the approach outlined above, an equal likelihood of measurement availability is assumed. In other words, (3.8) assesses the geometric value of adding the perceptual constraint *without regard to if, in fact, the constraint can be made.* As evident in our work, not all camera-derived constraints are equally obtainable, and these constraints are in fact largely influenced by the visual content within the scene. Candidate links with high information gain may not be the most plausible camera-derived links due to a lack of visual

**Figure 3.9** Illustration of link proposal using saliency. Sample result for link proposal using saliency incorporated information gain on the *R/V Oceanus*. In both cases, four links were proposed and the color of each link indicates the priority. (a) shows symmetric proposal while (b) is capable of weighting toward higher saliency nodes. In other words, given the same number of link proposals, (b) proposes more plausible links than (a).




(a) Link hypothesis using $\mathcal{I}$ (b) Link hypothesis using $\mathcal{I}_L$

saliency. We argue that the act of perception should play an equal role in determining candidate image pairs.

Based upon the local saliency metric developed earlier, and noting that $S_L \in [0, 1]$, we combine visual saliency with the expected information gain to arrive at a combined visual/geometric measure that accounts for perception:

$$\mathcal{I}_L = \begin{cases} \mathcal{I} \cdot S_L & \text{if } S_L \geq S_L^{\min} \text{ and } \mathcal{I} \geq \mathcal{I}^{\min} \\ 0 & \text{o.w.} \end{cases} \tag{3.10}$$

Strictly speaking, (3.10) is no longer a direct measure of information gain in the mutual information sense; however, it is a scaled version according to visual saliency. This allows us to prioritize candidate image pairs based upon their geometric informativeness as well as their visual registrability.

This combined approach results in better link hypotheses—forcing the link proposal scheme to lean toward visually salient nodes among those that are equally geometrically informative. Figure 3.9 depicts a sample result from the *R/V Oceanus* dataset. The color of a proposed link indicates how informative the link is (i.e., $\mathcal{I}$), while the color of a node represents how salient the imagery is (i.e., $S_L$). In the first case, only the geometry of the constraint is taken into account through the calculation of information gain. In the second case, the combined measure (3.10) guides the selection toward feature-rich image pairs, rather than processing visually uninformative images with high geometric gain. In doing so, it proposes realistically achievable camera-derived candidate links.

**Figure 3.10** Experimental setup for saliency evaluation. Underwater hull inspection experiments conducted using the Bluefin Robotics HAUV shown in (a). Three different ship hulls were surveyed: (b) *R/V Oceanus*, (c) *USCGC Venturous*, and (d) *SS Curtiss*.

(a) HAUV

(b) *R/V Oceanus*

(c) *USCGC Venturous*

(d) *SS Curtiss*

| | ***R/V Oceanus*** | ***USCGC Venturous*** | ***SS Curtiss*** |
|---|---|---|---|
| **Length** | 54 m | 64 m | 183 m |
| **Beam** | 10 m | 10 m | 27 m |
| **Draft** | 5.3 m | 3.0 m | 9.1 m |
| **Displacement** | 960 t | 759 t | 24,182 t |

(e) Vessel characteristics

## 3.4 Saliency Results

In this section, we first present the saliency map on different vessels, using the two saliency metrics defined in the previous section. We illustrate the performance of the two different saliency metrics using real-world data collected from a series of underwater ship hull inspection surveys using the HAUV platform (Vaganay et al., 2006). We surveyed three ship hulls: the Woods Hole Oceanographic Institution (WHOI) *R/V Oceanus*, the United States Coast Guard Cutter (USCGC) *Venturous* and the *SS Curtiss* as depicted in Figure 3.10.

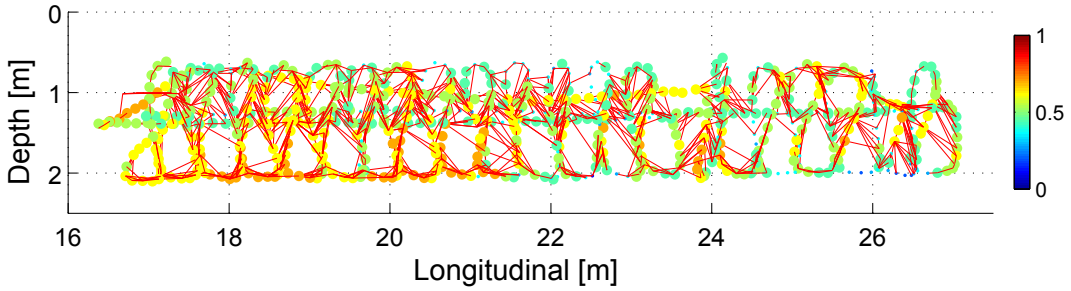
### 3.4.1 Local and Global Saliency Maps

To verify the performance of the two saliency metrics (i.e., local and global), their respective normalized saliency maps have been overlaid atop our pose-graph visual SLAM results.

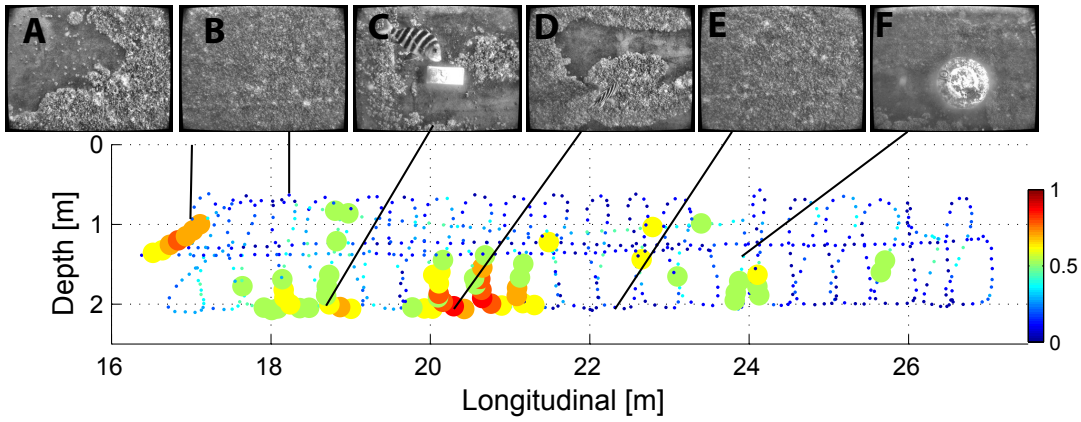
#### Local Saliency Maps

Overlaying the local saliency map on the SLAM result shows the coincidence of successful camera registrations and areas with a high local saliency score. To have a successful pairwise camera measurement, both images need to be locally salient (i.e., texture rich). Note that successful measurements (red lines in local maps) have been made only when both of the images have a high local saliency score. When either image lacks saliency,

**Figure 3.11** Local and global saliency maps on the *USCGC Venturous*. SLAM trajectory of the HAUV with successful cross-track camera registrations depicted as red edges. The normalized local saliency measure and the normalized global saliency measure, respectively, are depicted for the *USCGC Venturous*, and enlarged where they are larger than ($S_L/S_G > 0.4$). (a) Most of the scene is locally salient, and thus, the camera measurements and locally salient nodes are evenly distributed on the hull. (b) The surface of the vessel is populated with marine growth (B and E) that lowers its globally saliency score. Two artificial targets (C and F), and distinguished scene where there are no barnacles, are noted as rare areas on the hull.




(a) Local saliency map on *USCGC Venturous*




(b) Global saliency map on *USCGC Venturous*

image registration fails (i.e., regions with missing edges in the graph). Figure 3.1(c) depicted the result of applying our local saliency score to the *R/V Oceanus* dataset, where the density of successful cross-track links is clearly spatially correlated with the image feature content. Our normalized local saliency measure, $S_L$, which spans from 0 to 1, is overlaid on top of the SLAM graph and correlates well with camera link density. These successful camera measurements typically correspond to nodes with a local saliency score of 0.4 or greater. A counter example is image C, which has low local saliency ($S_L$=0.18), but which nonetheless was successfully registered due to a strong PCCS SLAM prior (same image as Figure 2.5(c) top). This agreement between the local saliency and the successful camera measurements is further verified from different deployments on two different

vessels—Figure 3.11(a) and Figure 3.12(c).

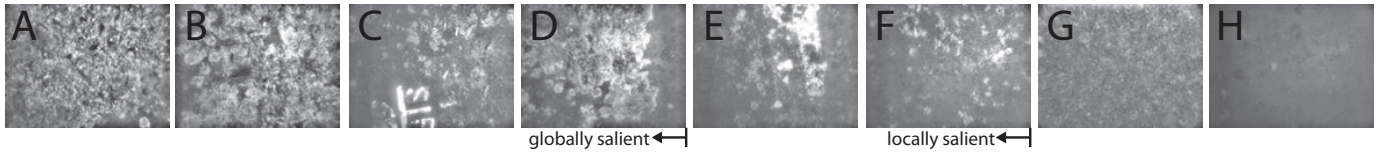
For the *USCGC Venturous* mission, the images of the vessel were abundant with features throughout the survey, as evident from the top figure in Figure 3.11(a), which shows evenly distributed camera registration. In contrast, a mission on *SS Curtiss* shows a segregated local saliency map. The hull of the *Curtiss* was feature-less on the side, but covered by marine growth rich in texture at the bottom (Figure 3.12(c)). In particular, note how $S_L$ correlates spatially with where successful camera-edges occurred in the exhaustive SLAM graph. The bottom of the hull had a high concentration of marine growth (e.g., $A$ to $F$ in Figure 3.12(a)), making it visually feature-rich for pairwise image registration; this is where the majority of cross-track image registrations occurred. The vertical side of the hull was relatively clean and, thus, feature-empty (e.g., $G$ and $H$ in Figure 3.12(a)), so relatively few pairwise registrations occurred in those regions. As in the other results, the local saliency map $S_L$ predicts well where underwater camera registrations actually happened.

### Global Saliency Maps

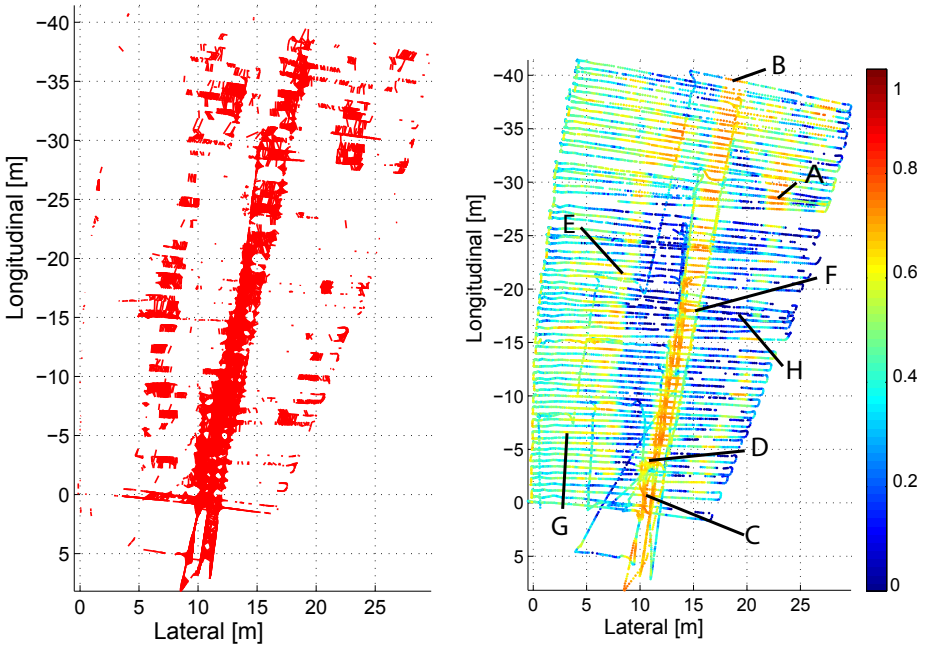
Unlike the local saliency metric, the global saliency metric reacts to rare features. As a validation, three global saliency maps on the *R/V Oceanus* (Figure 3.1(d)), *USCGC Venturous* (Figure 3.11(b)), and *SS Curtiss* (Figure 3.12(d)) are shown overlaid on the SLAM results. This global saliency score, $S_G$, which also spans from 0 to 1, is overlaid on top of the SLAM trajectory and nodes with $S_G > 0.4$ are enlarged for easier visualization. Note that global saliency can be used to identify visually distinct (i.e., rare) scenes with respect to the rest of the hull. These visually distinctive regions, for example, could serve as useful locations for guiding where the robot should revisit for attempting loop-closure. For example, in the *USCGC Venturous* (Figure 3.11(b)) survey, its hull was covered with barnacles in most regions (B and E in Figure 3.11(b)), except for two locations where artificial targets (inert mines) were attached to the hull. High global saliency is reported at these two target positions (C and F in Figure 3.11(b)) since they are rare. Also, other visually uncommon scenes (A and E in Figure 3.11(b)) scored high.

Next, we will show an implementation of saliency-informed SLAM and compare its result with the saliency-ignored SLAM case. For this saliency-informed SLAM, only local saliency is used to improve SLAM performance—global saliency will be reintroduced in Chapter IV in our development of PDN.

**Figure 3.12** Local and global saliency maps on the *SS Curtiss*. (a) Sample images of *SS Curtiss*. (b) A top-down view of the pose-graph depicting where the successful pairwise camera-derived edges occur. (c) A top-down view of the pose-graph with our local saliency metric overlaid. (d) A top-down view of the pose-graph with our global saliency metric, $S_G$, overlaid. The node size has been scaled by its saliency level for visual clarity.



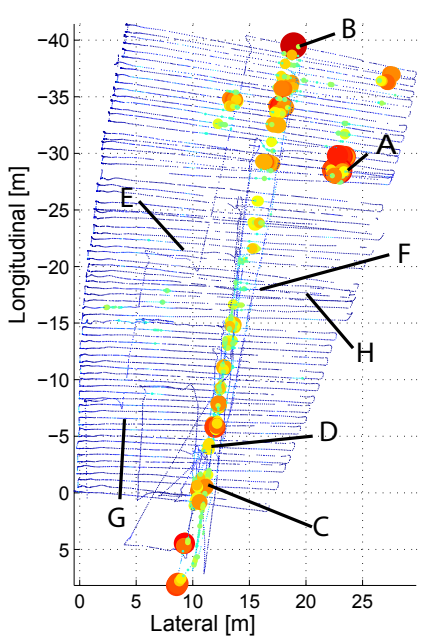

(a) Sample images




(b) Camera constraints




(c) Local saliency $S_L$




(d) Global saliency $S_G$

### 3.4.2 Saliency-informed SLAM

Here, we report experimental results that evaluate our real-time visual SLAM algorithm. The presented dataset is from a February 2011 survey of the *SS Curtiss* using the HAUV. The hull survey mission consisted of vertical track-lines, extending from the waterline to the keel, spaced approximately $0.5\ \mathrm{m}$ apart laterally. The survey started near the bow and continued toward the stern while maintaining a vehicle standoff distance of approximately $1\ \mathrm{m}$ from the hull at all times using Doppler velocity log (DVL) measured range. This configuration resulted in approximately 30% cross-track image overlap for a $\sim 45^{\circ}$ horizontal camera field of view (in water). Occasionally the vehicle was commanded to swim back toward the bow, orthogonal to its nominal track-line trajectory, to obtain image data useful for time-elapsed loop-closure constraints. The total survey area comprised a swath of approximately $45\ \mathrm{m}$ along-hull by $25\ \mathrm{m}$ up-down hull for a total path length of $2.7\ \mathrm{km}$ and a 3.4 hour mission duration. The camera was operated in underwater mode (Figure 3.10(a)) at a fixed sample rate of $2\ \mathrm{Hz}$, which resulted in a dataset of 24,773 source images. The dataset was logged using the Lightweight Communications and Marshalling (LCM) publish/subscribe software framework (Huang et al., 2010), which supports a real-time playback capability useful for post-mission software development and benchmark analysis. The results presented here are for post-process real-time playback using the visual SLAM algorithm implementation, as described in this paper.

#### 3.4.2.1 Saliency-Ignored SLAM Baseline Results

For this experiment, we ran the visual SLAM algorithm in a "perceptually naive" mode to benchmark its performance in the absence of saliency-based keyframe selection and saliency-incorporated link hypothesis (described earlier in §3.3.2). For this test we added keyframes at a fixed spatial sample rate resulting in approximately 70% sequential image overlap, and used geometric information gain only (i.e., not saliency incorporated) for link hypothesis. We ran with three different levels of link hypothesis: $n_{plink}$=3, $n_{plink}$=10, and $n_{plink}$=30, where $n_{plink}$ represents the maximum number of proposed hypotheses per node. We increased the number of hypotheses per node to 30 (which is 10 times more than the 3 links per node used in the saliency incorporated case to be discussed in the next section) so that the number of successfully registered camera links in the pose-graph can be used as a baseline for comparison. We refer to this baseline result as "the exhaustive SLAM graph", as all nominal nodes were added and all geometrically informative links where tried. The resulting 3D trajectory is depicted in Figure 3.13(a). This baseline result contains 17,207 camera nodes, 29,426 5-DOF camera constraints, and it required a cumulative processing

time of 10.70 hours (this includes image registration and iSAM inference). Figure 3.12(b) shows a top-down view of the registered pairwise camera-constraint edges and where they spatially occurred in the 3D pose-graph.

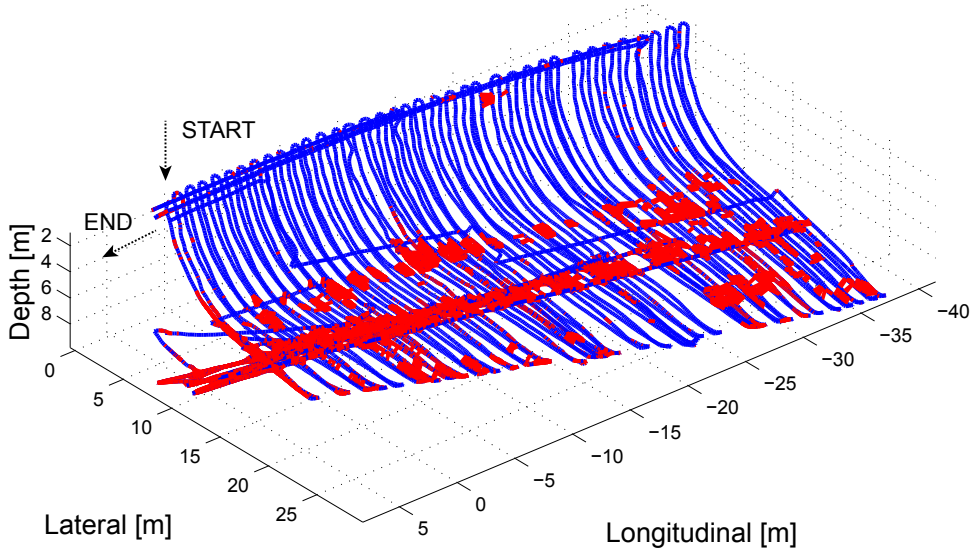
We have noted that the local saliency level could be a criteria to improve the link hypothesis. To quantitatively evaluate the utility of local saliency for discriminating successful camera registration, Figure 3.13(b) depicts a scatter plot showing the occurrence of all link hypotheses attempted by the exhaustive SLAM result, as plotted in local saliency space. Each dot in the plot represents an attempted link registration between camera nodes $\mathbf{x}_i$ (candidate node) and $\mathbf{x}_j$ (current node), while each circle represents those pairs that resulted in success. Each axis in the graph represents the local saliency level $(S_{L_i}, S_{L_j})$ between the image pair. The plot shows a positively correlated distribution between proposed $(S_{L_i}, S_{L_j})$ pairs. This is in fact encouraging, and indicates that the local saliency metric is consistent—meaning that spatially neighboring image pairs tend to score the same in saliency. Also noticeable in the graph is that the distribution of registered links (circles) is concentrated in the top-right corner, where both nodes have a high saliency score, whereas there is no such concentration on proposed links. This distribution reveals that a large number of non-visually-plausible links could in fact be pruned from the SLAM process by incorporating local saliency into the keyframe and link hypothesis selection phases.

#### 3.4.2.2 Saliency-Informed SLAM Result

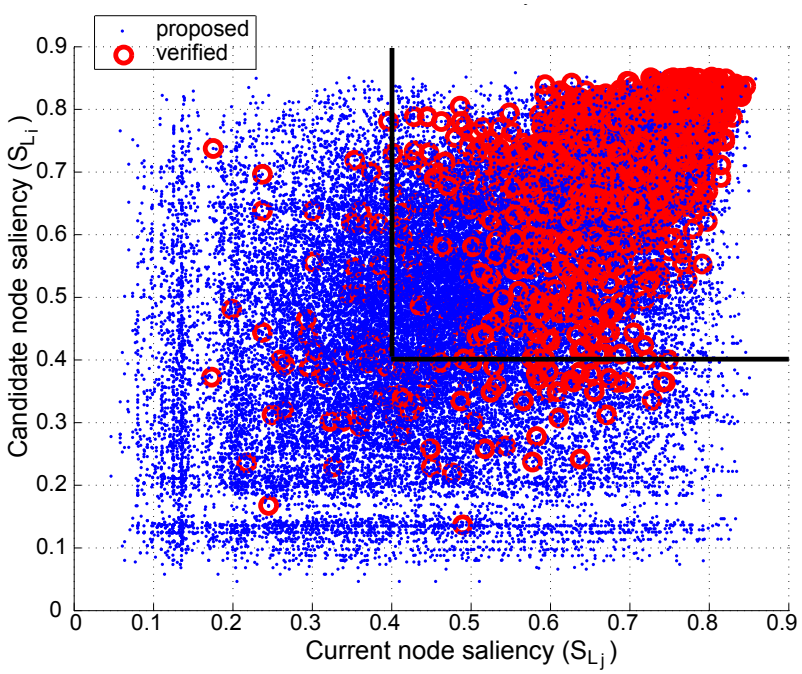
For this experiment, we re-ran the visual SLAM algorithm, but this time with the saliency-based keyframe selection and saliency informed information gain link hypothesis enabled. Based on our earlier tests with the *R/V Oceanus* (Figure 3.8), we used a saliency threshold of $S_L^{\text{min}}$=0.4 (black bounding box depicted in Figure 3.13(b)). The final resulting SLAM trajectory is depicted in Figure 3.14. Using the saliency-based front-end, we reduced the total number of keyframes from 17,207 (in the exhaustive set), to only 8,728—a 49.3% reduction by culling visually uninformative nodes from the graph. Table 3.1 and Table 3.2 summarize the overall computational efficiency improvement.

The saliency informed SLAM graph consists of 8,728 nodes and used $n_{plink}$=3 for link hypotheses per node. The cumulative iSAM inference time in this case is $0.52$ hours, and when accounting for image processing time, the entire SLAM result can be computed in less than $1.31$ hours, which is 2.6 times faster than the actual mission duration time of $3.4$ hours (Table 3.1). The time elevation graph (Figure 3.14(b)) makes it easier to visualize the elapsed duration between loop-closure camera measurements. In the top-down view (Figure 3.14(c)), the images on the right depict the keyframes and registered loop-closure event, verifying the overall consistency of the metric SLAM solution. The

**Figure 3.13** Exhaustive SLAM for the *SS Curtiss*. (a) Baseline exhaustive SLAM result. (b) Scatter plot depicting all attempted pairwise image hypotheses for the exhaustive SLAM result as viewed in saliency space. Each dot represents a single link hypothesis and indicates the $(S_{L_i}, S_{L_j})$ local saliency value for the image pair; successfully registered image pairs are circled. Note the strong positive correlation that exists between successfully registered pairs and their local saliency values. For reference, hypotheses that would be eliminated by a local saliency threshold of $S_L^{\min}$=0.4 lie outside the drawn box.

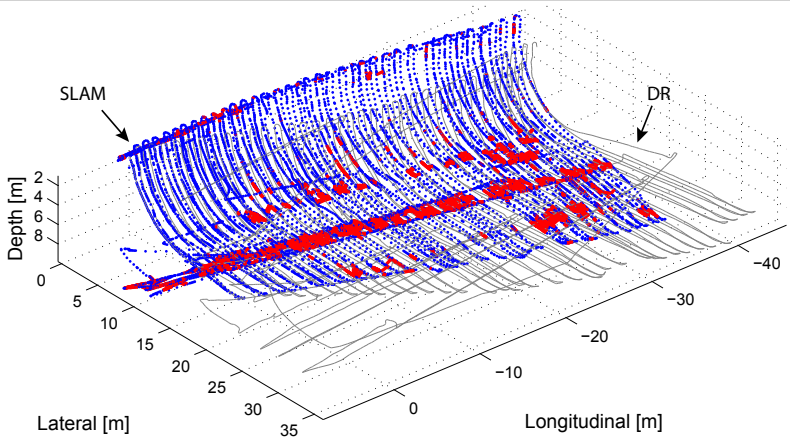



(a) Exhaustive SLAM result with all nodes ($n_{plink} = 30$)

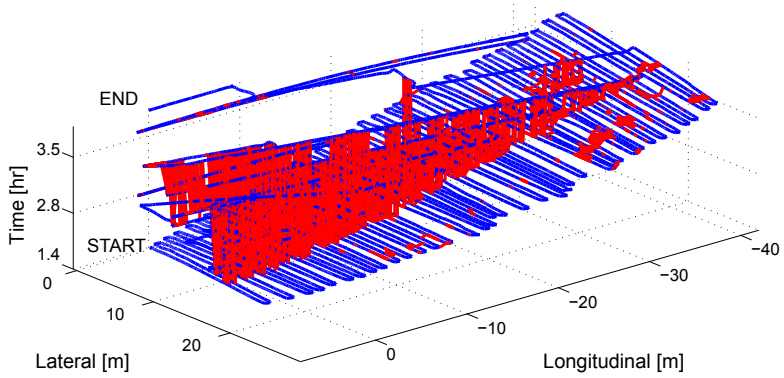



(b) Scatter plot

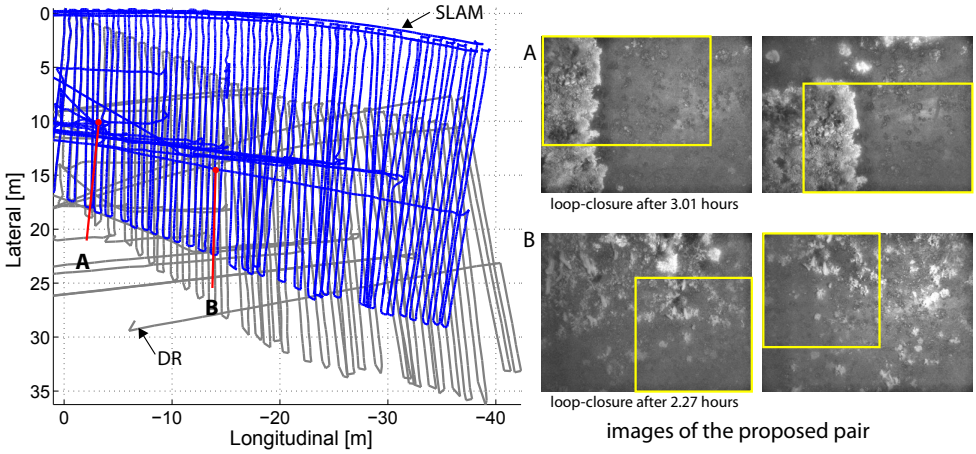
**Figure 3.14** Saliency-informed SLAM result for the *SS Curtiss*. (a) The blue dotted trajectory represents the iSAM estimate with camera constraints depicted as red edges, while the gray trajectory represents dead-reckoned (DR) trajectory. (b) The $xy$ component of the SLAM trajectory estimate is plotted versus time, where the vertical axis represents mission time. (c) A top-down view of the SLAM estimate versus DR. The positions marked $A$ and $B$ are two examples of where large loop-closure events take place.




(a) Visual SLAM result with only salient nodes




(b) Time elevation graph depicting loop-closure measurements




(c) Top-down view of SLAM vs. DVL dead-reckoning graph

**Table 3.1** Improvement summary of using saliency-informed SLAM. Compared to the saliency-ignored case, the total number of camera nodes has been reduced by half enabling real-time performance.

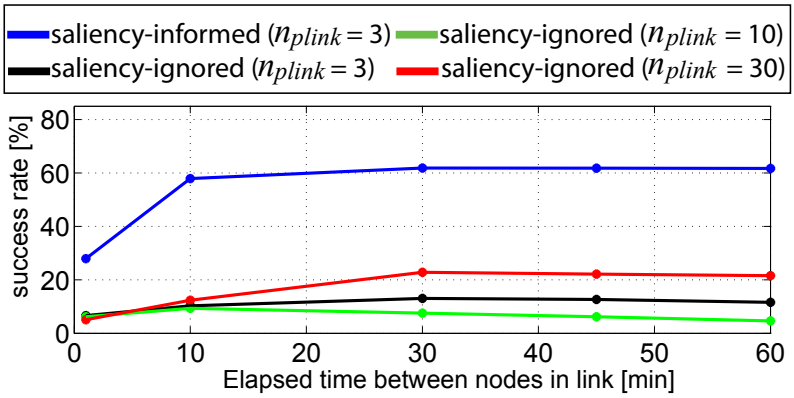
| No. of | Saliency-ignored | Saliency-informed | Fraction |
|---|---|---|---|
| Camera nodes | 17,207 | 8,728 | 50.7% |
| Hypoth. per node | 30 | 3 | 10.0% |
| iSAM CPU time | 8.70 hr | 0.52 hr | 6.0% |
| Total CPU time | 10.70 hr | 1.31 hr | 12.2% |

yellow box indicates the overlap between the two loop-closure images. The maximum difference between saliency incorporated and exhaustive SLAM is $1.10$ meters, whereas the DR trajectory shows significantly larger error due to the navigation drift. Two other saliency-ignored SLAM results also show large error throughout the mission.

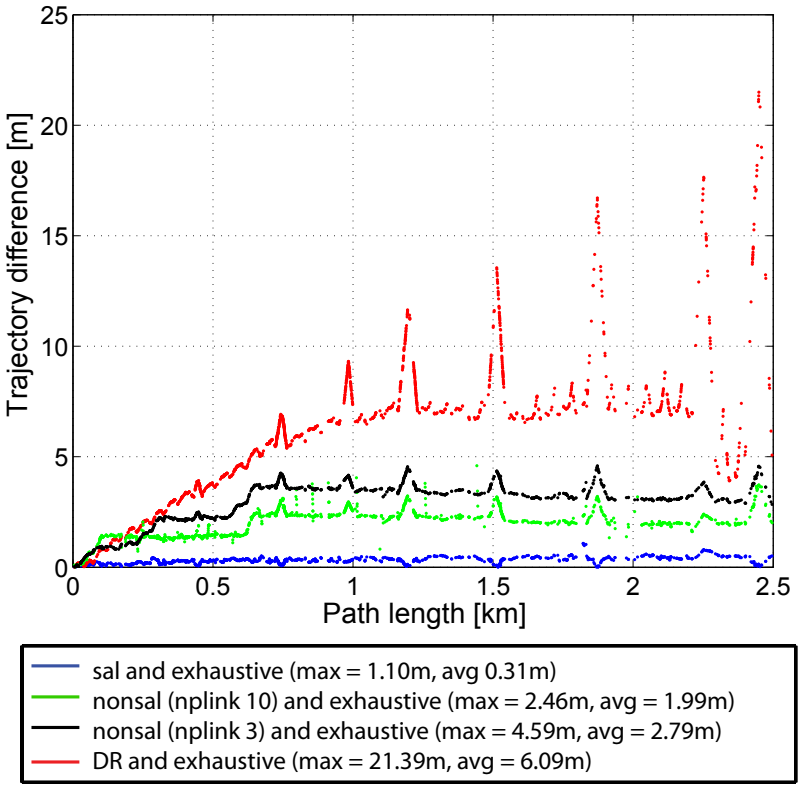
In terms of saliency's affect on SLAM performance, we note that even with far fewer nodes in the graph, we were still able to achieve almost the same total number of camera measurements in the graph. Using only half (50.7%) of the exhaustive graph nodes and a significantly smaller number of link proposals (3.4%), we achieved important cross-track camera measurements (Table 3.2). For easier loop-closure visualization, Figure 3.14(b) shows a time elevation graph of camera registration constraints—here the vertical axis indicates elapsed mission time. Camera measurements with large time differences indicate loop-closure events—for example, the SLAM estimate was accurate enough to register image pairs with over two hours of elapsed time difference (Figure 3.14(c)). As Figure 3.14(a) and Figure 3.14(c) show, this is a significant improvement over the dead-reckoned odometry result. While saliency ignored SLAM also shows reduced error over DR, saliency informed SLAM substantially outperforms it by resulting in more (112.4%) verified links and, thus, less error (15%) relative to the baseline exhaustive SLAM result, even with a smaller number of link proposals ($n_{plink} = 3$). This is because the saliency-ignored SLAM result failed to obtain critical, cross-track camera measurements that the saliency-informed SLAM successfully achieved.

Figure 3.15(a) shows the success rate of proposals for hypothesized pairs versus elapsed time. Links with a large time difference correspond to loop-closures after a large time period, and thus are of major interest. As can be seen in Table 3.2, the saliency-informed link proposal achieves a higher rate of success than the other two cases because we propose plausible links in saliency-informed SLAM. The fraction of the number of links of saliency-ignored case to the saliency-informed is given in parentheses. As our focus is on links with a large time differences, (i.e., loop-closure after a large time period), the number of verified links in the saliency-informed SLAM case exceeds the saliency-ignored cases.

**Figure 3.15** Statistics on saliency-informed SLAM for the *SS Curtiss*. Comparison between saliency-informed and saliency-ignored SLAM. (a) Plot comparing the success rate for saliency-informed (red), saliency-ignored (green) and exhaustive (blue). (b) A plot of the differences between the different trajectory estimates relative to the baseline exhaustive SLAM result.




(a) Success rate ($R_{succ}$) plot




(b) Difference in trajectories

**Table 3.2** Link proposal statistics for saliency-informed SLAM. Presented are the number of links and success rate ($R_{succ}$) statistics. The success rate is computed from the ratio of successful registration to the proposed links.

| | | Exhaustive $n_{plink} = 30$ | | w/o Saliency $n_{plink} = 10$ | | w/o Saliency $n_{plink} = 3$ | | w Saliency $n_{plink} = 3$ |
|---|---|---|---|---|---|---|---|---|
| $\Delta t$ | | No. of | Frac. | No. of | Frac. | No. of | Frac. | No. of |
| 1 min | No. of proposed link | 457,165 | (3.4%) | 124,653 | (12.5%) | 12,524 | (124.2%) | 15,553 |
| | No. of verified link | 23,125 | (18.8%) | 7,772 | (56.3%) | 829 | (524.5%) | 4,348 |
| | Success rate ($R_{succ}$) | 3.6% | (472.2%) | 1.9% | (894.7%) | 6.6% | (257.6%) | 17.0% |
| 10 min | No. of proposed link | 133,282 | (3.4%) | 25,182 | (18.1%) | 2,848 | (160.3%) | 4,565 |
| | No. of verified link | 16,476 | (16.0%) | 2,353 | (112.4%) | 293 | (902.4%) | 2,644 |
| | Success rate ($R_{succ}$) | 12.4% | (468.6%) | 9.3% | (620.1%) | 10.2% | (567.6%) | 57.9% |
| 1 hour | No. of proposed link | 38,701 | (6.2%) | 11,300 | (21.2%) | 1,001 | (239.4%) | 2,397 |
| | No. of verified link | 8,348 | (17.7%) | 527 | (280.6%) | 116 | (1275.0%) | 1,479 |
| | Success rate ($R_{succ}$) | 21.5% | (286.1%) | 4.6% | (1324.0%) | 11.6% | (531.9%) | 61.7% |

## 3.5 Conclusion

This chapter reported on a real-time 6-DOF visual SLAM algorithm and its experimental validation for autonomous underwater ship hull inspection. Two novel image saliency measures were introduced: local and global. Local saliency was shown to provide a normalized measure of intra-image feature diversity, while global saliency was shown to provide a normalized measure of inter-image rarity. We showed how local saliency can be used to guide keyframe selection, as well as how it can be combined with geometric information gain to propose visually plausible links. The overall applicability and utility of saliency for underwater visual SLAM was demonstrated through its application to three distinct hull datasets. Chapter IV will investigate the use of saliency in the area of active SLAM, in particular, the use of global saliency for loop-closure path-planning.

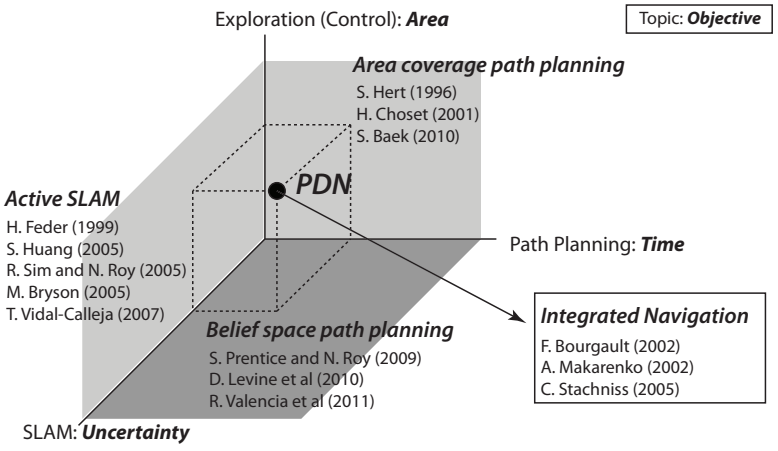
# CHAPTER IV

# Perception-driven Navigation

To enable robotic autonomous navigation over an area of interest, a robot needs to explore and map the area, while localizing itself accurately on the map that it builds. This autonomous navigation capability involves three topics, namely simultaneous localization and mapping (SLAM), path planning, and control. In the previous chapters, the major focus was on investigating a conventional approach to SLAM, whereby SLAM is treated as a passive process relative to robot planning and control. This passive form of SLAM is usually performed on preplanned or human-controlled trajectories; however, as we have seen, the robot trajectory strongly affects SLAM's performance. This interwoven nature of the navigation problem motivates our research toward an active SLAM approach, which is the main topic of this chapter. A fully autonomous agent needs the ability to plan a motion given a high-level command, for example, a task-level command from a human supervisor. In this instance, the robot should preplan and then modify the plan accordingly to accomplish the given task, and should not require detailed input by a human supervisor. Therefore, planning should be integral to robot navigation and should be considered concurrently with the SLAM problem. To achieve this integrated navigation framework, this chapter presents a decision theoretic algorithm that solves the SLAM and path planning problems concurrently.

In particular, this chapter considers the task of area coverage (i.e., to cover a certain area of interest) under the constraint of bounded navigation error. Specifically, our area coverage objective seeks a balanced control between exploration and revisiting in order to achieve better SLAM performance. Without loop-closure, SLAM will inevitably accumulate navigation drift over time; thus, we need to revisit portions of the map to bound error growth. At the same time, we need to pursue exploration, which is a competing objective that requires mapping the entire area in a reasonable time. Furthermore, and more importantly, SLAM, path planning, and control are interwoven and, thus, inseparable problems. For example, imprecise SLAM results affect the accuracy of the area coverage and, thus,

**Figure 4.1** Related work to perception-driven navigation (PDN). Representative prior works in the areas of SLAM, exploration, and path planning. Works are visualized along each axis in their relation to PDN.




the planning accuracy, while a mis-planned trajectory deteriorates SLAM performance. In this chapter, we introduce the idea of perception-driven control, a mathematical framework that seeks to balance the competing objectives between SLAM, control, and exploration for the autonomous robotic area coverage problem.

## 4.1 Related Work

Figure 4.1 visualizes the space of representative literature in the three topic areas of SLAM, path planning, and exploration[1]. Each axis in Figure 4.1 indicates a topic in autonomous navigation and associated objective. Exploration seeks to find a set of control actions to cover an unexplored area; SLAM solves for the map and robot position with a goal of bounded navigation uncertainty; and path planning performs at a higher level to complete the given tasks in a timely manner. There have been previous studies that investigated different aspects of this interwoven navigation problem and presented integrated results, namely (*i*) active SLAM (between SLAM and exploration) (Feder et al., 1999; Huang et al., 2005; Sim and Roy, 2005; Bryson and Sukkarieh, 2005; Davison et al., 2007), (*ii*) area coverage path planning (between exploration and path planning) (Hert et al., 1996; Choset, 2001; Baek et al., 2011), and (*iii*) belief space path planning (between SLAM and path planning) (Prentice and Roy, 2009; Levine, 2010; Valencia et al., 2011). Furthermore, Bourgault et al. (2002), Makarenko et al. (2002), and Stachniss et al. (2005) have looked into the problem of integrating these three areas in the name of integrated navigation. All

[1]This thesis follows the definition of Thrun et al. (2005) in defining exploration as a control to maximize knowledge of the external world.

of these approaches solve for an optimal solution (e.g., path or control)—each defining optimality differently, but typically as a function of area, time, or uncertainty level. This section will briefly summarize studies from each topic and their integrated approaches.

**Area-Coverage Path Planning:** Although the basic path planning algorithms often consider the problem of point to point path planning with obstacle avoidance given a map (i.e., find a shortest path between a start position and a goal position), area-coverage planning seeks to acquire such a map in the first place. This area-coverage problem has been addressed by Choset (2001), and is known as the *coverage path planning algorithm*, which is also closely related to robotic exploration and sensor deployment (Li and Cassandras, 2005; Batalin and Sukhatme, 2007). Many studies have tried to find an optimal solution to tackle this coverage problem in various applications, including a robot vacuum cleaner (Baek et al., 2011), robotic demining (Acar et al., 2003), and terrain coverage for AUVs (Hert et al., 1996). Optimality is defined in terms of the total amount of area covered with respect to the total time taken. To efficiently model the area, a grid/cell-based map representation is often adopted (e.g., an occupancy grid), so that optimality can be efficiently evaluated by the number of cells covered with respect to the total time taken. However, uncertainty in the localization and mapping phase is not considered; these previous studies focused on generating preplanned paths without considering localization or mapping error.

**Belief Space Path Planning:** There have been some efforts on merging SLAM and path planning into an integrated framework. The major difficulty in coupling path planning (whether it is deterministic or stochastic) and SLAM is that path planning assumes that a map is known a priori, while SLAM assumes that a path is given. Even stochastic path planning algorithms (LaValle and Kuffner, 1999; Kavraki et al., 1996; Kaelbling et al., 1995) start with the assumption of a known map. They focus on how to sample nodes from the area and then plan a path on them. Recently, some approaches have started to evaluate the resulting uncertainty in optimizing the path, such as the work of Prentice and Roy (2009) in Belief Roadmaps (BRMs), Valencia et al. (2011) on planning paths on SLAM constrained maps, or Levine (2010) on calculating possible information gain on a planned path using Rapidly-Exploring Random Trees (RRTs). Of these, the BRMs approach is closest to ours in that it considers the state's uncertainty when it is computing the objective function, though it is different from us since the nodes are sampled from a map that is not learned online during SLAM. In the work of Valencia et al., the authors perform SLAM first, then use the resulting pose-graph to plan a path to a goal position considering information gain through the graph.

However, and most importantly, exploration has not been considered in these previous studies. Their optimality was evaluated only by the uncertainty of the robot and the map,

not by time or area. This is because exploration was excluded in their evaluation, where the main focus was instead on point-to-point path planning. This thesis broadens the optimality definition to take into account area coverage together with SLAM by including the area coverage rate in the cost function (since our focus is on area covering navigation).

**Active SLAM:** The SLAM community has also made some efforts to add exploration functionality in SLAM, termed "active SLAM". Stemming from the seminal work of *active perception* by Bajcsy (1988), which pointed out that control can improve the quality of sensor data, these studies assume a preplanned path to follow and then undertake some variations that a robot can make for improvements. Active SLAM is an area in SLAM that tries to find the optimal trajectory that can improve map building and localization performance. Most active SLAM research consists of two parts: *(i)* defining a metric to be used as a measure of information gain and *(ii)* optimizing this measure to find control policies that maximize information gain. This line of research is found in the work of Feder et al. (1999), who used Fisher information (FI) as a metric in the objective function to construct an adaptive control action. More recently, Sim and Roy (2005) and Sim (2005) used FI to improve exploration, reporting the need to consider the path in localization and mapping. Their work pointed out the instability of the update step, which has been further extended to account for the control action by Davison et al. (2007). In this work, the authors considered a discrete set of actions to reduce state uncertainty. Similarly, in Bryson and Sukkarieh (2005), simulation results for unmanned aerial vehicles (UAVs) using a similar approach for on-line path planning was presented. Their work determined the proper action and strategies to improve the overall map quality on the basis of mutual information (MI), which relies on entropy and observability to provide the proper action sequence. Although this approach established a basis for combining the control architecture with SLAM, it only applied the optimal control input and did not globally solve for path planning. Frintrop and Jensfelt (2008) presented an active gaze control algorithm for SLAM by defining the usefulness of a landmark and tracking these useful landmarks. Aside from the active SLAM research, an optimal control strategy based approach has been introduced by Huang et al. (2005), where the authors optimized over the uncertainty of the last pose within a finite time window using a variant of model predictive control (MPC). In their work, they pointed out that the computational cost increases exponentially with the number of landmarks, the size of the map, and the size of the time window. In these studies from the SLAM community, optimality has been defined very similarly to the belief space path planning in that it has considered only the localization uncertainty in the cost function and not area coverage.

**Integrated Navigation:** Toward addressing PDN in a fully integrated approach, some studies have focused on this problem from a similar point of view to ours. This group of

research solves for exploration strategies considering both navigation and exploration performance. In earlier work on autonomous exploration, Whaite and Ferrie (1997) introduced a way to explore considering the uncertainty of the explored model. Although not directly addressing the SLAM problem, their work considered the reduction in model uncertainty through motion as a way to explore. Gonzalez-Banos and Latombe (2002) proposed exploration strategies analogous to the Next-Best-View (NBV) problem in computer vision (§1.1.3). First discussed by Connolly (1985), NBV seeks to find the best view of the scene that reveals the model details, and thus can be considered similar to the active exploration problem.

There are some studies that considered SLAM performance in the exploration phase. These integrated studies tried to search for an optimal solution to maximize area coverage and SLAM performance at the same time. Makarenko et al. (2002) presented an integrated exploration scheme based on mutual information. Similarly, Bourgault et al. (2002) considered map coverage and localization accuracy in order to generate an adaptive control action. Stachniss et al. (2005) pointed out the gist of this unsolved problem between SLAM and exploration. Their SLAM implementation compares two utilities associated with the action of exploration and revisit in order to determine whether to continue exploration or to revisit a previous location. This work is most similar to our approach, but in their work every measurement is considered to be equally likely in its availability, while in PDN it is not. Importantly, Stachniss et al. incorporated the cost of detours into the objective function in evaluating the robot and map uncertainty results. Recently, Kollar and Roy (2008) presented an exploration strategy using reinforcement learning. Because they assume a priori access to the ground truth, their algorithm is trained to learn the trajectory that maximizes the accuracy of the map (i.e., minimizes the error between the estimate and the ground truth). In their paper, the authors recommended the use of uncertainty in the absence of such true data. For a multi-robot case, Julian et al. (2012) suggested an exploration strategy using MI. Their work solves for an optimal SLAM control strategy by evaluating the gradient of MI.

As illustrated by these many studies, there is a gap between the areas of SLAM, exploration and path planning, which mainly comes from the fact that each approach has some assumptions on what priors are available at the initial phase. An attempt at categorizing the assumptions, objective functions and computational costs of previous studies are presented in Table 4.1. As seen our approach has similarities to previous work in integrated navigation, but differs by considering the measurement likelihood in the objective function.

An overarching assumption that this thesis makes is to start from very little prior information on the area of interest. The planning should consider the anticipated SLAM accu-

**Table 4.1** Summary of related works to PDN. The previous studies are summarized with respect to prior knowledge, objective function, consideration of measurement likelihood, and computational cost. In the calculation of time complexity, algorithms are compared assuming an $n$ element state vector. Area coverage planning (Hert et al., 1996; Choset, 2001; Baek et al., 2011) focuses on computing an optimal path offline where the memory complexity scales with the size of the map and the planning parameters, and the major operation and time complexity are not indicated in the table. In many studies, the EKF has been a popular choice for the SLAM back-end (Bourgault et al., 2002; Bryson and Sukkarieh, 2005; Davison et al., 2007). When the objective function includes MI-based term, computing MI requires calculation of the covariance matrix determinant. Even when using an EIF, Valencia et al. (2011) need to compute the inverse of the information matrix for MI evaluation prior to the planning phase. In the integrated navigation studies (Bourgault et al., 2002; Stachniss et al., 2005), which are most similar to PDN, computation cost appears in two terms, one related to the SLAM interference and the other related to the action path length.

| | | Prior | Objective function | | Measurement | | Major operation | Computational cost |
|---|---|---|---|---|---|---|---|---|
| | | | Uncertainty | Area | Likelihood | Type | | |
| **Area-Coverage Path Planning** | Hert et al. (1996) | map | no | yes | no | camera/sonar | – | – |
| | Choset (2001) | map | no | yes | no | general | – | – |
| | Baek et al. (2011) | map | no | yes | no | laser | – | – |
| **Active SLAM** | Feder et al. (1999) | no | yes | no | no | sonar | determinant of covariance matrix | $O(n^3)^a$ |
| | Sim and Roy (2005) | no | yes | no | no | range sensor | EKF update and $m$ candidate states | $O(n^2 \cdot m)^b$ |
| | Bryson and Sukkarieh (2005) | landmarks | yes | no | no | laser/vision | determinant of covariance matrix | $O(n^3)$ |
| | Davison et al. (2007) | no | yes | no | no | camera | determinant of covariance matrix | $O(n^3)$ |
| **Belief Space Path Planning** | Prentice and Roy (2009) | map | yes | no | no | beacon | EKF process and update for $k$ edges each of length $l$ | $O(kl)$ [c] |
| | Valencia et al. (2011) | SLAM map | yes | no | no | laser | inversion of information matrix | $O(n^3)$ [d] |
| **Integrated Navigation** | Stachniss et al. (2005) | no | yes | yes | no | laser | $N$ particles and action path length $l$ | $O(l \cdot N)$ |
| | Bourgault et al. (2002) | no | yes | yes | no | laser | determinant of covariance matrix | $O(n^3)$ |
| | PDN | no | yes | yes | yes | camera | covariance recovery cost $S(n)$ and action path length $l$ | $O(S(n) \cdot l)$ [e] |

[a] $n$ indicates state vector dimension.

[b] $m$ is the candidate state space where the path is planned. Since no nodes are used twice, $m$ will be decreasing.

[c] From pre-sampled nodes, the algorithm considers $k$ edges between nodes each of length $l$.

[d] This inversion happens once before planning. With pre-computed inversion, the online time complexity is $O(e \log^2 n)$ where $e$ is the number of edges. Note that if we were to use their planning scheme concurrently with SLAM, the inversion needs to be performed in every evaluation step.

[e] $S(n)$ is the cost for covariance recovery in reward computation, and will be discussed in §4.3.3.4.

racy in the planning phase, providing a nominal path to SLAM that will be detoured from accordingly. Starting from no prior knowledge on the environment, we perform SLAM online to build a map for localizing the robot in the area, while simultaneously planning paths to improve overall navigation subject to efficient area-coverage.

As indicated in Table 4.1, active SLAM only focuses on the robot uncertainty from SLAM, and area coverage planning only solves for the optimal coverage plan without accounting for the actual SLAM performance. Unlike these studies, which are constrained to only one aspect of the navigation problem, this thesis pursues a balanced strategy for both SLAM and area coverage in an integrated framework. Integrating SLAM and planning is also a focus of belief space planning, however, our approach solves for an area coverage problem and differs from the belief space planning in this regard, since belief space planning is typically only point-to-point. While integrated active exploration is most similar to our own approach, they impose an optimistic—and thus impractical in underwater navigation— assumption of obtaining all measurements predicted in the evaluation phase. Specifically, the novelty of our work is in consideration of the visual measurement likelihood within an integrated framework of SLAM and planning.

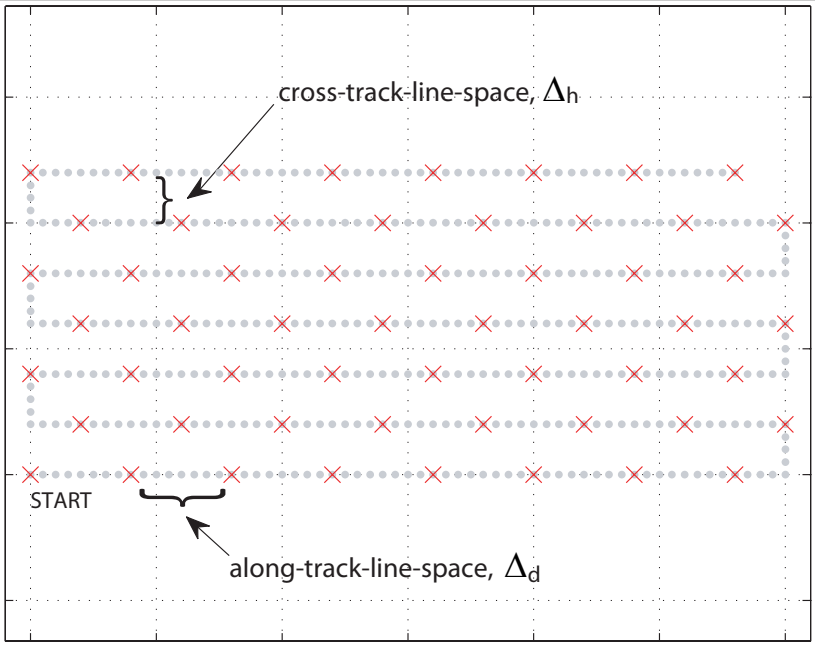
## 4.2 Motivation

The research question we address in this chapter is how to plan for 100% area-coverage while considering SLAM performance. We first introduce some typical methods used in conventional SLAM to design a survey, considering (*i*) sensor footprint coverage and/or (*ii*) the total information of the planned trajectory. Once generated, this nominal survey pattern is usually modified to include several revisit paths so that a robot can yield loop-closures by revisiting a previous location in the map. In conventional SLAM, these revisit motions are often preplanned or controlled by a human. This section introduces two typical nominal survey design methods and the motivation for PDN.

### 4.2.1 Conventional Preplanned Survey Design

To explore and map a target area, a prototype mission needs to be planned in advance of the survey. A typical survey geometry for area coverage is the regular grid pattern depicted in Figure 4.2. This pattern is determined by two design parameters, along-track-line spacing ($\Delta_d$) and cross-track-line spacing ($\Delta_h$). During the mission, a robot samples keyframes (red $\times$'s) along the trajectory and performs frame-to-frame comparison to establish relative constraints between nodes in the graph. The physical distance between nodes is induced by the frame acquisition rate and the speed of the robot, shown as the along-track-line distance

**Figure 4.2** A typical underwater grid pattern mission for area coverage. In this typical pattern, two design parameters, cross-track-line spacing, $\Delta_h$, and along-track-line spacing, $\Delta_d$, parameterize a survey. The sample keyframe nodes are marked with red $\times$'s along the trajectory.




in Figure 4.2. The other distance in the cross-track-line spacing, which is set based upon the sensor field of view (FOV) and desired overlap ratio. The choice of survey parameters (e.g., cross-track-line distance and along-track-line distance) are critical to the coverage area and desired navigation precision. We will briefly cover two possible preplanning methods.

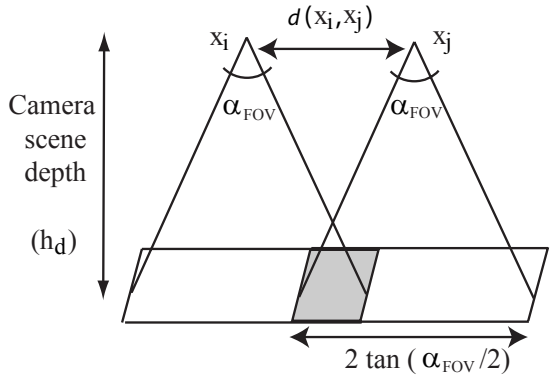
### Sensor Coverage-based Nominal Survey Design

One straightforward way to design this type of survey is by considering the sensor FOV and aiming for 100% sensor footprint coverage. In this approach, calculation of the survey pattern is based upon the desired percentage overlap in the sensor FOV (e.g., image overlap in the case of visual SLAM). This FOV is naively computed from the expected vehicle trajectory without taking into account the effect of navigation error. A typical selection of parameters would be 50-70% overlap along-track and 30% cross-track. With a fixed scene depth ($h_d$), known camera FOV ($\alpha_{\text{FOV}}$), and translational inter-node distance ($d(\mathbf{x}_i, \mathbf{x}_j)$), the inter-frame overlap ratio can be calculated as

$$O = \frac{2 \cdot h_d \cdot \tan(\alpha_{\text{FOV}}/2) - d(\mathbf{x}_i, \mathbf{x}_j)}{2 \cdot h_d \cdot \tan(\alpha_{\text{FOV}}/2)}. \tag{4.1}$$

For a survey at a constant speed ($v$) with a fixed sensor framerate ($fps$), the along-track-

**Figure 4.3** Calculation of camera sensor overlap ratio. Depiction of the overlap ratio computation using an idealized camera sensor FOV. The $\mathbf{x}_i$ and $\mathbf{x}_j$ represent two keyframe nodes with sensor overlap, and the scene depth, $h_d$, indicates the distance from $\mathbf{x}_i$ to the image plane. The overlap ratio is the portion of overlap area, shaded in gray, with respect to the sensor footprint.




line distance is specified as

$$d(\mathbf{x}_i, \mathbf{x}_j) = \Delta_d = \frac{v}{fps}. \tag{4.2}$$

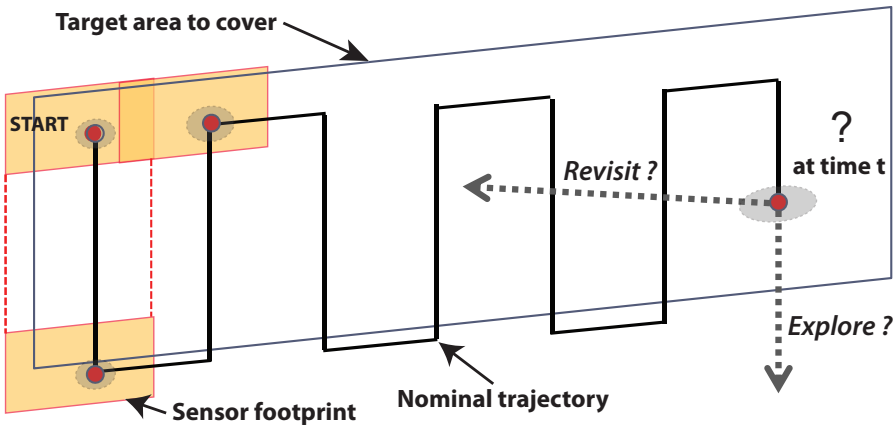
To achieve a desired along-track sensor overlap ratio, the vehicle speed and sensor frame rate need to be appropriately determined via (4.1) and (4.2). In the case of the desired cross-track sensor overlap ratio, an appropriate cross-track-line distance is $\Delta_h$, which can also be directly computed from (4.1).

### SLAM-considered Nominal Survey Design

Although this simple sensor footprint survey design allows for an easy way to preplan a survey, information theoretic approaches can improve the design by evaluating the effect of these parameters on SLAM's performance. One way to consider the trajectory's effect on SLAM in the preplanning phase is by using the Cramer Rao Lower Bound (CRLB). The CRLB is a conservative lower bound for an unbiased estimator, which herein is used to measure the preplanned survey localization uncertainty. For a camera-modality survey, we can precompute the CRLB for an assumed set of odometry and camera measurements (a detailed derivation is given in Appendix §B). Then, by examining the CRLB for the covariance matrix, the determinant of each covariance matrix sub-block reveals the uncertainty of each node in the graph.

For each set of survey design parameters, we can compute the expected performance from the CRLB in terms of the resulting graph uncertainty, together with the expected area coverage. To achieve an optimal combination of these parameters, we examine the relation of trajectory uncertainty and area coverage with respect to the design parameters to achieve an optimal balance given these competing considerations. A similar consideration can be

**Figure 4.4** Illustration of PDN. The nominal trajectory aims for 100% coverage over the given target area. The black line depicts the nominal trajectory that a robot follows, shown in orange is the sensor footprint. The red dot indicates a robot pose associated with a sensor measurement and its uncertainty ellipsoid is overlaid. PDN solves online for a proper control action at time $t$ to reduce the navigation error while maintaining survey efficiency, balancing between revisit and exploration.



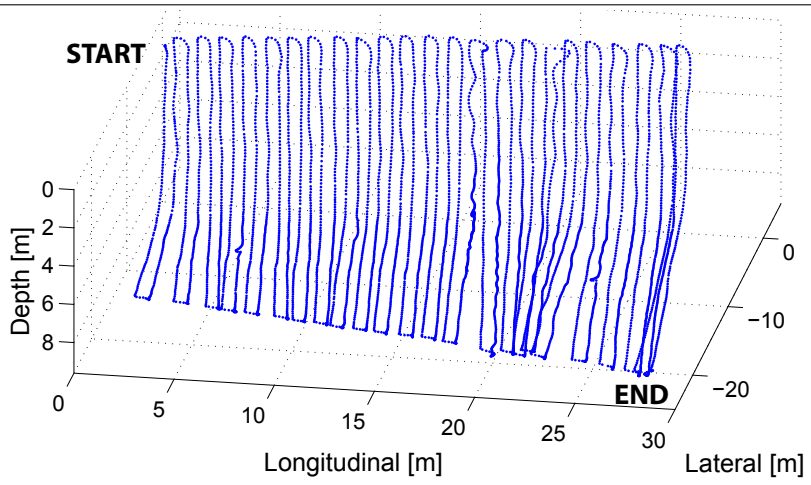

found by Hollinger et al. (2012), who presented a survey planning method for underwater missions considering the 3-D reconstruction uncertainty. Although localization uncertainty was not their focus, their work is similar to the CRLB approach as it considers the output trajectory uncertainty and shows an efficient preplanning and replanning scheme using non-parametric Bayesian regression.

### 4.2.2 Revisit Planning / Control for Loop-closures

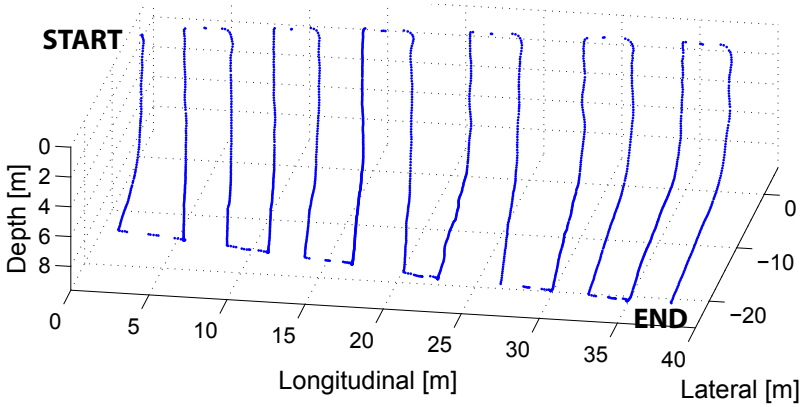
Given a preplanned trajectory (using the methods already discussed), a robot then carries out a survey of the target area following this preplanned trajectory. The nominal trajectory implicitly assumes that measurements are uniformly available everywhere spatially, and that by preserving a certain amount of sensor overlap, it will result in successful loop-closing constraints. However, the previously introduced methods preplan the nominal trajectory without knowing the actual distribution of features in the environment. This feature distribution significantly affects the ability of SLAM to perform successful loop-closure (as shown in Chapter §III), which may lead to few cross-track measurements in feature-poor regions.

To obtain large loop-closures, revisit actions are needed; similar to what humans do when they are lost while navigating. In other words, revisit actions are needed to reduce the navigation uncertainty by introducing loop-closures to previously visited scenes. As shown in Figure 4.4, the nominal trajectory is designed to guarantee 100% coverage over the given target area. However, when the uncertainty of the robot position increases during

**Figure 4.5** Two mission profiles in used in PDN's evaluation: camera and sonar. (a) A narrow-spaced camera-oriented mission has $0.5$ m track-line spacing for 70% cross-track overlap in camera footprint coverage. (b) A wide-spaced sonar-oriented mission has $4$ m track-line spacing for 100% sonar sensor coverage.




(a) Camera mission




(b) Sonar mission

execution of the mission, the robot needs to control itself for a revisit action. This revisit action is typically preplanned or controlled by a human pilot. However, using preplanned or human piloted revisit actions not only deteriorates autonomy, but they can also be ineffective and/or inefficient because the feature distribution is not known a priori. To tackle this problem, PDN aims to provide an intelligent and fully autonomous control scheme that balances between revisit and exploration, considering SLAM's navigational uncertainty and area coverage in a decision theoretic way.

In this chapter, two types of mission profiles are examined, and are largely distinguished by the type of primary sensor that they use. The two types of mission profiles are a camera-oriented mission and a sonar-oriented mission. The camera mission (Figure 4.5(a)) is a

narrow-spaced mission designed for 100% camera sensor footprint coverage. The spacing between track-lines is typically $0.5$ m, which yields about 70% cross-track overlap between images. The sonar mission (Figure 4.5(b)) is a wide-spaced mission designed for 100% sonar sensor footprint coverage. Since the sonar sensor footprint is 6 times larger than that of the camera, 100% sonar coverage can be achieved with $4$ m track spacing. Figure 4.5(b) shows a sonar mission with the larger track-line-spacing. In the sonar mission, no cross-track camera measurements are available due to the larger track-line-spacing, and camera loop-closures are only made by deliberate revisiting actions.

## 4.3 Perception-driven Navigation

PDN consists of three modules: clustering, planning, and reward evaluation (Figure 4.6), which will be presented in detail in the following subsections. While the normal SLAM process passively localizes itself and builds a map, PDN (*i*) clusters salient nodes into a set of candidate revisit waypoints (§4.3.1), (*ii*) plans a point-to-point path for each candidate revisit waypoint (§4.3.2), and (*iii*) computes a reward for revisiting each waypoint candidate (§4.3.3.4). The calculated reward measures the utility of revisiting that waypoint (i.e., loop-closure) versus continuing exploration for area coverage. By comparing the maximum reward for revisiting or exploring, the robot is able to choose the next best control step.

For the derivation of PDN, we start with the following assumptions and problem definitions:

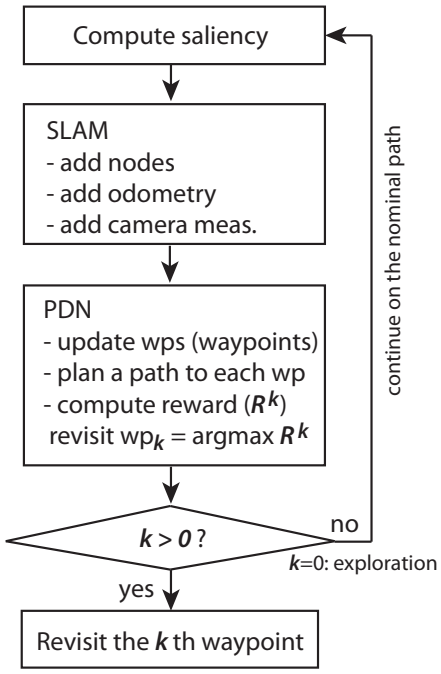
1. We assume an open underwater area without obstacles. The boundaries of the target coverage area are given, and the nominal mission profile is preplanned.

2. The target coverage area is designated. The robot starts with a certain size of area that it aims to cover with its primary sensor footprint. In this application, we assumed an open-area to cover without considering obstacles, selecting the grid pattern trajectory for the initial path[2].

3. The reference allowable navigation uncertainty is defined by the user, and will be used to trigger PDN's revisit decision. If the map and localization is certain enough, there is no point in revisiting for loop-closure. This uncertainty level is a user input, and also controls how conservative a user wants the SLAM performance to be.

[2]Although it is not the focus of this thesis, for the confined area with obstacles, a nominal path can be generated from a blueprint, and PDN applied with local control to avoid obstacles.

**Figure 4.6** Illustration of PDN's flow diagram. Provided with a SLAM pose-graph and saliency map, PDN updates a set of waypoints (§4.3.1), plans a path to these waypoints (§4.3.2), and computes rewards for each of the waypoints (§4.3.3.4). The reward $\mathcal{R}^k$ is computed for each waypoint $k$ where the reward from exploration $\mathcal{R}_{exp} = \mathcal{R}^0$ corresponds to the $0^{\text{th}}$ waypoint ($k = 0$). Lastly, either a revisiting or exploration action is executed to yield the maximum reward.




4. No prior information on the environment is provided except for the coverage boundaries. Planning and evaluation will be performed online on the pose-graph that the robot builds. No nodes or sample poses are given outside of SLAM.

Given the desired target area coverage and user defined allowable navigation uncertainty, PDN solves for an intelligent solution to the area coverage planning problem while considering SLAM's performance. In PDN's derivation, we address three issues. First, as our approach considers the resulting SLAM performance, the current robot uncertainty should play a role in the path planning. Thus, the current robot uncertainty is a control parameter that triggers the re-planning for better localization and mapping. Second, because the robot should complete the area coverage mission in a timely manner, the map uncertainty in terms of uncovered area needs to be considered. Finally, PDN needs to be computationally scalable for real-time, real-world performance. Since the complexity of the algorithm scales linearly with the number of revisit points, maintaining a feasible number of candidate revisit nodes is our focus.

### 4.3.1 Waypoint Generation

The complexity of PDN scales linearly with the number of revisit points $N_{wp}$ in the map, called *waypoints* since we need to evaluate the reward for each waypoint. To implement this in real-world applications, the algorithm should be computationally scalable (i.e., should be applicable to a large size map with a large number of nodes). Although all nodes in the pose-graph could be considered as waypoints, evaluating the outcome for all possible revisits is impractical. Moreover, due to the uneven spatial distribution of feature-rich areas in the environment, not all nodes are visually plausible for loop-closure. Therefore, PDN computes expected rewards for only a subset of candidate nodes selected for their visual saliency levels, resulting in only locally and globally salient nodes as waypoints.

This waypoint generation consists of two parts: salient node clustering and waypoint selection for each cluster. First, we threshold keyframes based upon their local saliency level (similar to $S_L^{\mathsf{min}}$ in §3.2.1) to generate a set of 3D points with strong local saliency. In PDN, a threshold ($S_L^{\mathsf{wp}}$) of $0.5$ is used to select a subset of highly salient nodes. Locally salient nodes represent texture-rich scenes, and, thus, identify feature-rich areas in the environment. The next step is to online cluster locally salient nodes into local neighborhoods, forming clusters. Finally, within each cluster, we select a representative waypoint node by considering both its visual uniqueness (i.e., global saliency level) and usefulness for loop-closure. This process allows us to compute the $N_{wp}$ best candidate waypoints for revisitation.

#### Online Clustering

For the first phase of waypoint generation, several online clustering algorithms were examined for their use in waypoint generation. K-means clustering (MacKay, 2003) is a popular algorithm for clustering data into a set of groups, but requires that the number of clusters be known in advance. As described in the fourth assumption, no information on the area is given a priori, thus the number of clusters is not known in advance. In that sense, k-means is not suitable for PDN and, therefore, PDN needs to find an alternative unsupervised clustering method.

One unsupervised learning algorithm, called mean shift clustering (Fukunaga and Hostetler, 1975; Cheng, 1995), provides a feasible solution to clustering data without requiring the number of clusters in advance. Mean shift clustering is widely used in image segmentation and is known to work efficiently for large dimensional data. Instead of pre-defining the number of clusters in advance as in k-means clustering, mean shift clustering specifies a kernel that it uses in clustering. We tested two types of mean shift kernels for

salient node clustering, a flat kernel and a Gaussian kernel. Another unsupervised clustering algorithm that we considered is Density-Based Spatial Clustering of Applications with Noise (DBSCAN) (Ester et al., 1996; Daszykowski et al., 2001). DBSCAN is a density-based algorithm that performs well on spatial points of arbitrary shape. This section evaluates the three different clustering algorithms (flat kernel mean shift, Gaussian kernel mean shift, and DBSCAN) for the two different types of missions (camera and sonar).

**1) Flat Kernel Mean Shift:** The flat kernel considers a ball of radius $\lambda$, and takes data points within $\lambda$ radius as inliers (i.e., members of a cluster). Starting from a random point, the algorithm repeatedly calculates the sample mean from points within range to update the mean until it converges. Once a mean does not change, it establishes a cluster. Then another random point from the remaining unclustered data is chosen and this procedure repeats until convergence. The radius, $\lambda$, that determines the boundary for inliers is defined by the user. Due to the shape of kernel, the resulting clusters have a spherical shape. For the current positional mean, $\boldsymbol{\mu}$, and a data point, $\mathbf{p}$, the flat kernel is defined as

$$K(\mathbf{p}, \boldsymbol{\mu}) \begin{cases} = 1, & \text{if } ||\mathbf{p} - \boldsymbol{\mu}|| \leq \lambda \\ = 0, & \text{otherwise} \end{cases}, \tag{4.3}$$

where the data point $\mathbf{p}$ refers to the positional vector ($\mathbf{p} = [x, y, z]^\top$).

**2) Gaussian Kernel Mean Shift:** The Gaussian kernel mean shift uses a Gaussian distribution for its kernel. This method also starts from a random sample point and accepts points within the $n$-$\sigma$ confidence bound[3] as inliers by measuring a Mahalanobis distance from the current mean. It follows the same mean update procedure as the flat kernel mean shift, but is different in using the $n$-$\sigma$ confidence bound in its inlier selection. Unlike the flat kernel mean shift, non-circular point distributions can be clustered properly by choosing $\Sigma$ for the Gaussian kernel, where the kernel is defined as

$$K(\mathbf{p}, \boldsymbol{\mu}) = \frac{1}{\sqrt{|2\pi\Sigma|}} \exp\left(-\frac{1}{2}(\mathbf{p} - \boldsymbol{\mu})^\top \Sigma^{-1} (\mathbf{p} - \boldsymbol{\mu})\right) \tag{4.4}$$

For both the flat and Gaussian kernels, the mean shift update rule is the same and is given by

$$\boldsymbol{\mu}' = \frac{\sum_{\mathbf{p}} K(\mathbf{p}, \boldsymbol{\mu}) \cdot \mathbf{p}}{\sum_{\mathbf{p}} K(\mathbf{p}, \boldsymbol{\mu})}, \tag{4.5}$$

with the order of complexity $O(n^2)$ for a set of $n$ points.

[3]Statistically, for the 3-DOF Gaussian, the 1-$\sigma$ bound contains 19.87%, the 2-$\sigma$ bound contains 73.85%, the 3-$\sigma$ bound contains 97.07%, and the 4-$\sigma$ bound contains 99.89% of the distribution. We used 4-$\sigma$ to accept 99.89% of the distribution as inliers.

**3) DBSCAN:** When the data to be clustered is a set of spatial points, there exists an efficient clustering algorithm that can be applied to this problem called DBSCAN (Daszykowski et al., 2001; Ester et al., 1996). DBSCAN is a density-based method for clustering spatially distributed data points by checking connectivity to other nearby data points. Unlike the kernel-based method, DBSCAN does not require knowledge of the shape of the clusters (e.g., spherical or Gaussian), and is known to work well for clusters of arbitrary shape. This method starts from a point in the dataset and scans through the entire point cloud, checking connectivity with neighboring points in an efficient way. The order of complexity is $O(n \log n)$ for a point set size $n$. DBSCAN is known to run fast and result in an intuitive spatial clustering, using a minimum number of points per cluster defined by the user. Since we have different track-line spacings depending upon the mission type (camera or sonar), we parameterize DBSCAN to evaluate the normalized distance with respect to mission track-line spacing in order to calculate connectivity. The original connectivity check, using Euclidean distance, is modified to consider the normalized distance with respect to the along-track-line spacing, $\Delta_d$, and cross-track-line spacing, $\Delta_h$,

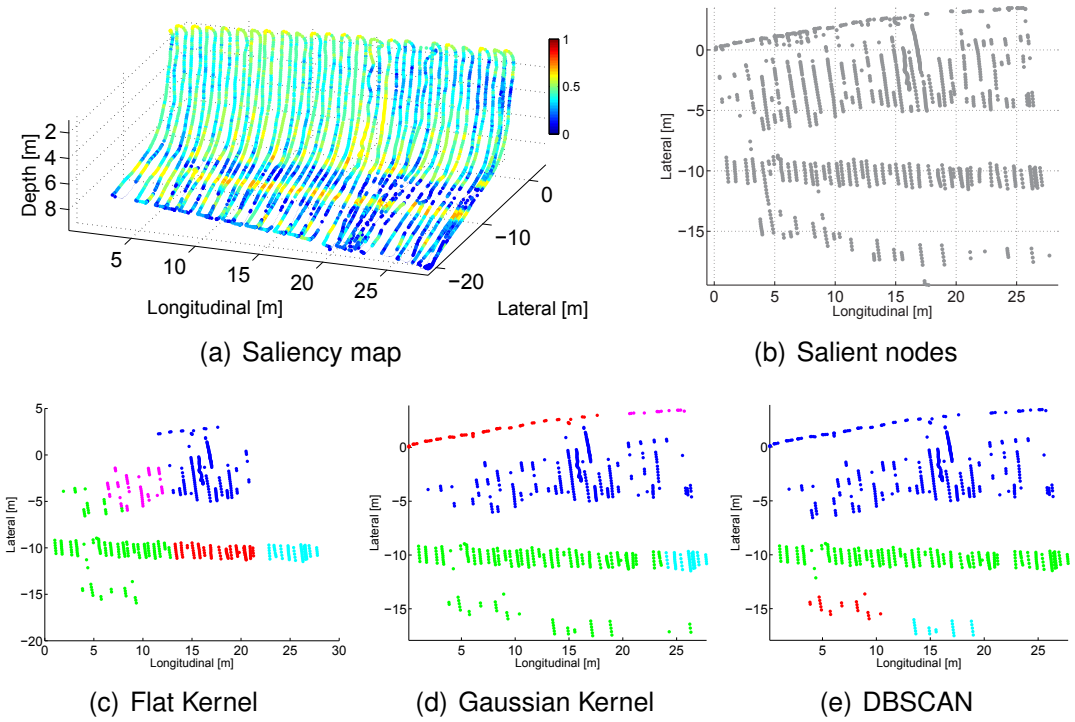
$$d(\mathbf{p}_i, \mathbf{p}_j) = \sqrt{\frac{(x_i - x_j)^2}{\Delta_d^2} + \frac{(y_i - y_j)^2}{\Delta_h^2} + (z_i - z_j)^2}. \tag{4.6}$$

These three clustering algorithms (Flat, Gaussian and DBSCAN) do not require the number of clusters in advance, and, thus, are applicable for PDN's online clustering module. Since the computational complexity of the PDN algorithm scales proportionally with the number of waypoints (i.e., number of clusters), we select the $N_{wp}$ largest clusters (based upon the number of members). For the two mean shift methods (where we do not have control of the minimum number of members in a cluster) we enforce a minimum cluster size when selecting the $N_{wp}$ largest clusters.

We evaluated these three clustering algorithms (flat kernel mean shift, Gaussian kernel mean shift and DBSCAN) over two different types of typical mission profiles, a narrow-spaced camera mission (Figure 4.7) and a wide-spaced sonar mission (Figure 4.8). For the clustering parameters, we used a $4$ m radius ($\lambda = 4$) for the flat kernel mean shift, a 4-$\sigma$ (or 99.89%) confidence bound for the Gaussian mean shift with $\Sigma = \begin{bmatrix} 1^2 & 0 & 0 \\ 0 & 2^2 & 0 \\ 0 & 0 & 1^2 \end{bmatrix} \mathrm{m}^2$ to capture the longitudinal distribution of the salient nodes, and a minimum number of 15 points in DBSCAN.

For both mission profiles (camera and sonar), the flat kernel mean shift resulted in more number of clusters than the others (Figure 4.7(c) and Figure 4.8(c)). Also, depending on the sample mean selection for a new cluster, the overall clustering result changes frequently, which is not desirable. Both the Gaussian kernel mean shift and DBSCAN successfully

**Figure 4.7** Online clustering for a densely-spaced camera mission; red circles designate the center of each cluster. (a) Trajectory with local saliency map overlaid. (b) Salient nodes based upon a threshold of $S_L^{\text{wp}} = 0.5$. The five largest clusters (c)–(e) are color-coded by group. (c) Flat kernel mean shift with radius $\lambda$ = 4 m. (d) Gaussian mean shift with 4-$\sigma$ confidence bound with $\Sigma = \begin{bmatrix} 1^2 & 0 & 0 \\ 0 & 2^2 & 0 \\ 0 & 0 & 1^2 \end{bmatrix}$ m$^2$. (e) DBSCAN with minimum number 15. For a densely spaced mission, DBSCAN results in a natural and intuitive clustering, since it is based on connectivity.




(a) Saliency map (b) Salient nodes

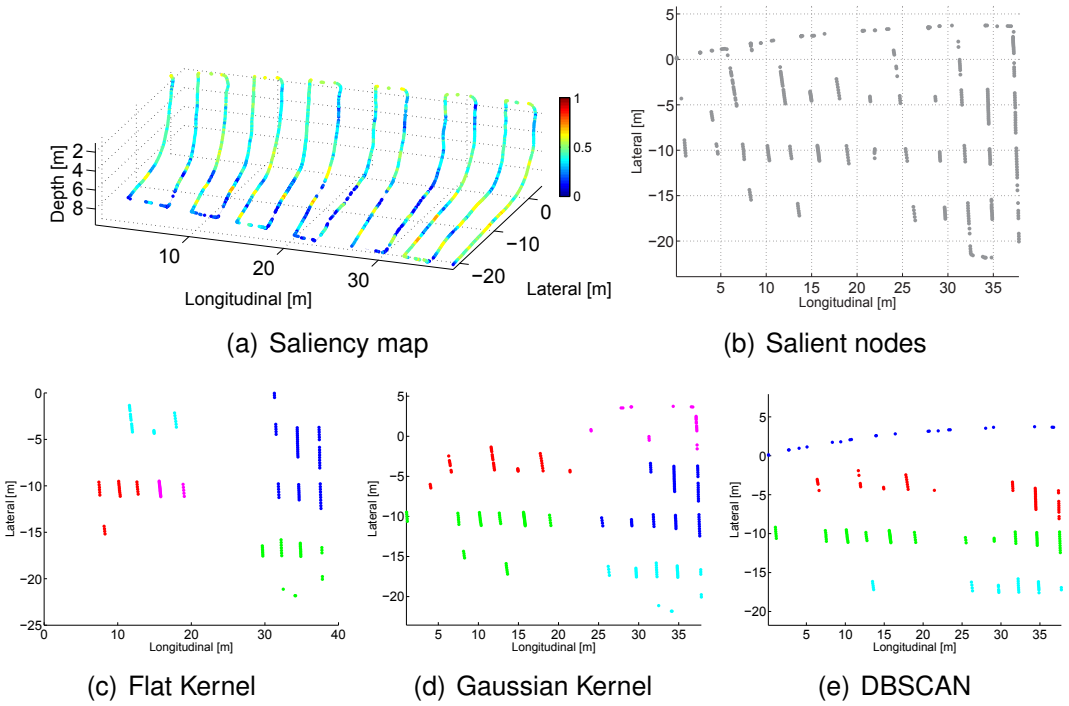
(c) Flat Kernel (d) Gaussian Kernel (e) DBSCAN

capture the elongated distribution of the salient regions, though DBSCAN (Figure 4.7(e) and Figure 4.8(e)) clusters the long-banded region as a single cluster, while the other two mean shift algorithms (Figure 4.7(c), Figure 4.7(d), Figure 4.8(c), and Figure 4.8(d)) cluster them into two or more groups. We concluded that DBSCAN outperforms the other two algorithms due to its connectivity-based formulation and computational efficiency and, therefore, is most appropriate for PDN's waypoint clustering module.

### Waypoint Selection

The final step of waypoint generation is to select a representative revisit node within each cluster based upon its global saliency level and effectiveness for loop-closure. Since the clusters are generated using local saliency, the members within each cluster guarantee texture-rich scenes. The next step then is to select a representative waypoint that is good for loop-closure. We propose to use global saliency for this purpose as a measure of visual

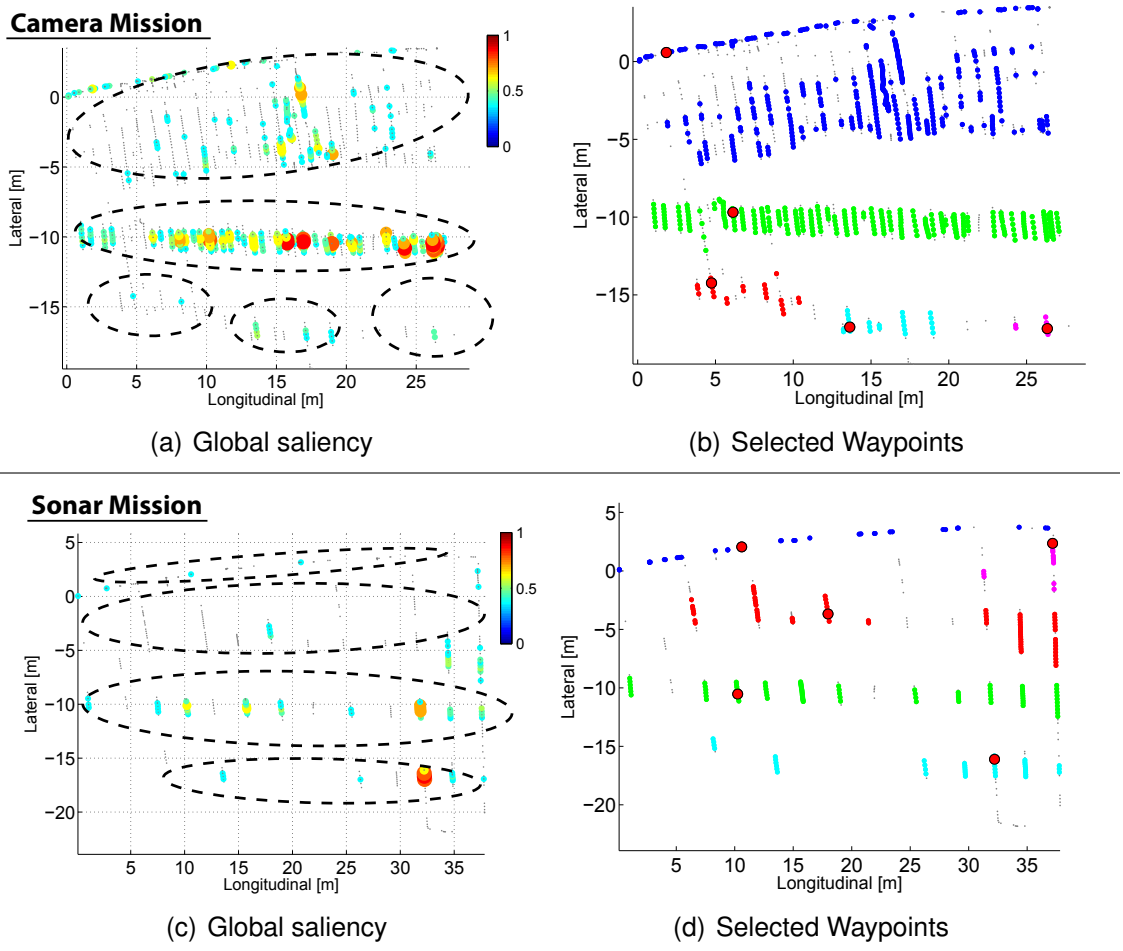
**Figure 4.8** Online clustering for a sparsely-sampled sonar mission; red circles designate the center of each cluster. (a) Trajectory with local saliency map overlaid. (b) Salient nodes based upon a threshold of $S_L^{\text{wp}} = 0.5$. The five largest clusters (c)–(e) are color-coded by group. (c) Flat kernel mean shift with radius $\lambda$ = 4 m. (d) Gaussian mean shift with 4-$\sigma$ confidence bound with $\Sigma = \begin{bmatrix} 1^2 & 0 & 0 \\ 0 & 2^2 & 0 \\ 0 & 0 & 1^2 \end{bmatrix}$ m$^2$. (e) DBSCAN with minimum number 15. Like the camera mission, DBSCAN successfully groups locally salient points into proper and stable clusters.




uniqueness for loop-closure. There are two criteria we consider when selecting a representative waypoint. First, the waypoint needs to be unique and discerning in the environment with a high global saliency level. Second, selection of the waypoint needs to be biased toward earlier indexes in the graph to induce a large loop-closure. For this bias term, we select the node with earliest index among all globally salient nodes within the cluster.

Considering these two factors in waypoint selection, the resulting waypoints for each type of mission profile are shown in Figure 4.9. In Figure 4.9(a) and Figure 4.9(c), globally salient nodes are overlaid on the locally salient nodes (gray). Dotted ellipses denote the extent of each cluster. Within each cluster, the node with earliest index among the globally salient nodes is selected as the representative waypoint (Figure 4.9(b) and Figure 4.9(d)). This is to bias toward loop-closures that produce a larger reward, which will be described in §4.3.3.4. The clustered waypoints are sorted by time and assigned with a waypoint number from 1 to $N_{wp}$—waypoint number 0 is reserved for exploration.

**Figure 4.9** Waypoint selection for two typical mission profiles (camera and sonar). Globally salient nodes are overlaid on the locally salient points as shown in (a) and (c). The gray dots represent the locally salient nodes used in clustering. Each cluster is roughly marked with a dotted ellipse. Globally salient nodes are color-coded by their global saliency level, and enlarged in size for visualization. The second column (b), (d) shows the selected waypoints for each cluster. Red circles represent the selected waypoint for each cluster based upon their visual uniqueness and usefulness for the loop-closure ($N_{wp} = 5$ is used here).




(a) Global saliency (b) Selected Waypoints

(c) Global saliency (d) Selected Waypoints

### 4.3.2 Path Generation

With this set of waypoints, the robot evaluates the amount of reward that can be obtained by revisiting these waypoints versus exploring. In this procedure, prior to the reward evaluation, the robot computes a shortest path from its current pose to each waypoint to evaluate the expected reward along that path. In a 2D space, this can be solved easily by ray-casting (Stachniss et al., 2005); however, in 3D, this becomes a geodesic (a shortest path on a curved surface) problem over a 3D manifold since the mission is constrained to the ship hull surface.

**Algorithm 2** Point-to-point path planning.

**Require:** Start node $\mathbf{x}_s$ and goal node $\mathbf{x}_g$
**Require:** Termination criteria $\epsilon$ and along-track-line survey parameter $\Delta_d$
  Set initial node: $\mathbf{x}_i \leftarrow \mathbf{x}_s$
  Set final node: $\mathbf{x}_f \leftarrow \mathbf{x}_g$
  Initialize path with empty set of nodes: $\mathcal{P} = \emptyset$

  **repeat**
    {Do global A* for a bisecting node}
    $\mathbf{x}_b = \text{argmin } \mathcal{D}(\mathbf{x}) \leftarrow$ Eqn (4.7) and Eqn (4.9)

    {Check local minima}
    **if** $\mathbf{x}_b$ is the closest node from $\mathbf{x}_f$ **then**
      $\mathbf{x}_b = \text{argmin } \mathcal{D}(\mathbf{x}) \leftarrow$ Eqn (4.7) and Eqn (4.9) with $w(S_L^k) = 1$ for all $k$
    **end if**

    **if** $\mathbf{x}_b$ is the closest node from $\mathbf{x}_i$ **then**
      {Interpolate between $\mathbf{x}_i$ and $\mathbf{x}_b$ with $\Delta_d$}
      $\{\mathbf{x}_p\} \leftarrow \text{interpolate}(\mathbf{x}_i, \mathbf{x}_b, \Delta_d)$

      {Add a set of interpolated nodes to the path}
      $\mathcal{P} \leftarrow [\mathcal{P}, \{\mathbf{x}_p\}]$

      {Reset initial node}
      $\mathbf{x}_i \leftarrow \mathbf{x}_b$
    **else**
      {Reset final node}
      $\mathbf{x}_f \leftarrow \mathbf{x}_b$
    **end if**
  **until** $\mathbf{x}_i$ and $\mathbf{x}_f$ are close enough ($||\mathbf{x}_i - \mathbf{x}_f|| < \epsilon$)
  **return** $\mathcal{P}$

For the general geodesic problem, Tenenbaum et al. (2000) and Silva and Tenenbaum (2002) present a nonlinear dimensionality reduction algorithm for this type of problem. These approaches improve the existing principal component analysis (PCA) and multidimensional scaling (MDS) based machine learning techniques by providing a three step procedure: *(i)* construction of a neighborhood graph, *(ii)* finding the shortest path on the graph, and *(iii)* conducting a reduced dimensional embedding by which the method finds correspondence between the original data to the data in the reduced dimension.

In our application, finding the shortest path can be viewed as a traditional point-to-point path planning problem without obstacles. We found that, although the surface is curved, applying A* (Russell and Norvig, 2003) globally on the existing nodes results in a fast and

globally optimal path on the sample nodes because the nodes are continuously distributed without obstacles. By using the existing nodes as sample nodes, we compute the path as a sequence of nodes. Applying global A* results in a bisecting node between a goal and a start node, and repeatedly applying global A* through bisecting nodes (Algorithm 2) computes a globally optimal shortest path over the sample nodes, called *milestones*. These milestones consist of existing nodes in the pose-graph. For the case when the distance between milestones is large, we interpolate virtual nodes along a straight line between two milestones in order to keep all virtual nodes within the along-track-line distance ($\Delta_d$ in §4.2.1). Generally, the resulting path will be a sequence of milestones and virtual nodes between the start and goal positions.

The bisecting node between two nodes is simply computed by evaluating a cost function $\mathcal{D}(\mathbf{x})$ for a node $\mathbf{x} = [x, y, z, \phi, \theta, \psi]^\top$, which is defined as

$$\mathcal{D}(\mathbf{x}) = \underbrace{d(\mathbf{x}_i, \mathbf{x})}_{\text{cost to initial node}} + \underbrace{d(\mathbf{x}_f, \mathbf{x})}_{\text{cost to final node}}, \tag{4.7}$$

where $\mathbf{x}_i$ is the initial node, $\mathbf{x}_f$ is the final node, and $d(\mathbf{x}_i, \mathbf{x}_k)$ is the heuristic cost defined to be Euclidean distance weighted with local saliency between two nodes:

$$d(\mathbf{x}_i, \mathbf{x}_k) = w(S_L^k) \cdot \sqrt{(x_i - x_k)^2 + (y_i - y_k)^2 + (z_i - z_k)^2}. \tag{4.8}$$

The weight term $w(S_L^k)$ is modeled as $2 - S_L^k$, which doubles the Euclidean distance to nodes of zero saliency, while preserving the original distance to nodes of maximal saliency. Due to this weighting, the path may fall into local minima. To cope with this, we impose a perturbation action (Kavraki and LaValle, 2008) by evaluating a pure Euclidean distance heuristic in the occurrence of local minima.

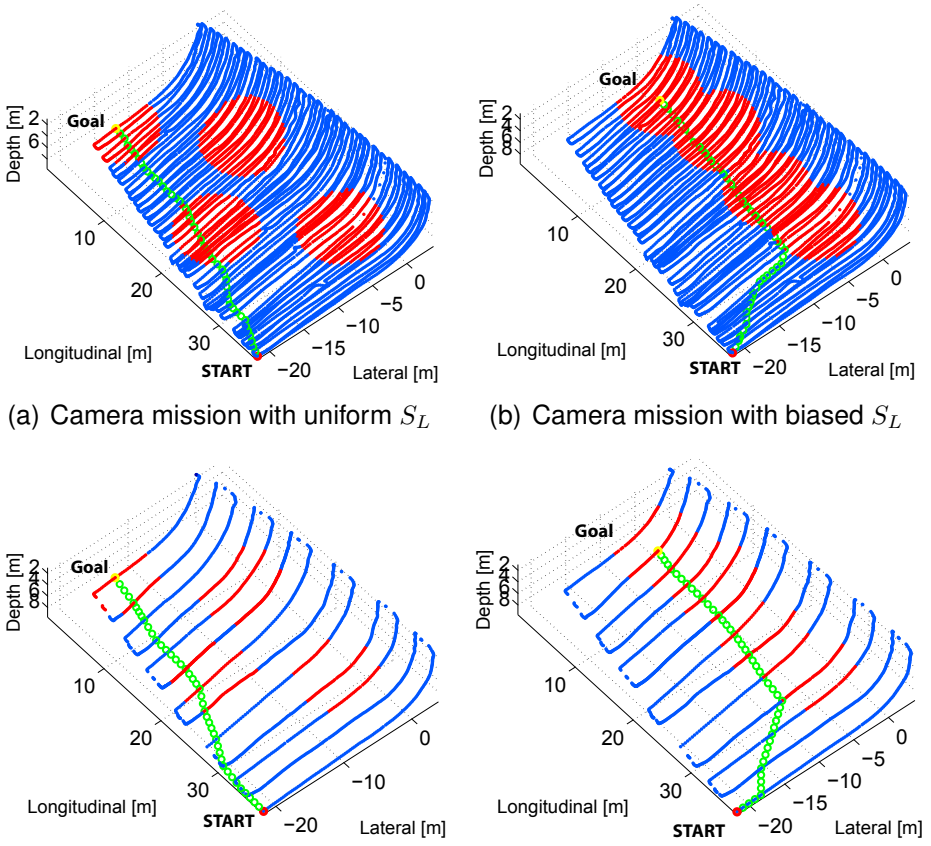
The bisecting node, $\mathbf{x}_b^*$, is the node that minimizes this cost, and is globally optimal when the cost is computed for all nodes $\mathbf{x}$ in the graph,

$$\mathbf{x}_b^* = \text{argmin}\, \mathcal{D}(\mathbf{x}) = \text{argmin} \Big( d(\mathbf{x}_i, \mathbf{x}) + d(\mathbf{x}, \mathbf{x}_f) \Big). \tag{4.9}$$

Repeated bisection of sample nodes yields a sequence of nodes, called milestones $\{\mathbf{x}_b\}$. During the generation of these milestones, we interpolate between milestones if necessary to complete a path, $\mathcal{P} = \{\mathbf{x}_p\}$, that a vehicle can follow with along-track-line spacing $\Delta_d$.

Figure 4.10 shows sample point-to-point path for the two typical types of missions. Using the current robot node as the starting node, the computed paths reach the waypoint labeled as the goal node. The pose-graph nodes are depicted as blue dots and the planned

**Figure 4.10** Point-to-point path planning for camera and sonar missions. For both missions, blue dots represent non-salient nodes in the graph with salient regions marked with red. The planned path is depicted with a sequence of green circles linking the start and the goal position. (a) and (b) are two sample point-to-point paths for camera missions on two different distributions of local saliency. (c) and (d) are two sample point-to-point paths for sonar missions for two different types of saliency distributions. Note that saliency weighted A* results in paths biased toward the salient regions in the environment.




(a) Camera mission with uniform $S_L$ (b) Camera mission with biased $S_L$

(c) Sonar mission with uniform $S_L$ (d) Sonar mission with biased $S_L$

path is marked with green circles with red nodes representing the locally salient regions. For the narrow-spaced camera mission, most of the nodes in the path are from the existing sample nodes, as shown in Figure 4.10(a)–(b). On the other hand, for the wide-spaced sonar mission, there are not enough nodes in the pose-graph to build a shortest path by linking directly, and, hence, virtual nodes are placed between sample nodes. For all cases, the resulting path is biased toward the salient regions in the environment.

### 4.3.3 Reward for a Path

Reward for a path is defined in terms of the robot's navigation uncertainty and achieved area coverage. For the robot uncertainty we use the terminating pose covariance, and for

the area coverage we use an area coverage ratio for the performance measure.

#### 4.3.3.1 Saliency-based Measurement Likelihood

Along the point-to-point candidate path, we solve for the estimated final robot uncertainty using an extended information filter (EIF). Using expected odometry and camera measurements along the path, the robot can estimate the final resulting covariance along that trajectory. However, estimating the amount of information gained from future camera measurements is not exact, and we need to develop a way of approximating the camera measurement likelihood. Camera measurements are binary, either success (1) or failure (0), and the camera link event, $L$, is a Bernoulli random variable. When adding a set of expected camera measurements, we use local saliency to empirically model the likelihood of successful registration in order to compute the expected information gain along the path. The observation is that we can model this probability using statistics from prior SLAM and saliency results.

For a Bernoulli random variable $L$, we seek to model a probability of a link to be successful, $P_L$,

$$P_L(l = 1) = P_L = \text{Prob (a link to be successful).} \tag{4.10}$$

Because each link is associated with two images and their local saliency ($S_{L_c}$) levels, the current node saliency ($S_{L_c}$) and the target node saliency ($S_{L_t}$), we can build a probability of a link to be successful ($P_L$) as a function of these two saliency levels,

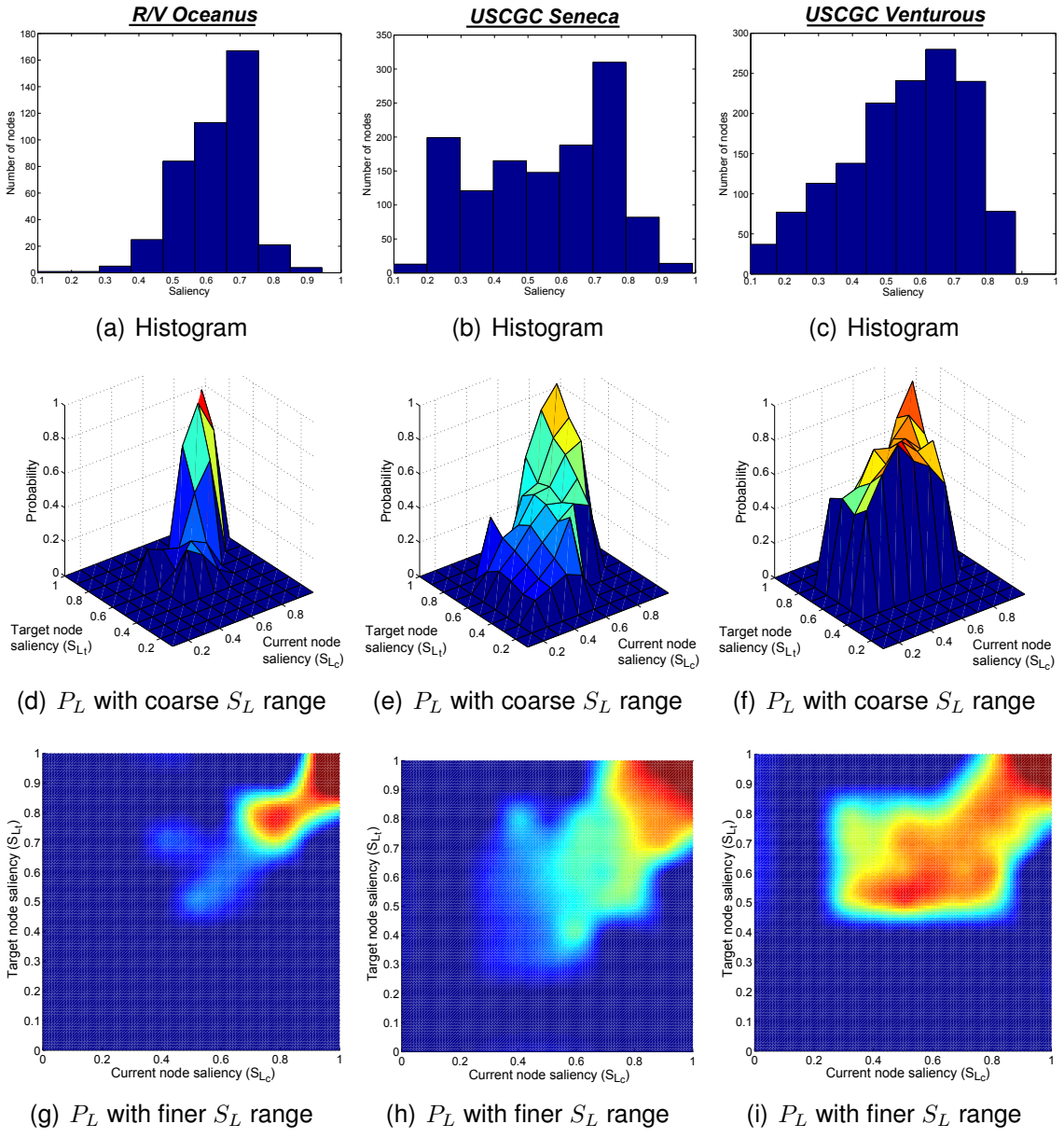
$$P_L = P_L(l = 1; S_{L_c}, S_{L_t}) \sim \text{Bernoulli.} \tag{4.11}$$

To empirically measure this, we use data collected from three different prior missions from three different vessels, the ***R/V Oceanus***, the *USCGC Venturous* and the *USCGC Seneca* as shown in Figure 4.11. Next, we generate a scatter plot from this data and divide the scatter plot into a set of bins with bin size of $\delta = 0.1$ (Figure 4.12(a)). The empirical probability of a link to be successful ($P_L$) is then calculated by counting the number of proposed links and the number of verified links in each bin, which builds up a coarse model Figure 4.12(c) as a function of the two associated saliency values, the current node saliency and the target node saliency:

$$P_L(l = 1; S_{L_c}, S_{L_t}) = \frac{N_v(S_{L_c}, S_{L_t})}{N_p(S_{L_c}, S_{L_t})}, \tag{4.12}$$

$$P_L(l = 0; S_{L_c}, S_{L_t}) = 1 - \frac{N_v(S_{L_c}, S_{L_t})}{N_p(S_{L_c}, S_{L_t})}. \tag{4.13}$$

**Figure 4.11** Empirical probability of link success for visual SLAM surveys on three different vessels. The successful link probability, $P_L$, is generated for three missions on three different vessels, the *R/V Oceanus* (the first column), the *USCGC Venturous* (the second column), and the *USCGC Seneca* (the last column). A histogram of nodes with respect to their saliency level is given in the first row. Each vessel shows a different characteristic in feature distribution. The *USCGC Venturous* has a texture-rich hull, the *USCGC Seneca* has a relatively clean hull showing uniformly distributed node saliency, and the *R/V Oceanus* is biased toward a high saliency level. The coarse $P_L$ model is computed using (4.12) and (4.13) (i.e., (d), (e), (f)) and then smoothed (i.e., (g), (h), (i)).

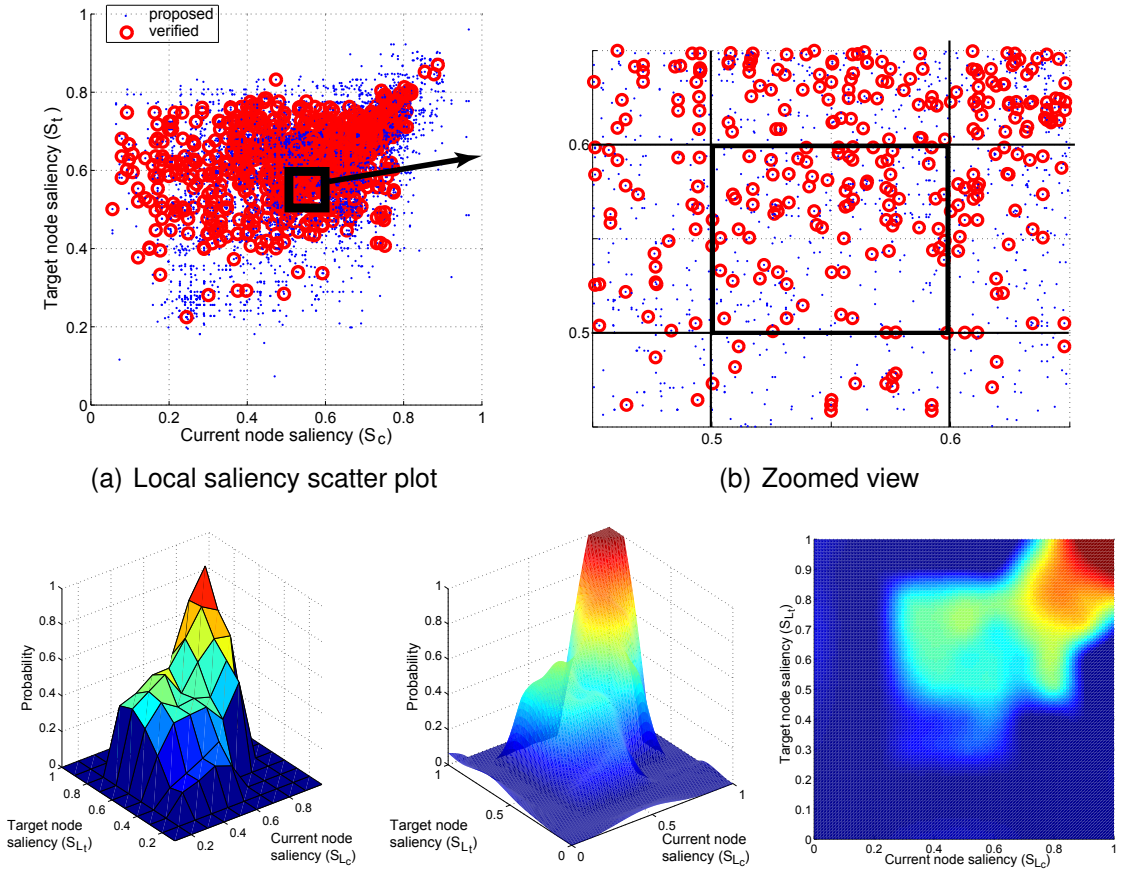



(a) Histogram (b) Histogram (c) Histogram

(d) $P_L$ with coarse $S_L$ range (e) $P_L$ with coarse $S_L$ range (f) $P_L$ with coarse $S_L$ range

(g) $P_L$ with finer $S_L$ range (h) $P_L$ with finer $S_L$ range (i) $P_L$ with finer $S_L$ range

**Figure 4.12** Construction of empirical probability of successful link, $P_L$. Empirical model of $P_L$ is generated as a function of pairwise saliency level ($S_{L_c}$ and $S_{L_t}$). To model this, we use a scatter plot of links as in (a) and (b). The links and saliency scores are generated using data from three different missions from three different vessels. A coarse distribution is built by counting the number of verified links in the scatter plot (Figure 4.12(a)). Then, surface-fitting to this coarse distribution results in the finer distribution as shown in (d) and (e).




(a) Local saliency scatter plot (b) Zoomed view

(c) $P_L$ with coarse $S_L$ range (d) $P_L$ with finer $S_L$ range (e) Top view

Here, $N_v(S_{L_c}, S_{L_t})$ represents the number of verified links in the bin with current node saliency $[S_{L_c}, S_{L_c} + \delta)$ and target node saliency $[S_{L_t}, S_{L_t} + \delta)$, and $N_p(S_{L_c}, S_{L_t})$ is the number of proposed links in the same range defined by $S_{L_c}$ ,$S_{L_t}$, and $\delta$. This coarse empirical distribution of $P_L(l; S_{L_c}, S_{L_t})$ is then smoothed by applying a surface fitting algorithm[4] as in Figure 4.12(d) and Figure 4.12(e) to produce a finer scale model.

From this model, the probability of a link to be successful is computed as a function of the two saliency scores associated with the link. Using this measure for the probability of a successful camera measurement, we evaluate the expected information gain from camera measurements in the following section.

[4] A Matlab open source implementation called Gridfit by D'Errico (2010) is used in surface smoothing.

#### 4.3.3.2 Robot Uncertainty ($\mathcal{U}_{robot}$)

Part of the reward function needs to represent the robot's navigation uncertainty. There are two popular metrics in defining this uncertainty. One is to consider the mutual information (MI) gain obtained along the path (Stachniss et al., 2005; Davison et al., 2007; Levine, 2010; Valencia et al., 2011). This MI approach accumulates the information gain from measurements along a path to evaluate the reward of the path. The other approach (Sim and Roy, 2005; Sim, 2005; Prentice and Roy, 2009), which is closer to Fisher information (FI), examines the final terminating covariance to estimate the expected uncertainty reduction along the path.

Two recent works in belief space path planning share similarities to our approach: Prentice and Roy (2009) and Valencia et al. (2011). The Belief Roadmaps (BRMs) by Prentice and Roy is analogous to PDN in that it evaluates the terminating covariance as a measure of uncertainty along a path. To efficiently evaluate the covariance along the path, they introduced a one-step covariance transfer function using the Redheffer star product (Redheffer, 1946) for linear covariance propagation and update. However, this approach is not applicable to our visual pose-graph SLAM application for two reasons. First, BRMs does not keep a history of poses, but instead only tracks the most recent node in the graph. It is not obvious how to extend their one-step transfer function to the case of a pose-graph representation. Secondly, BRMs achieves significant computational efficiency by pre-calculating a large stacked matrix for their star product operator, which violates the assumption of not having prior information on the area. Since PDN starts with no prior nodes, pre-computing the matrices for the one-step transfer function is infeasible.

Valencia et al. expanded BRMs to a pose-graph without using sample nodes of a given area. Their implementation is in two steps, (*i*) running SLAM to build a pose-graph and (*ii*) using the graph to find a point-to-point optimal path along the pose-graph. During path evaluation, the accumulated relative uncertainty drop (in terms of entropy) is evaluated to choose a path with minimum uncertainty. After building the pose-graph, the authors invert the information matrix so that marginal covariances are accessible for the path evaluation phase. Their implementation presents a novel attempt to integrate SLAM and path planning by planning on the pose-graph, yet fails to plan and perform SLAM concurrently. A comparison between Valencia et al. (2011) and PDN for point-to-point path will be provided at the end of this section.

For PDN, we use the FI approach by evaluating the resulting covariance matrix for integrated SLAM and path planning. Because the camera measurement is not certain, we compute the expected information gain from a path, and evaluate the expected terminating covariance matrix. We use the determinant of the covariance matrix (not the trace) as a

measure of navigation uncertainty due to its monotonicity. When the initial covariance is large, as in the case of PDN, the trace of the propagated covariance loses monotonicity depending on the motion the robot takes. A detailed derivation and example are found in Appendix §C.

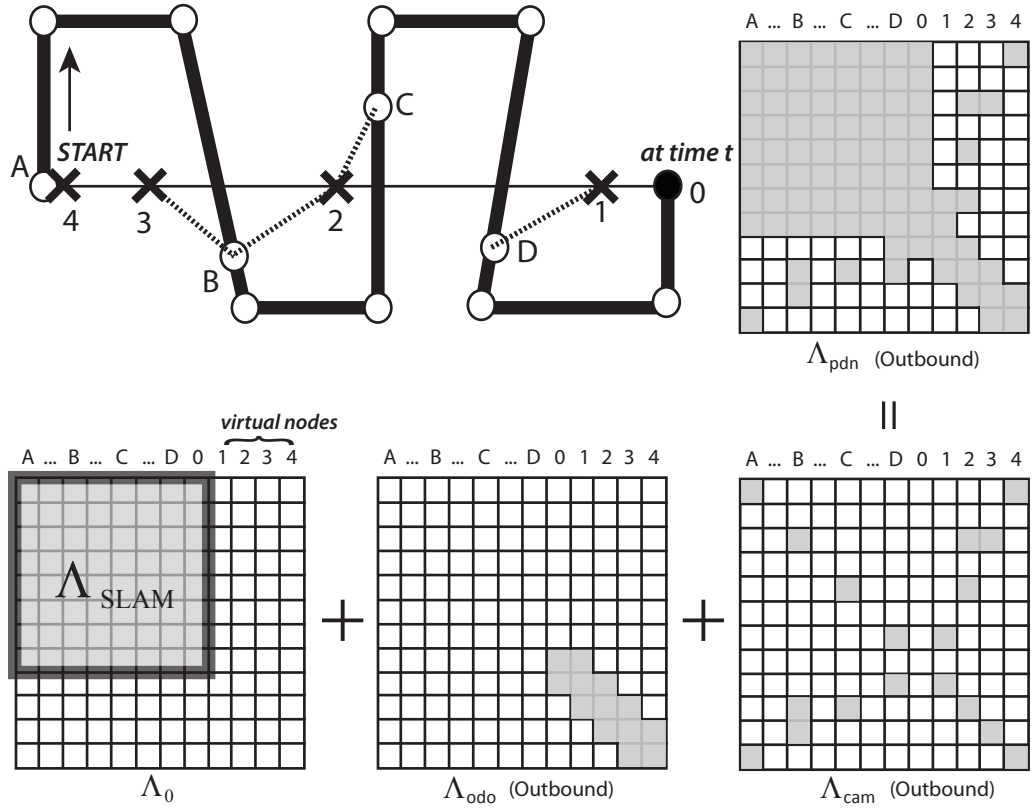
For each waypoint, we compute the expected covariance propagation along the point-to-point path generated in §4.3.2. The resulting robot uncertainty ($\mathcal{U}_{robot}$) from a revisit action is computed as the expected terminating covariance from the round-trip travel to the waypoint. Two algorithms were considered in computing this terminating covariance, an exactly sparse delayed-state filter (ESDF) (Eustice et al., 2008) and iSAM (Kaess et al., 2012). In this thesis, we chose to bookkeep an information matrix during the SLAM process using an ESDF as a way to estimate the terminating covariance from a revisit action. This includes taking a snapshot of the current information matrix and adding delta information from expected measurements. The time complexity of evaluation for each waypoint, is, $O(S(n) \cdot l(wp))$, where $S(n)$ is the complexity in covariance recovery for the last pose and $l(wp)$ is the path length of the revisit action.

The process of constructing the information matrix using an ESDF is illustrated in Figure 4.13. Note that only the outbound portion of the revisit action is shown for visual clarity (PDN computes the information for the round-trip). The ESDF-based approach is to construct a small extended information filter (EIF) by adding a set of odometry constraints and a set of expected camera measurements in the form of delta information to the current SLAM information matrix, $\Lambda_0$. In the toy example of Figure 4.13, a robot starts from node A and moves along the thick line, reaching the current node, denoted 0, at time $t$. To evaluate the terminating covariance from revisiting the A node, two sources of delta information are added: one from odometry ($\Lambda_{\mathtt{odo}}$) and the other representing camera constraints ($\Lambda_{\mathtt{cam}}$). The revisit action is marked with a thin line linking 0 and A with virtual nodes (1,2,3, and 4) along the path to A. Nodes A,B,C, and D are existing nodes in the pose-graph, and also are candidate nodes with which these virtual nodes make measurements. As one can see, the delta information from odometry constraints are only associated with virtual nodes 1–4, whereas the information from camera measurements add off-diagonal matrices for pairwise camera measurements. Lastly, summing up these three information matrices ($\Lambda_0, \Lambda_{\mathtt{odo}}, \Lambda_{\mathtt{cam}}$) builds the expected information matrix from PDN ($\Lambda_{\mathtt{pdn}}$),

$$\Lambda_{\mathtt{pdn}} = \Lambda_0 + \Lambda_{\mathtt{odo}} + \Lambda_{\mathtt{cam}}. \tag{4.14}$$

The expected delta information from odometry measurements is built from a sequence of virtual nodes. From the current node $\mathbf{x}_0$, odometry noise, and a path $\mathcal{P}$ (as a sequence

**Figure 4.13** Robot pose uncertainty propagation. Illustration of the information matrix through robot pose covariance propagation in PDN is presented. Only the outbound revisit action is illustrated for simplicity (PDN computes the information for the round-trip). The robot starts from node A moving along the thick line, and reaches the current node 0 at time $t$. This illustration shows construction of the PDN information matrix when the robot executes a revisit action from the current node 0 to a revisit point A. The revisit action is marked with a thin line linking 0 and A with virtual nodes 1,2,3, and 4 along the revisit path to A. Nodes A,B,C, and D are existing nodes in the pose-graph, and also are the candidate nodes that these virtual nodes make camera measurements with. The expected camera measurements are marked with a dotted line between 1–4 and A–D. Building the resulting information matrix ($\Lambda_{\mathtt{pdn}}$) consists of three parts, a snapshot of the current information matrix ($\Lambda_0$), the expected odometry induced information matrix ($\Lambda_{\mathtt{odo}}$), and the expected camera induced information matrix ($\Lambda_{\mathtt{cam}}$).



of $p$ virtual nodes from §4.3.2 $\{\mathbf{x}_i, \cdots, \mathbf{x}_p\}$), $\Lambda_{\texttt{odo}}$ is computed by adding all expected odometry measurements for the round-trip travel to the waypoint along the revisit path:

$$\begin{aligned}\Lambda_{\texttt{odo}} &= \sum_{i=0}^{p-1} \mathrm{H}_{\texttt{odo}_{i,i+1}}^{\top} \cdot \mathrm{Q}_{i,i+1}^{-1} \cdot \mathrm{H}_{\texttt{odo}_{i,i+1}} \quad \cdots \text{Outbound} \\ &+ \sum_{i=p}^{1} \mathrm{H}_{\texttt{odo}_{i,i-1}}^{\top} \cdot \mathrm{Q}_{i,i-1}^{-1} \cdot \mathrm{H}_{\texttt{odo}_{i,i-1}} \quad \cdots \text{Inbound} \end{aligned} \tag{4.15}$$

The noise for the odometry constraint ($\mathrm{Q}_{i,i+1}$) is scaled with the travel distance between two nodes, $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$. The odometry measurement model is the relative-pose between two sequential nodes ($\mathbf{x}_i$ and $\mathbf{x}_{i+1}$) and can be represented using the tail-to-tail operation by Smith et al. (1990) with Jacobian $\mathrm{H}_{\texttt{odo}_{i,i+1}}$,

$$\mathbf{z}_{i,i+1} = \mathbf{x}_{i,i+1} = \ominus\mathbf{x}_{l,i} \oplus \mathbf{x}_{l,i+1} = \ominus\mathbf{x}_i \oplus \mathbf{x}_{i+1} \tag{4.16}$$

$$\mathrm{H}_{\texttt{odo}_{i,i+1}} = \left[0, \cdots, 0, \frac{\partial \mathbf{x}_{i,i+1}}{\mathbf{x}_i}, \frac{\partial \mathbf{x}_{i,i+1}}{\mathbf{x}_{i+1}}, 0, \cdots, 0\right]. \tag{4.17}$$

In the above equation, $\mathbf{x}_{i,i+1}$ is the odometry measurement between node $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$, where $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$ are written with respect to the local frame $\{l\}$ and the subscript $l$ is omitted for convenience in (4.16). The resulting Jacobian ($\mathrm{H}_{\texttt{odo}_{i,i+1}}$) is sparse with nonzero block matrices on the $i^{\text{th}}$ and $(i+1)^{\text{th}}$ element. Summing all odometry information, thus, results in a block-tridiagonal matrix as depicted in Figure 4.13.

For the camera measurements, we similarly add all expected camera measurements along the revisit path. Because PDN proposes the same number of link hypotheses ($n_{plink}$) as in the normal SLAM process, there are multiple ($n_{plink}$) expected camera measurements per each virtual node along the path. When a virtual node is $\mathbf{x}_i$ and the candidate paired for camera measurement is node $\mathbf{x}_m$[5], the camera measurement between $\mathbf{x}_i$ and $\mathbf{x}_m$, and its related Jacobian ($\mathrm{H}_{\texttt{cam}_{m,i}}$), are as defined in (2.3), and are re-written here for convenience:

$$\mathbf{z}_{mi} = h_{\texttt{5dof}}(\mathbf{x}_m, \mathbf{x}_i) = \left[\alpha_{im}, \beta_{im}, \phi_{im}, \theta_{im}, \psi_{im}\right]^{\top}, \tag{4.18}$$

$$\mathrm{H}_{\mathbf{x}} = \begin{bmatrix} 0 & \cdots & \frac{\partial \mathbf{h}_{\texttt{5dof}}}{\partial \mathbf{x}_m} & \cdots & 0 & \frac{\partial \mathbf{h}_{\texttt{5dof}}}{\partial \mathbf{x}_i} & \cdots & 0 \end{bmatrix} = \mathrm{H}_{\texttt{cam}_{m,i}}. \tag{4.19}$$

Using the above measurement model and its first-order Jacobian, the expected information

[5] Since $\mathbf{x}_m$ is the existing node, it has earlier index than $\mathbf{x}_i$ and usually comes before $\mathbf{x}_i$.

gain from camera measurements is calculated as

$$
\begin{aligned}
\Lambda_{\mathtt{cam}} &= \sum_{i=0}^{p-1} \sum_{m \in \mathcal{L}_i} P_L(l=1; S_{L_c}, S_{L_t}) \cdot \mathrm{H}_{\mathtt{cam}_{m,i}}^{\top} \mathrm{R}^{-1} \mathrm{H}_{\mathtt{cam}_{m,i}} + P_L(l=0; S_{L_c}, S_{L_t}) \cdot 0_{n \times n} \\
&+ \sum_{i=p}^{1} \sum_{m \in \mathcal{L}_i} P_L(l=1; S_{L_c}, S_{L_t}) \cdot \mathrm{H}_{\mathtt{cam}_{m,i}}^{\top} \mathrm{R}^{-1} \mathrm{H}_{\mathtt{cam}_{m,i}} + P_L(l=0; S_{L_c}, S_{L_t}) \cdot 0_{n \times n} \\
&= \sum_{i=0}^{p-1} \sum_{m \in \mathcal{L}_i} P_L(l=1; S_{L_c}, S_{L_t}) \cdot \mathrm{H}_{\mathtt{cam}_{m,i}}^{\top} \mathrm{R}^{-1} \mathrm{H}_{\mathtt{cam}_{m,i}} \quad \cdots \text{Outbound} \\
&+ \sum_{i=p}^{1} \sum_{m \in \mathcal{L}_i} P_L(l=1; S_{L_c}, S_{L_t}) \cdot \mathrm{H}_{\mathtt{cam}_{m,i}}^{\top} \mathrm{R}^{-1} \mathrm{H}_{\mathtt{cam}_{m,i}} \quad \cdots \text{Inbound}, \qquad (4.20)
\end{aligned}
$$

where $\mathrm{R}$ is the fixed camera measurement noise[6], $\mathcal{L}_i$ is the index set of camera measurement candidates of $i$, and $P_L(l = 1; S_{L_c}, S_{L_t})$ is the empirical probability of the link to be successful. Unlike odometry measurements, not all expected camera measurements are available and depend greatly upon the visual feature distribution within the environment. To account for this, we have modeled the probability of a link to be successful ($P_L$) as a function of two saliency scores associated with the link as in §4.3.3.1. From Figure 4.12(d), the two saliency scores in a link specify the probability of a link to be successful, $P_L$.

Finally, adding these three information matrices (4.14) yields the expected information matrix for pursuing a virtual path to the waypoint. For the reward calculation, we are interested in the final pose uncertainty in order to evaluate the usefulness of this action. In other words, we are only interested in the last block covariance matrix ($\Sigma_{nn}$) where the $n^{\text{th}}$ node corresponds to the last node in the virtual pose-graph. This can be efficiently obtained by computing the $i^{\text{th}}$ column of the covariance matrix ($\Sigma_{*i}$) using the $i^{\text{th}}$ basis of the identity matrix ($\mathrm{I} = [\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_n]$) (Eustice et al., 2006b), which avoids inverting the entire information matrix ($\Lambda_{\mathtt{pdn}}$),

$$
\begin{aligned}
\Lambda_{\mathtt{pdn}}\, \Sigma &= \mathrm{I}_{n \times n}, \\
\Lambda_{\mathtt{pdn}}\, \Sigma_{*n} &= \mathbf{e}_n. \qquad (4.21)
\end{aligned}
$$

After obtaining the last block-column matrix $\Sigma_{*n}$, taking the last block covariance reveals the terminating covariance matrix $\Sigma_{nn}$ from the revisit action. This is the terminating covariance for the $k^{\text{th}}$ waypoint, $\Sigma_{nn}^{k}$, and is computed for all $N_{wp}$ waypoints. This round-trip pose covariance is calculated for each waypoint (i.e., for each revisit action), and is com-

[6]For the 5DOF camera measurement covariance, we assume $\pm 1^{\circ}$ noise for azimuth and elevation, and $\pm 0.1^{\circ}$ noise for the orientation change.

pared to the covariance from exploration.

The terminating covariance for exploration is computed by propagating the current covariance one step forward. From the current SLAM node, we compute the resulting covariance assuming that the previous odometry holds for this one-step propagation. Index $r$ refers to the current robot node, which is also the last node in the existing pose-graph, and all nodes later than $r$ are virtual.

$$\begin{aligned}
\Sigma_{\text{exp}} = \Sigma_{r+1,r+1} &= \mathrm{H}_{\circ\mathrm{d}\circ_{r,r+1}} \Sigma_{rr} \mathrm{H}^{\top}_{\circ\mathrm{d}\circ_{r,r+1}} && (4.22)\\
\mathbf{x}_{r,r+1} &= \mathbf{x}_r \oplus \mathbf{x}_{r-1,r} && (4.23)\\
\mathrm{H}_{\circ\mathrm{d}\circ_{r,r+1}} &= \left[0, \cdots, 0, \frac{\partial \mathbf{x}_{r,r+1}}{\mathbf{x}_r}, 0, \cdots, 0\right] && (4.24)
\end{aligned}$$

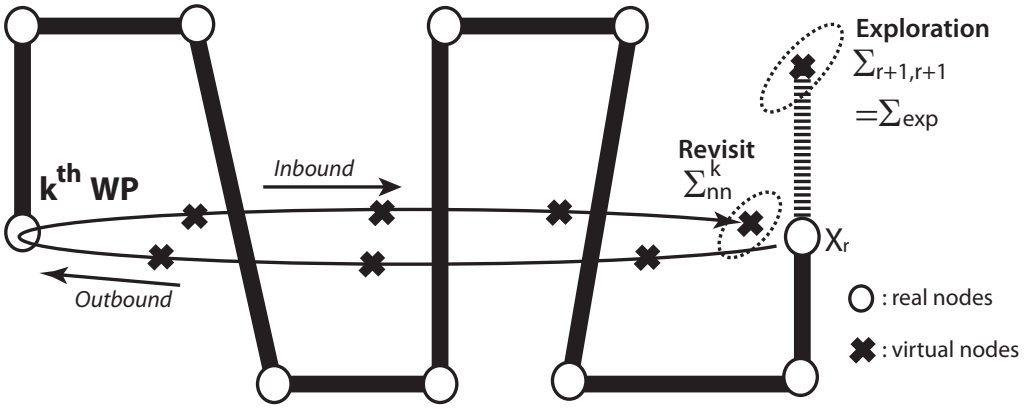
Lastly, the reward term for robot uncertainty, $\mathcal{U}^k_{robot}$, is computed as the ratio of the localization uncertainty for the next-best-action to the user-defined allowable navigation uncertainty, $\Sigma_{\text{allow}}$. For the $k^{\text{th}}$ waypoint, the robot uncertainty term is defined as

$$\begin{aligned}
\mathcal{U}^{k=0}_{robot} &= \begin{cases} 0, & \text{if } \frac{|\Sigma_{\text{exp}}|}{|\Sigma_{\text{allow}}|} < 1 \\ \frac{|\Sigma_{\text{exp}}|^{\frac{1}{6}}}{|\Sigma_{\text{allow}}|^{\frac{1}{6}}}, & \text{otherwise} \end{cases} \\
\mathcal{U}^{k>0}_{robot} &= \frac{|\Sigma^k_{nn}|^{\frac{1}{6}}}{|\Sigma_{\text{allow}}|^{\frac{1}{6}}}, \quad k = 1, \cdots, N_{wp}
\end{aligned} \tag{4.25}$$

where $k = 0$ is the candidate exploration action, $k > 0$ are the $1, \cdots, N_{wp}$ candidate revisit waypoints, and we have taken the $6^{\text{th}}$ root of the 6-DOF pose determinant in the numerator and denominator terms so that individually their SI units are $\mathrm{m} \cdot \mathrm{rad}$, which provides a more physically meaningful length scale for taking ratios. We use the entire 6-DOF pose covariance to preserve monotonicity. Kelly (2004) shows that monotonicity is not preserved in propagated covariances when considering only subelements of the covariance matrix (Appendix §C). Hence using the entire matrix allows us to preserve monotonicity, but yields a measure with units of both position and orientation.

This $\mathcal{U}^k_{robot}$ captures the robot navigation uncertainty term in the reward calculation. Basically, PDN compares the two propagated uncertainties from revisiting and exploring, and then chooses the smaller one as in Figure 4.14. When the expected exploration covariance is below the allowable covariance, the cost in the robot pose uncertainty term, $\mathcal{U}^0_{robot}$, is zero, leading the robot to pursue exploration. On the other hand, when the exploration covariance exceeds the allowable covariance, then the robot pose uncertainty term for exploration, $\mathcal{U}^0_{robot}$, is compared against all candidate revisit actions, $\mathcal{U}^{k>0}_{robot}$, which will be

**Figure 4.14** Robot pose uncertainty from revisiting versus exploration. Two terminating node uncertainties from revisiting and exploration are compared. Real nodes on the graph are shown as circles whereas virtual nodes are marked with 'X'.



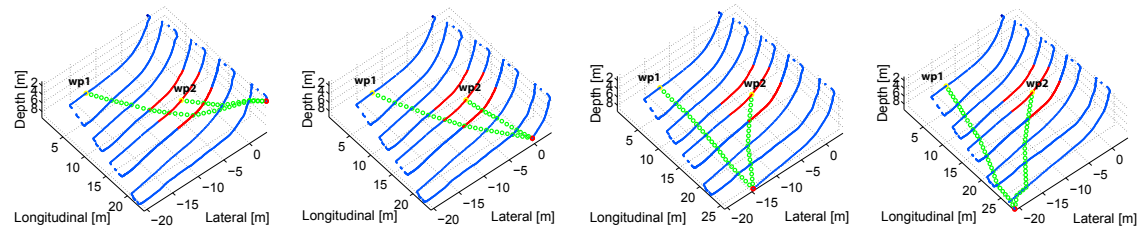

smaller when revisiting is likely to obtain enough loop-closures to overcome the increased navigation uncertainty from detouring. Unlike previous studies in active exploration of Bourgault et al. (2002), Makarenko et al. (2002) and Stachniss et al. (2005), where the authors did not consider the actual likelihood of obtaining perceptual loop-closures, our approach introduces a realistic expectation in the reward calculation for the likelihood of camera loop-closures based upon visual saliency.

### Measurement Likelihood using Saliency: A comparison to Previous Studies

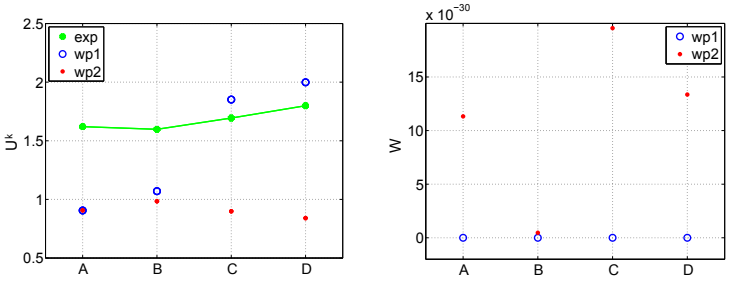
Two point-to-point planning approaches, Valencia et al. (2011) and Prentice and Roy (2009), were mentioned for their similarity to the PDN's robot uncertainty term $\mathcal{U}_{robot}$ but are different in that they did not model or take into account the camera's non-uniform registration performance, which is a function of scene visual feature content. Before introducing PDN's second area-coverage term in the reward computation, we first present the effectiveness of using saliency in the measurement likelihood to capture this effect.

For the evaluation, we compare the $\mathcal{U}_{robot}$ uncertainty term in PDN to the metric developed by Valencia et al. (2011) who evaluated work ($W$) required for a path using MI to select an optimal path with less work required. As shown in Figure 4.15, PDN's $\mathcal{U}_{robot}$ term (4.25) is computed and compared to Valencia's required work for a path for four different cases (A to D) with two waypoints. In this comparison, PDN estimates a more accurate uncertainty propagation by taking into account the camera registration likelihood, while Valencia's yields a rather optimistic evaluation on the loop-closure likelihood. An example of such a case would be the first waypoint (wp1) in sample paths C and D, which travel over non-salient regions where camera measurements are less likely. For these cases, PDN es-

**Figure 4.15** Comparison of PDN's robot uncertainty calculation relative to previous work. For four different scenarios with two waypoints each, PDN's uncertainty term, $\mathcal{U}_{robot}$, has been compared to the work, $W$, required for a path as calculated by Valencia et al. (2011). (a)-(d) depict trajectories with sample paths to two different waypoints, where nodes are color-coded by their saliency level. The first waypoint (wp1) is located in a low saliency region and the second waypoint is at a highly salient region. Among four cases, A and B are when the path to the first waypoint passes through a salient region on the map, and C and D are when it travels through an entirely non-salient region. (e) shows the evaluation from PDN's uncertainty term with respect to each case. A and B are predicted to obtain loop-closures on the way to the wp1, and scores a lower $\mathcal{U}_{robot}$ uncertainty for $k = 1$. When the path to wp1 no longer passes through a salient region (C and D), the $\mathcal{U}_{robot}$ becomes higher than the exploration (green line) and $\mathcal{U}_{robot}$ for $k = 2$ (red dot) is lower. (f) Valencia et al. computes required work for a path in order to choose paths with minimal required work. In this computation, no consideration to the actual measurement availability is addressed and the metric is mainly computed based upon geometry only. When we evaluate this required work for each case, the first waypoint always yields the minimal work and is chosen to be the proper waypoint to revisit although the path may not results in any loop-closures on the way. (g) Lastly, the uncertainty change before and after revisiting a waypoint is plotted to compare the estimates from (e) and (f). Plotted is the determinant of the 6-DOF robot pose covariance. Green dots are the uncertainty of the robot before revisiting a waypoint, where blue (wp1) and red (wp2) are the uncertainty measured after revisiting a waypoint. The resulting uncertainty after revisiting wp1 is larger for cases C and D, as expected from PDN, but not from the work calculation as expected.




(a) Case A (b) Case B (c) Case C (d) Case D




(e) Uncertainty term $\mathcal{U}_{robot}$ in PDN (f) Required work $W$ for a path by Valencia et al. (2011) (g) Uncertainty change

timates a realistic evaluation of the detour uncertainty as shown in Figure 4.15(e), whereas Valencia et al. estimates a smaller required work for those waypoints since they consider only geometry.

#### 4.3.3.3 Map Uncertainty (Area Coverage, $\mathcal{A}_{map}$)

As a final step in the reward calculation, we add a bias term for area coverage. The purpose of the mission is to cover a target area in a timely manner while considering SLAM's navigation performance. In other words, without an area coverage term, there will be a trivial solution to this problem—to repeatedly revisit to keep the uncertainty very small. To prevent this, the area coverage term for the $k^{\text{th}}$ waypoint is defined as the ratio of area-to-cover with respect to the target-coverage-area, where the target area is provided by the user,

$$\mathcal{A}^k_{map} = \frac{\mathcal{A}_{\text{to\_cover}}}{\mathcal{A}_{\text{target}}} = \frac{\mathcal{A}_{\text{target}} - \mathcal{A}_{\text{covered}} + \mathcal{A}^k_{\text{redundant}}}{\mathcal{A}_{\text{target}}}, \tag{4.26}$$

$$\mathcal{A}^k_{\text{redundant}} = \text{redundant coverage by revisiting} \tag{4.27}$$

$$\begin{cases} = 0, & \text{for exploration} \\ = l(\mathcal{P}^k) \cdot D > 0 & \text{for revisiting} \end{cases}. \tag{4.28}$$

Here, $l(\mathcal{P}^k)$ is the expected path length added by revisiting the $k^{\text{th}}$ waypoint, $D$ is the sensor field of view width as depicted in Figure 4.3, $\mathcal{A}_{\text{target}}$ is the pre-defined target coverage area as set in the mission planning phase, and $\mathcal{A}_{\text{redundant}}$ is the expected redundant area coverage produced by a revisiting action. This additional area is the result of multiplication of the revisit path with the sensor field of view width and has nonzero value, $\mathcal{A}^k_{\text{redundant}} = l(\mathcal{P}^k) \cdot D$.

#### 4.3.3.4 Combined Reward ($\mathcal{R}$)

To integrate these two reward terms, we introduce a weight $\alpha$ that determines the balance between pose uncertainty and area coverage. Although we maximize the reward, the formulation can be more intuitively understood when we consider each term as a penalty. The navigation uncertainty term corresponds to the penalty for SLAM, where the action with minimal uncertainty increase is preferred. The area coverage metric is the penalty in area coverage when performing an action. By taking a weighted sum of these two costs, we can evaluate the total penalty, $\mathcal{C}^k$, for a waypoint $k$. The reward is the minus of this penalty, and PDN selects an action with the largest reward, or in other words, with the minimal

penalty.

$$\mathcal{C}^k = \alpha \cdot \mathcal{U}^k_{robot} + (1-\alpha) \cdot \mathcal{A}^k_{map} \tag{4.29}$$

$$\mathcal{R}^k = -\mathcal{C}^k \tag{4.30}$$

By adjusting $\alpha$, we can change the emphasis on robot navigation uncertainty versus area coverage in the reward calculation. When $\alpha = 0$, no weight is imposed on the pose uncertainty and the algorithm tries to cover the area as fast as possible. This corresponds to an open-loop survey over the target area. On the other hand when $\alpha = 1$, full weight is on the pose uncertainty, and the robot will revisit whenever it exceeds the allowable uncertainty. Lastly, the revisiting waypoint $k^*$ is determined by maximizing the reward,

$$k^* = \text{argmax } \mathcal{R}^k = \text{argmin } \mathcal{C}^k, \tag{4.31}$$

where $k \in \{0, 1, 2, \cdots N_{wp}\}$ and $k = 0$ corresponds to the exploration action.

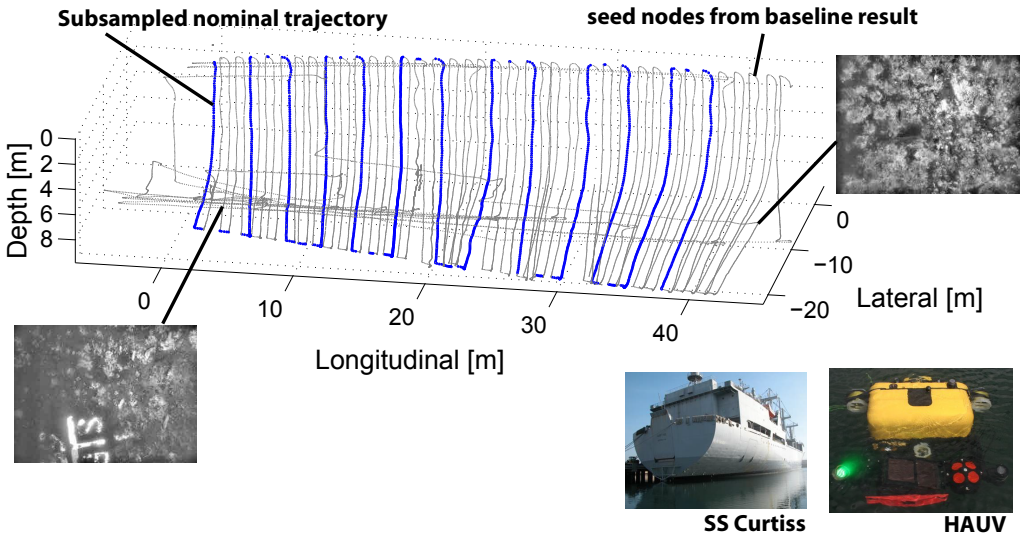
## 4.4 Results

In this section, we present an evaluation of PDN as applied to a hybrid simulation trajectory generated from real ship hull inspection data. In the first set of tests, we provide a synthetic saliency distribution on the mapping area to evaluate the performance of PDN. In the second set of tests, we present results from PDN working on two different mission profiles using real underwater hull images as input. In all test cases, PDN results are compared with other traditional preplanned survey schemes, in terms of robot uncertainty (as a measure of SLAM performance) and area coverage rate (as a measure of coverage performance).

### 4.4.1 Simulation Setup

Since there is no ground-truth available for our underwater missions, we use the baseline exhaustive SLAM result from §3.4.2 to generate a hybrid simulation with preplanned nominal trajectory. Two different types of missions, camera mission and sonar mission, are simulated from the entire data set. As shown in Figure 4.16, the mission has been preplanned with a nominal trajectory by selecting a subset of nodes from the baseline result. The gray dots show the entire set of nodes from the baseline result and the blue dots represent the subsampled nodes, the unused nodes will be used as seed nodes in the control phase. For any sequence of nodes, simulated odometry constraints can be generated using

**Figure 4.16** Simulation setup for PDN evaluation. Gray dots are the nodes from the baseline SLAM result in §3.4.2.1 on *SS Curtiss* using HAUV, which includes all of the preplanned revisit actions. The sampled nominal trajectory in blue mimics the simulated mission by sub-sampling from the baseline SLAM result. The existing preplanned revisits are removed in the nominal trajectory generation. The nodes not used in the nominal trajectory planning will be used as a seed nodes in the simulated control phase. Note that each seed node (gray) is associated with a real underwater image.

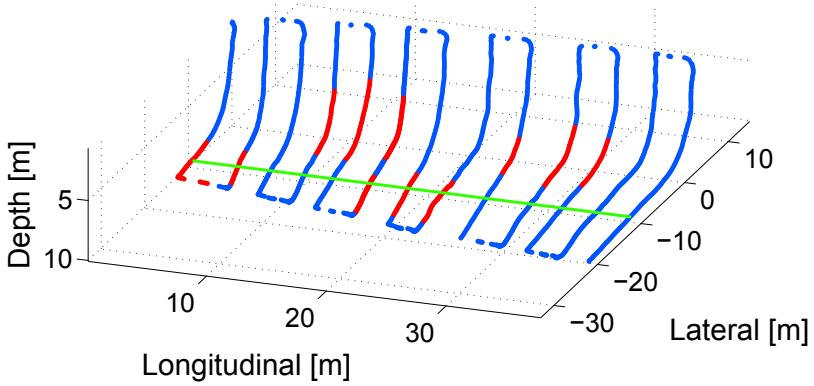



tail-to-tail operation on the baseline nodes corrupted with noise (scaled with distance traveled). The open-loop mission performs on this subsampled nominal trajectory by adding sequential odometry constraints without any revisit action.

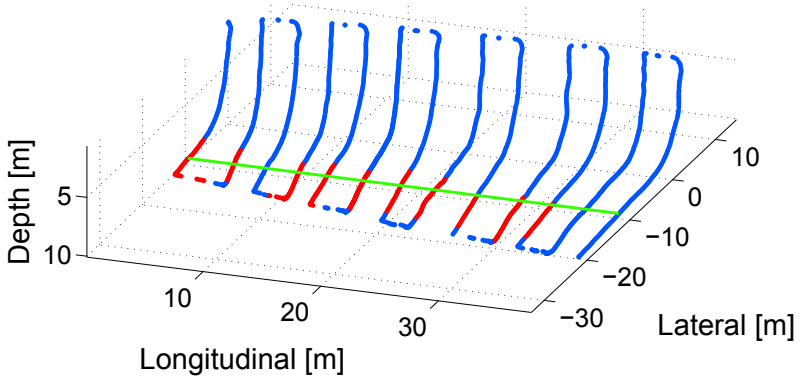
Initially, PDN also begins with the same nominal trajectory, but seeks for the next best control action by evaluating the rewards. When the optimal strategy is to explore, the robot will continue the mission on the given nominal trajectory, performing the same as the open-loop mission. On the other hand, when revisit is selected as the optimal control, we control the robot to visit the target waypoint. In a real-world mission, we can control the vehicle to revisit the target waypoint using the waypoint navigation technique as presented in Hover et al. (2012). Hover et al. control the robot to approach the target waypoint via feedback control with the target historic node assigned from the existing graph. However, in this evaluation, we need to simulate the control phase using the baseline result. The optimal path generated from point-to-point path generation (§4.3.2) is a mixture of virtual and existing nodes in the graph. In the control phase, given the next node to approach, we find the closest node from the complementary seed set to generate an odometry constraint between the current node and the selected seed node.

We also need to pursue loop-closing camera measurements during the revisit action for the two different simulation tests in the PDN evaluation. One is for PDN with a simulated

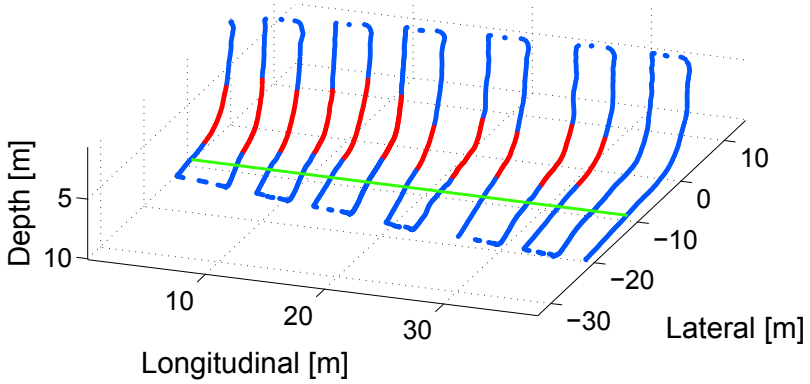
**Figure 4.17** Pose-uncertainty-only PDN with synthetic saliency distribution. The nodes are with binary saliency score where each node is marked with blue for non-salient node and red for salient node. (a) The synthetic salient regions are distributed evenly throughout the hull. (b) The salient regions are located on the bottom of the hull. (c) The salient regions are on the side of the hull. The green line on (a) indicates the path that exhaustive revisit takes, and is the same for (b) and (c)




(a) Evenly distributed salient region




(b) Salient regions distributed on the bottom




(c) Salient regions distributed at the side

synthetic saliency map, and the other is with the online saliency map generated from the real images associated with the nodes. For the simulated synthetic saliency map, the actual saliency score for an image is ignored to test PDN's performance on the controlled saliency distribution. In this case, the camera measurements are determined from a random experiment using the successful link probability, $P_L(l = 1; S_c, S_t)$, generated in §4.3.3.1. The simulated arrival of a successful camera measurement can be considered as a coin toss where the coin is biased with a probability of landing "HEAD" equal to $P_L(l = 1; S_c, S_t)$[7]. For each link hypothesis, we run this coin tossing experiment. If the coin toss comes out "HEAD", the link is successful and a simulated camera measurement is generated from ground-truth, corrupted with noise. On the other hand, when testing with real images, the proposed pair is fed into the actual camera registration engine (§2.3). In other words, when a control action leads the vehicle through any sequence of nodes, we can provide the real image of the closest node to the camera engine and let the engine produce loop-closing camera measurements. Note that all of the gray nodes in Figure 4.16 are associated with real underwater images.

In all cases of evaluation, the PDN result is compared against two typical survey strategies. One pattern is an open-loop control that follows the given nominal trajectory without any revisiting. The other survey pattern is to preplan some deterministic revisit actions during the preplanning phase, which are aimed at achieving any possible loop-closures. This deterministic revisit strategy is typical of underwater vehicle operations, and is passively preplanned or executed by a human pilot. In this work, we call this preplanned regular revisit "exhaustive revisit". In the exhaustive revisit scenario, the vehicle is controlled to come back to a waypoint in every other track-line for possible loop-closure, regardless of the actual feature distribution in the environment.

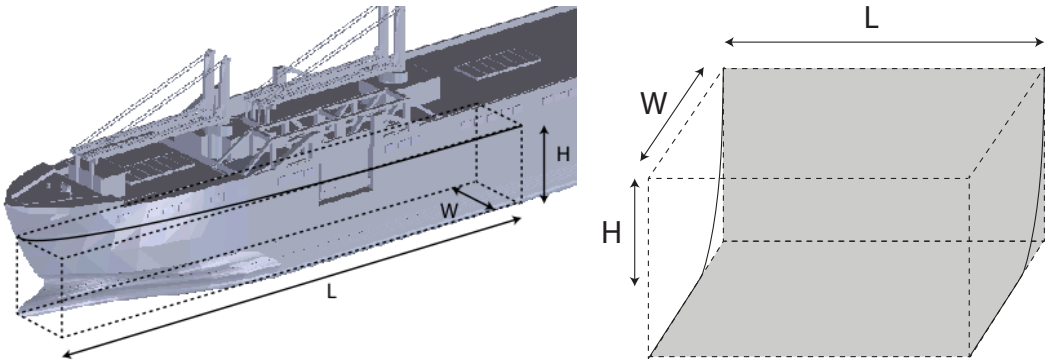
### 4.4.2 PDN Parameters

Three PDN parameters need to be determined before the mission starts, (*i*) allowable covariance determinant, (*ii*) target coverage area, and (*iii*) weight factor $\alpha$. The allowable covariance is dominated by $xy$ positional uncertainty in our application because depth is bounded with absolute measurements. For the desired limit of positional uncertainty $\sigma_{xy,allow}$, the allowable covariance bound is computed as

$$|\Sigma_{\text{allow}}| = \sigma^2_{xy,allow} \cdot \sigma^2_{xy,allow} \cdot \sigma^2_d \cdot \sigma^2_r \cdot \sigma^2_p \cdot \sigma^2_h,$$

[7]The implementation for this coin toss is done by using Matlab's random number generator. We first generate a random number between 0 and 1 from a uniform distribution, then compare this to $P_L(l = 1; S_c, S_t)$. If the random number exceeds $P_L(l = 1; S_c, S_t)$, it returns "HEAD" (success), and "TAIL" (fail) otherwise.

**Figure 4.18** Target coverage area calculation. The area of interest is approximated by a bounding box, where the target coverage area is computed using the vessel's length $L$, width (half of the beam) $W$, and draft $H$.




(a) Box approximation (b) Target coverage area

where depth uncertainty of $\sigma_d = 0.01\text{m}$ and attitude uncertainty of $\sigma_r = \sigma_p = \sigma_h = 0.1^\circ$ (roll, pitch and heading, accordingly) are used. In this evaluation, we specify $\sigma_{xy,allow}$ to intuitively set the navigation uncertainty bound. Next, the target coverage area is computed using the vessel's longitudinal length $L$, width (half-beam) $W$, and draft $H$ (Figure 4.18)[8]. When the actual model is available, an accurate target area can be provided. In this evaluation, the area for a bounding box of $L \times W \times H$ is considered to be the target coverage area,

$$\mathcal{A}_{target} = L \times (W + H). \tag{4.32}$$

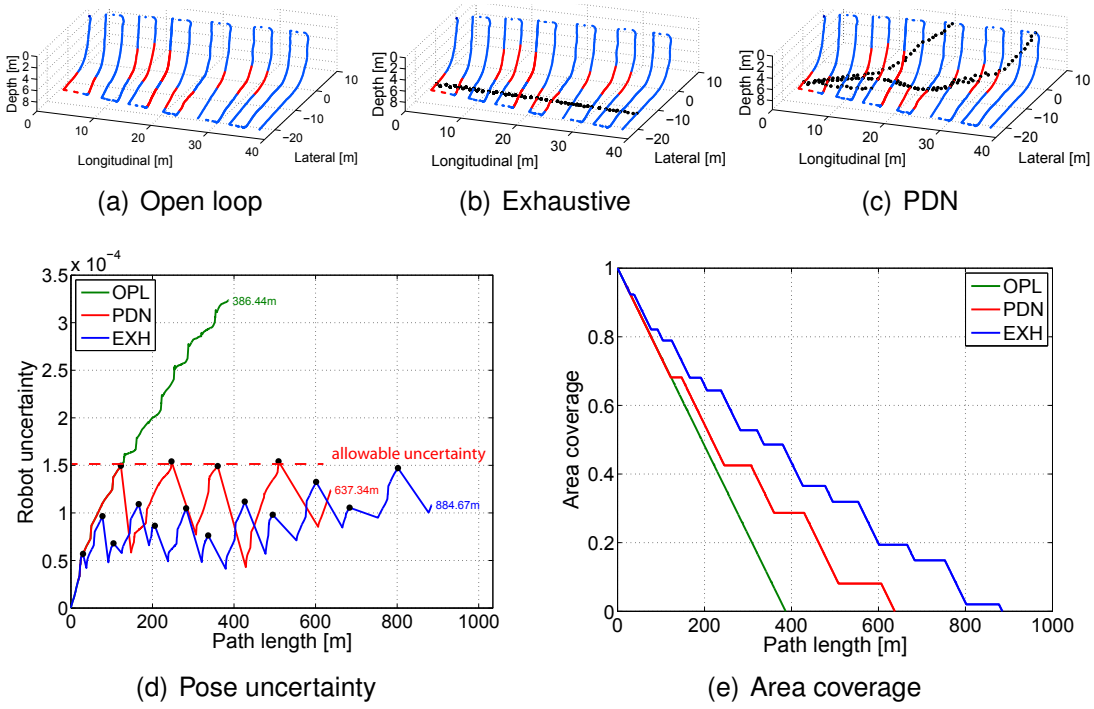
Lastly, the weight factor $\alpha$ needs to be specified. In the evaluation with real world data, the emphasis factor is set to be $0.75$ to impose bias toward the navigation uncertainty. The effect of this parameter will be investigated later in this chapter in a set of controlled studies.

### 4.4.3 PDN with Synthetic Saliency Map

The first set of tests are with a synthetic saliency map imposed over the area. Before testing with real underwater images, we first test PDN on saliency-ignored SLAM to validate the algorithm on the sonar-type mission. For this synthetic mission, the allowable covariance and target area are set to $\sigma_{xy,allow} = \pm 0.25$ m and $1200$ m$^2$ ($L = 40$ m, $W = 20$ m, and $H = 10$ m), respectively. First, we will test PDN with full weight on the pose uncertainty to verify that the control on pose uncertainty is valid. Then, the effect of the weighting factor $\alpha$ will be examined.

[8]Beam refers to the width of the hull and draft is the vertical distance between waterline and the vessel's bottom (Rossell et al., 1941).

**Figure 4.19** PDN results for evenly distributed saliency map. Shown are pose uncertainty and area coverage for open-loop, exhaustive revisit and PDN. (a), (b) and (c) are the final trajectory of the robot. (d) depicts the $6^{th}$ root of determinant of the robot pose with respect to the path length for open-loop (green), exhaustive revisit (blue) and PDN (red), where the black dots indicate points when revisit occurred. (e) shows the ratio of the remaining area to cover with respect to the path length. The uncertainty is not bounded for open-loop but results in the fastest area coverage. Exhaustive revisit has the longest path length and slowest area coverage due to the repeated revisit, but is capable of bounding the uncertainty very small when there are sufficient loop-closures. PDN performs in between open-loop and exhaustive revisit in the path length and area coverage, but allows for full control of the robot's uncertainty level. In total, 9 revisits are executed from the preplanned exhaustive mission, whereas 4 revisits are performed in PDN.

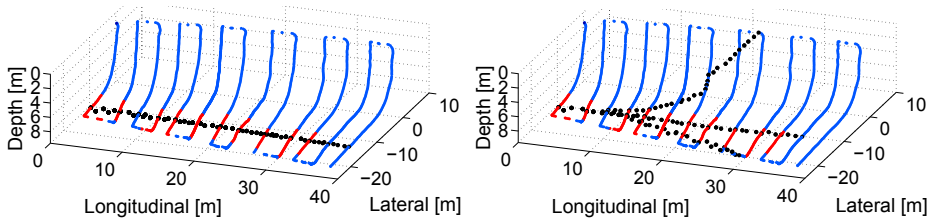



## Pose-Uncertainty-Only PDN

We first validate the pose-uncertainty-only PDN (i.e., $\alpha = 1$) with a synthetic sonar mission. The PDN action is verified for three different types of saliency distributions as shown in Figure 4.17, comparing it to exhaustive revisit control and open-loop survey.

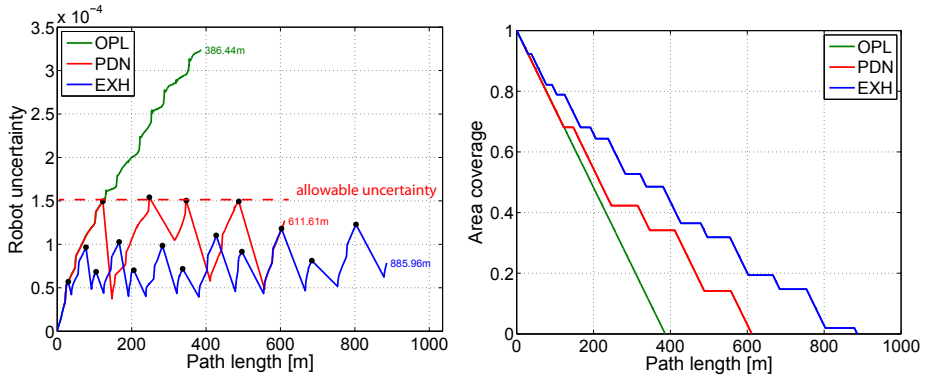
For the exhaustive revisit, the robot is commanded to revisit a point on the first track-line in every other track-line it travels. In this test, the exhaustive revisit happens on a line on the bottom of the hull, as marked in green in Figure 4.17. Because this repeated visit is preplanned without knowing the actual visual feature distribution in the environment, we assign the same exhaustive revisit control for all three cases, as shown in Figure 4.17. When salient regions are evenly distributed (Figure 4.17(a)), the exhaustive revisit will

**Figure 4.20** PDN results for biased saliency maps. Pose uncertainty and area coverage graph are compared for two biased saliency regions for open-loop (green), exhaustive revisit (blue) and PDN (red), where the black dots indicate points when revisit occurred. Open-loop and PDN perform equally well to the evenly distributed salient region case. However, exhaustive revisit strongly depends on the spatial distribution of feature-rich areas.

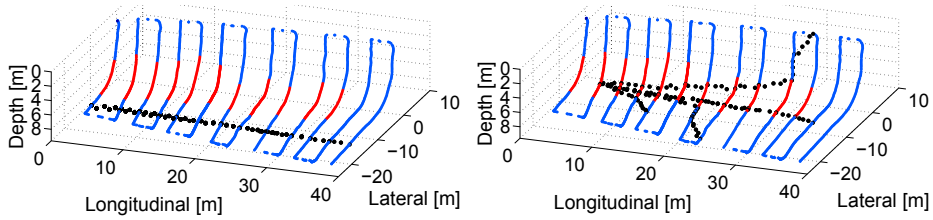



(a) Exhaustive (b) PDN

**Good Saliency Distribution for Exhaustive Revisit**

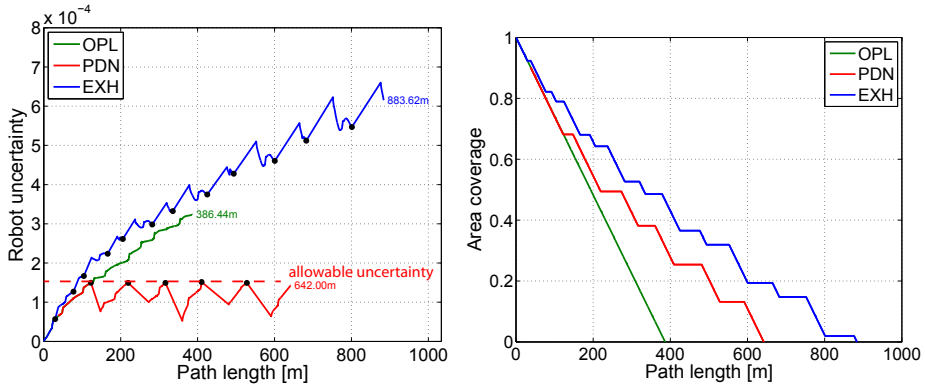



(c) Pose uncertainty (d) Area coverage




(e) Exhaustive (f) PDN

**Poor Saliency Distribution for Exhaustive Revisit**




(g) Pose uncertainty (h) Area coverage

pass through some salient regions for most of the action. However, when we have a biased feature distribution, the exhaustive revisit path can be always on the salient regions (Figure 4.17(b)) or never pass through the salient regions (Figure 4.17(c)).

Figure 4.19 shows the result when salient regions are evenly distributed. Figure 4.19(a), Figure 4.19(b) and Figure 4.19(c) are the final trajectory of the robot. The change of the robot pose uncertainty is plotted in Figure 4.19(d) and the area coverage rate is shown in Figure 4.19(e). Figure 4.19(d) depicts the robot pose uncertainty ($\sqrt[6]{|\Sigma_{rr}|}$) versus the path length for open-loop (green), exhaustive revisit (blue) and PDN (red). Figure 4.19(e) shows the ratio of the remaining area to cover with respect to the path length, where the black dots indicate points when revisit occurred. The uncertainty is not bounded for open-loop, but results in the fastest area coverage in Figure 4.19(e) because the open-loop does not execute any revisit actions. Exhaustive revisit has the longest path length and slowest area coverage due to the regular repeated revisits, but is capable of bounding the uncertainty very tightly when there are sufficient loop-closures. In some cases, this preplanned regular revisit leads the robot to revisit regardless of the localization performance, for example, it executes an unnecessary revisit even when the current robot pose uncertainty is small, or it commands no control even if the localization is getting uncertain. On the other hand, PDN performs in between open-loop and exhaustive revisit in the path length and area coverage, enabling full control over the uncertainty level of the robot and keeping it under the allowable user-defined uncertainty level.

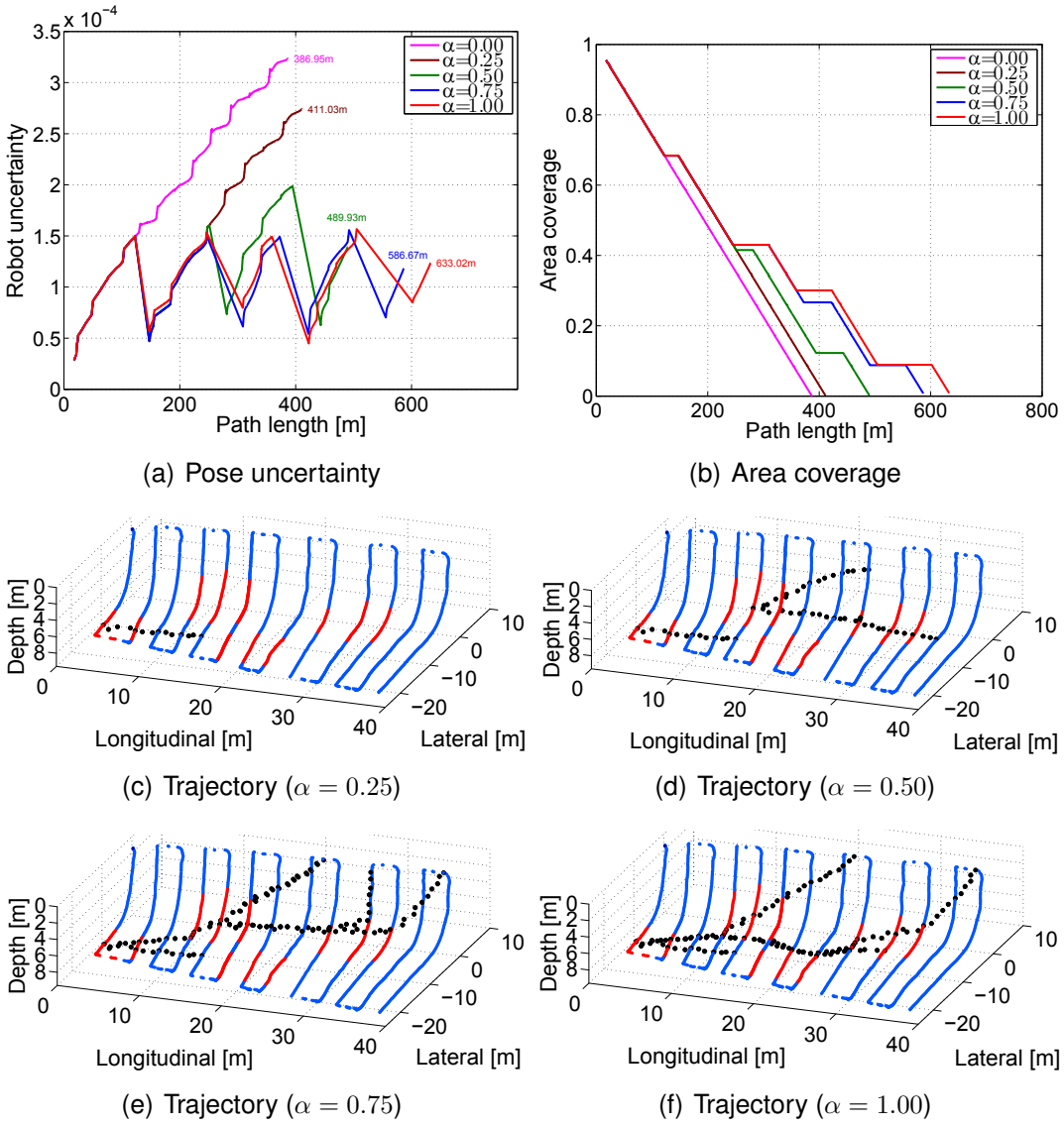
Next, we test with cases where the feature-rich distribution is biased to show how the preplanned exhaustive revisit succeeds and fails depending on the saliency distribution. When all of the exhaustive revisit paths land on the salient region, the likelihood of obtaining loop-closure during the revisit is higher, and the exhaustive revisit achieves tightly bounded uncertainty for the robot pose. On the other hand, when none of the revisit paths are on salient regions, as in the case of Figure 4.20(e), the algorithm basically performs worse than open-loop. Without meaningful loop-closures on the revisit, the control just increases the overall path length and slows coverage rate, as can be seen in Figure 4.20(g) and Figure 4.20(h). Unfortunately, the salient region distribution cannot be known a priori when the preplanning takes place. Note that for both cases, the total path length and the area coverage rate stays the same for the exhaustive revisit since it is preplanned. On the other hand, from PDN's point of view, there is no difference between all three cases, resulting in a consistent performance on uncertainty bounding and area coverage.

### Effect of $\alpha$ in PDN

In §4.3.3.3, the weighting factor $\alpha$ introduced a bias term for the area coverage. To examine the effect of $\alpha$, we ran PDN on the evenly distributed saliency map, varying the value from $0.00$ to $1.00$ incrementally by $0.25$.

The parameter $\alpha$ indicates how much weight is on the pose uncertainty versus area

**Figure 4.21** PDN-aided SLAM trajectory behavior with respect to $\alpha$. When $\alpha = 0$, PDN performs open loop control. When $\alpha = 1$, PDN reacts instantly once the pose uncertainty exceeds the allowable uncertainty level. (a) shows the change of pose uncertainty with respect to $\alpha$. The uncertainty is most well bounded when $\alpha = 1$ and relaxed as $\alpha$ decreases. (b) shows the area coverage rate in terms of $\alpha$. $\alpha = 0$ shows the fastest coverage rate, which is slowed as $\alpha$ increases and weights the pose uncertainty more ((c)-(f)). PDN-aided SLAM trajectories for different values of $\alpha$.




(a) Pose uncertainty (b) Area coverage

(c) Trajectory ($\alpha = 0.25$) (d) Trajectory ($\alpha = 0.50$)

(e) Trajectory ($\alpha = 0.75$) (f) Trajectory ($\alpha = 1.00$)

coverage. When $\alpha = 0$, PDN does not assign importance on the pose uncertainty, and the framework works the same as open-loop. When $\alpha = 1$, full weight is to the pose uncertainty, and PDN tries to reduce the uncertainty once it reaches the allowable navigation uncertainty threshold. In other words, the effect of $\alpha$ is to delay the execution of revisiting by PDN. The uncertainty is most well bounded when $\alpha = 1$, and relaxed as $\alpha$ decreases. For area coverage, $\alpha = 0$ shows the fastest coverage rate, which slows as $\alpha$ increases (weights pose uncertainty more) as can be seen in Figure 4.21(b). The effect of $\alpha$ shown in Figure 4.21, which presents several SLAM trajectories with different weight factors. As the weight on pose uncertainty increases (from $0.25$ to $1.00$), PDN tends to revisit the furthest waypoint more often to result in larger loop-closures. When this weight is small, PDN allows the pose uncertainty to increase in order to cover the area fast. In this case, revisit waypoints are likely to be nearby positions so as not to delay the area coverage performance.

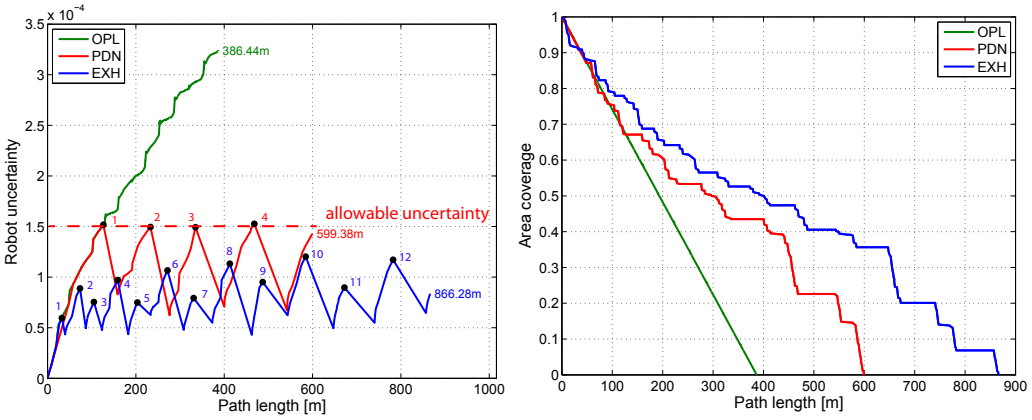
### 4.4.4 PDN with Real Image Data

We now evaluate PDN for saliency-informed SLAM using real underwater images for the two types of mission profiles, sonar and camera. As described in §4.4.1, the saliency map is generated and updated online from the real underwater images that are available from the baseline result. Using the pair of real-world images, the saliency score and camera registration engine are applied as in the normal saliency-informed SLAM process with the local saliency threshold of $S_L^{\mathsf{min}} = 0.4$. A weight factor of $\alpha = 0.75$ is selected in PDN to impose a biased weight on the pose-uncertainty rather than area coverage.

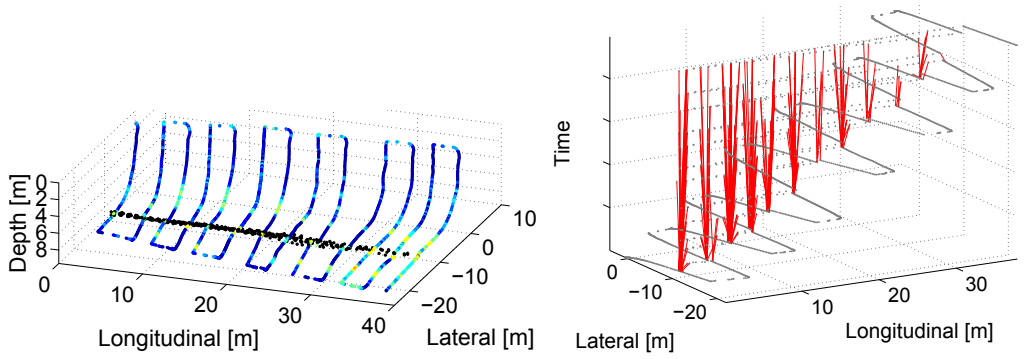
Similar to the synthetic saliency case, the uncertainty and area coverage graph for PDN is compared with open-loop (OPL) and exhaustive (EXH) revisit. Based upon the knowledge of the saliency distribution in the baseline result, we preplanned the exhaustive revisit path to be laid over the salient band to provide the best possible case to be compared with PDN. Because the exhaustive revisit is intentionally planned over the salient region, the resulting graph of exhaustive revisit shows the maximum SLAM performance—maintaining low uncertainty, but producing an exceeding number of revisits and longer path length. The exhaustive revisit includes 9 revisits in the sonar mission and 47 revisits in the camera mission.

Uncertainty change and the area coverage rate are presented in Figure 4.22(a)-(b) and Figure 4.23(a)-(b) for both types of missions. In case of the sonar mission with real images, the same allowable covariance with $\sigma_{xy,allow} = \pm 0.25$ m and target area of $1200$ m$^2$ are used as in the synthetic saliency case. For the camera mission, the same allowable covariance but different target area of $900$ m$^2$ ($L = 30$ m$, W = 20$ m, and $H = 10$ m) are

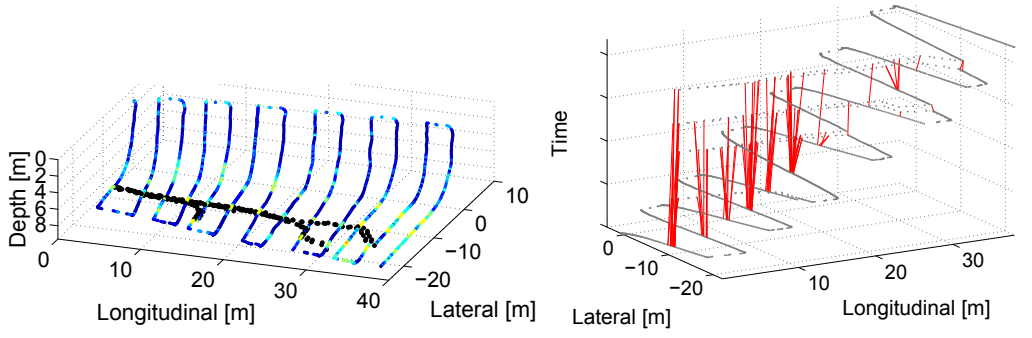
**Figure 4.22** PDN for saliency-informed SLAM on a sonar mission. Similar to the synthetic saliency case, (a) and (b) show the pose uncertainty and area coverage with respect to the path length with black dots on the revisit points annotated with the revisit count. In trajectories (c) and (e), nodes from nominal trajectory are color-coded by their saliency level from the real images and nodes in revisit path are shown as black dots. The exhaustive revisit was preplanned over the salient band, however, PDN is able to find this same optimal path to follow as shown in (e). In the time elevation graphs ((d) and (f)), PDN shows a similar number of successful loop-closures on salient regions as compared to the exhaustive revisit.




(a) Pose uncertainty (b) Area coverage

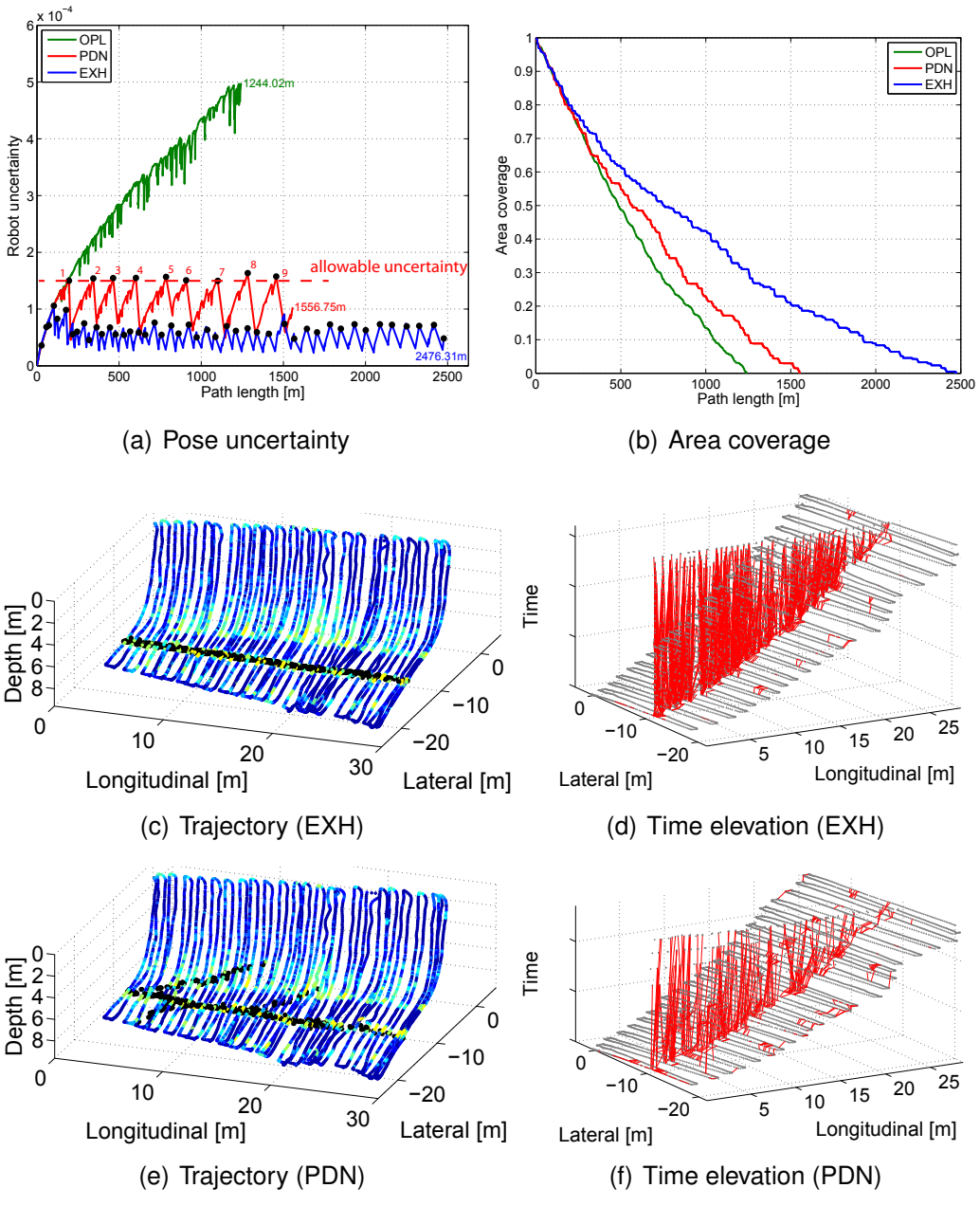



(c) Trajectory (EXH) (d) Time elevation (EXH)




(e) Trajectory (PDN) (f) Time elevation (PDN)

**Figure 4.23** PDN for saliency-informed SLAM on a camera mission. Similar to the synthetic saliency case, (a) and (b) show the pose uncertainty and area coverage with respect to the path length. In trajectories (c) and (e), nodes are color-coded by their saliency level from the real images. The exhaustive revisit was preplanned over the salient band, however, PDN is able to find this same optimal path to follow as in (e). In the time elevation graphs ((d) and (f)), PDN shows a comparable number of successful loop-closures to the exhaustive revisit. Note that the total number number of loop-closures in the exhaustive revisit case is significantly larger due to the 47 preplanned revisit actions versus PDN's 9 revisits.




(a) Pose uncertainty

(b) Area coverage

(c) Trajectory (EXH)

(d) Time elevation (EXH)

(e) Trajectory (PDN)

(f) Time elevation (PDN)

used. As shown in Figure 4.22(e) and Figure 4.23(e), PDN followed trajectories to obtain expected visual loop-closures to reduce the uncertainty whenever it exceeded the allowable covariance bound. Specifically, exhaustive revisits in the camera mission result in 47 revisit actions with twice the total path length of the nominal trajectory. Assuming a constant speed for the vehicle, this exhaustive revisit strategy would double the overall mission time. In this camera mission, note that the number of revisits by PDN (9) is substantially smaller than the exhaustive revisit case.

For both missions, PDN presents a result with less number of revisits while maintaining full control on the uncertainty level, and still achieving the important loop-closures. The loop-closing camera measurements are clearly illustrated in the time elevation graph of Figure 4.22(d)-(f) and Figure 4.23(d)-(f). The red lines in the graph depict the camera measurements made by the loop-closures. In the sonar mission, because there is no possible camera measurement between adjacent track-lines, all of the camera measurements in the graph are from the revisit action. As can be seen in the time elevation graphs, PDN obtained a similar number of loop-closures as compared to the exhaustive revisit case.

## 4.5 Conclusion

In this chapter, PDN was presented as an active SLAM approach that takes into account area coverage. To efficiently explore the target area of interest, PDN is proposed as a way to balance between exploration and pose-uncertainty by evaluating rewards that consider both SLAM performance and area coverage. A weighting factor, $\alpha$, provides a control between this balance. The pose uncertainty term is evaluated using the saliency metric to estimate plausible camera measurements. The simulated trajectories with both synthetic and real underwater images were tested to evaluate PDN's performance and to prove its ability to plan an optimal path while controlling the uncertainty level, and also achieve a balanced area coverage rate.

# CHAPTER V

# Conclusion

This thesis proposed an integrated approach toward robotic navigation and exploration for autonomous robot missions. SLAM and path planning have traditionally been considered as two separate problems, each assuming some prior information from the other. This thesis started from the viewpoint of SLAM and presented a metric for visual saliency that could be used with geometric information to improve loop-closure performance. This saliency-informed SLAM result was then combined with planning to lead the robot autonomously along trajectories that yielded better SLAM results and survey area coverage. Experiments using real underwater mission data and simulations were provided from several different vessels to evaluate the reported algorithms.

## 5.1 Contributions

- This thesis presented a real-time visual SLAM implementation that has been successfully applied on several real-world autonomous ship hull inspection.
- This thesis developed two novel measures of visual saliency and presented a way to leverage them within a visual SLAM framework. Local and global saliency. Local saliency was shown to indicate the texture-richness of a scene, whereas global saliency was shown to report the rarity of the scene within the dataset.
- This thesis presented a link proposal algorithm that considered the camera measurement likelihood using visual saliency. This saliency-informed SLAM approach improved performance by reducing the number of less meaningful nodes and culling non-plausible link proposals.
- A solution for concurrent SLAM and planning for the robotic area coverage problem was presented. No prior knowledge of the environment is needed because the

planning is performed online in the SLAM belief space. The planned path aids robot navigation by bounding the SLAM pose uncertainty while efficiently completing the target area coverage.

## 5.2 Future Work

Some suggested areas of future work are:

**• Model-based global saliency.**

The robot's motion is critical in updating the global saliency score, since this score is related to visual rarity. In the current framework, the proximity to the other nodes in the graph is checked in the global saliency statistics update. Instead of learning the probability of the appearance of a word from the statistics, a model-based approach can provide a general solution when combined with location information. The challenge in modeling global saliency is modeling the probability of a word to be found in a document, $p(w)$. Currently inverse document frequency (idf) is used as an approximation of this probability,

$$idf = \log \frac{N}{n_w} \sim \log p(w), \tag{5.1}$$

where $N$ is the total number of documents (images) in the dataset, and $n_w$ is the number of documents containing the word $w$. Instead of checking the motion when updating $N$ and $n_w$, modeling the probability of a word as a conditional probability given a sequence of images and the geometrical location, $p(w|I_{0,\cdots t}, \mathbf{x})$, will improve the result without a need for checking the motion explicitly. This will provide a generalized way to model $p(w)$ and the global saliency score.

**• Robust link proposals in appearance space.**

When dead-reckoning is not accurate, navigation error drifts over time and the current geometry-based link proposal strategy is limited. Furthermore, as in the case of the kidnapped robot problem (Choset, 2005), geometric link proposal might not even work. Two issues must be addressed for robust link proposal. First, the information gain based link proposal is significantly more effective than just using the mean. However, the information gain is computed based on the current mean of the belief space. In other words, when the current navigation is uncertain, the variance of the information gain is large, which is not captured in the expected information gain evaluation. The current saliency-informed

SLAM approach produces a weighted information gain to impose a bias toward two similar highly salient scenes. However, this weighted information gain does not explicitly evaluate the information gain variance.

Second, using an appearance-based metric would be essential when the navigation uncertainty is significantly large. Konolige et al. (2009) presented merging two maps in appearance space by initially connecting them with large uncertainty and then recognizing the scene in the other map to reduce the uncertainty. Recent success in appearance-based space SLAM (Angeli et al., 2008; Cummins and Newman, 2008; Kawewong et al., 2010) and place recognition can enhance SLAM performance by proposing visually feasible loop-closures. Robustness of the link proposal could be improved if the variance of information gain is considered together with their visual similarity score.

• **Ability to recover from failure mode.**

False image pair registrations can deteriorate the SLAM result. Undoing the measurement is possible in many back-end SLAM approaches, however, the false measurement must be identified first. Snderhauf and Protzel (2011) introduced another term indicating the activeness of a loop-closure additional to the uncertainty related to the measurements. More recently, Olson and Agarwal (2012) presented a back-end solution to robust data association using a minmax mixture model. Although validation at the front-end level is alleviated in this approach, intelligently determining the suspicious measurement to report to the SLAM back-end would be beneficial toward life-long SLAM. The robot can make a mistake while navigating, and should be able to detect the most questionable measurement and localize accordingly.

# APPENDICES

# APPENDIX A

# Implementation Details of In-water Hull Inspection

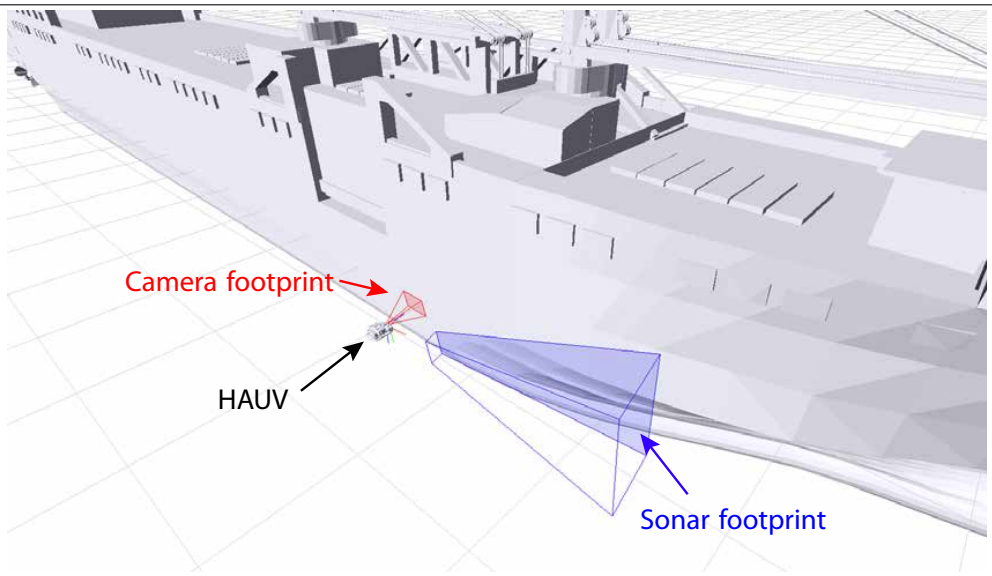
This appendix provides additional details of the in-water hull inspection project conducted in collaboration with the University of Michigan, Massachusetts Institute of Technology (MIT), and Bluefin Robotics.

Typical methods for port security and ship hull inspection require either deploying human divers (Mittleman and Wyman, 1980; Mittleman and Swan, 1993), using trained marine mammals (Olds, 2003), or piloting a ROV (Harris and Slate, 1999; Lynn and Bohlander, 1999; Carvalho et al., 2003; Negahdaripour and Firoozfam, 2006). Autonomous vehicles have the potential for better coverage efficiency, improved survey precision and overall reduced need for human intervention. As early as 1992, the Naval community had identified a need for developing such systems (Bohlander et al., 1992).

Recently, research effort in this area has resulted in the development of a number of autonomous hull inspection platforms (Trimble and Belcher, 2002; Vaganay et al., 2005, 2006; Menegaldo et al., 2009). Negahdaripour and Firoozfam (2006) developed underwater stereo-vision as a means of navigating an ROV near a hull; they used mosaic-based registration methods and showed early results for limited pool and dock trials. Ridao et al. (2010) reported on the closely related task of automated dam inspection using an autonomous underwater vehicle; their solution uses USBL and DVL-based navigation online during the mapping phase, followed by an offline image bundle adjustment phase to produce a globally-optimal photomosaic and vehicle trajectory. Walter et al. (2008) reported the use of an imaging sonar for feature-based SLAM navigation on a barge and showed results for offline processing using manually-established feature correspondence. More recently, this work was significantly extended by Johannsson et al. (2010) to work in real-time and to perform automatic registration of sonar imagery along the hull.

In parallel to these efforts, we have, since 2007, collaborated with the authors of Johannsson et al. (2010) and with Bluefin Robotics, Inc. on an Office of Naval Research sponsored project for autonomous hull inspection. Our contribution has been to develop a real-time visual SLAM capability for hull-relative navigation in the open areas of the hull. Through collaboration with our project partners, we have developed an integrated real-time

**Figure A.1** Hull inspection using sonar and camera. The sensor field-of-view for the imaging sonar and monocular camera during open-area hull inspection are depicted. Note that the two sensors concurrently image different portions of the hull. The footprint of the DVL's four beams is approximately the same as that shown for the camera.



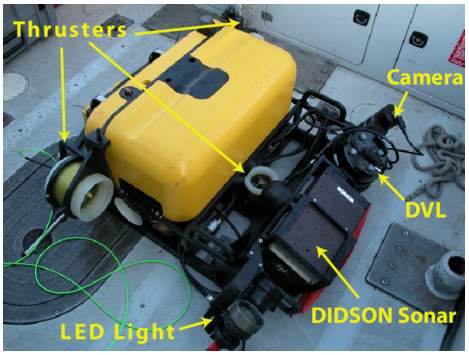

SLAM system for hull-relative navigation and control that has been recently demonstrated on the Bluefin Robotics HAUV (pronounced "H-A-U-V"). Specifications of the current generation vehicle design are documented in Vaganay et al. (2009), and an overview of our integrated work in perception, planning and control is presented in Hover et al. (2012).

## A.1 Hovering Autonomous Underwater Vehicle

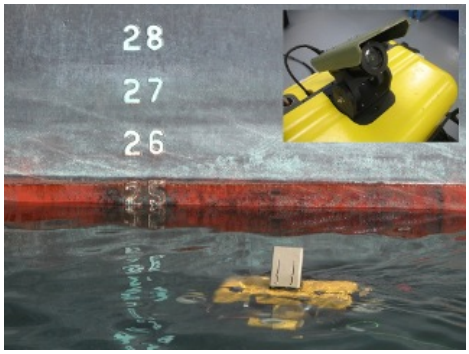
For the autonomous hull inspection project, we use the Bluefin Robotics, Inc. Hovering Autonomous Underwater Vehicle (HAUV) (Vaganay et al., 2009) (Figure A.2). This vehicle was developed with Office of Naval Research sponsorship for explosive ordnance disposal (EOD) inspection on the hull, and is currently in production for the U.S. Navy (Weiss, 2011). For navigation, the standard vehicle is equipped with a $1200$ kHz RDI Doppler velocity log (DVL), Honeywell HG1700 IMU, and Keller pressure sensor for depth. For inspection the vehicle is equipped with a $1.8$ MHz DIDSON imaging sonar (Belcher et al., 2002). Additionally, in collaboration with Bluefin Robotics, we have integrated a fixed-focus, monochrome, Prosilica GC1380 12-bit digital-still camera and $520$ nm (green) LED light source for optical imaging.

We run the vehicle in either one of two different camera configurations during an open-area survey of the hull: an "underwater" mode configuration (Figure A.2(a)) or a "periscope" mode configuration (Figure A.2(b)). In both modes of these, the vehicle nav-

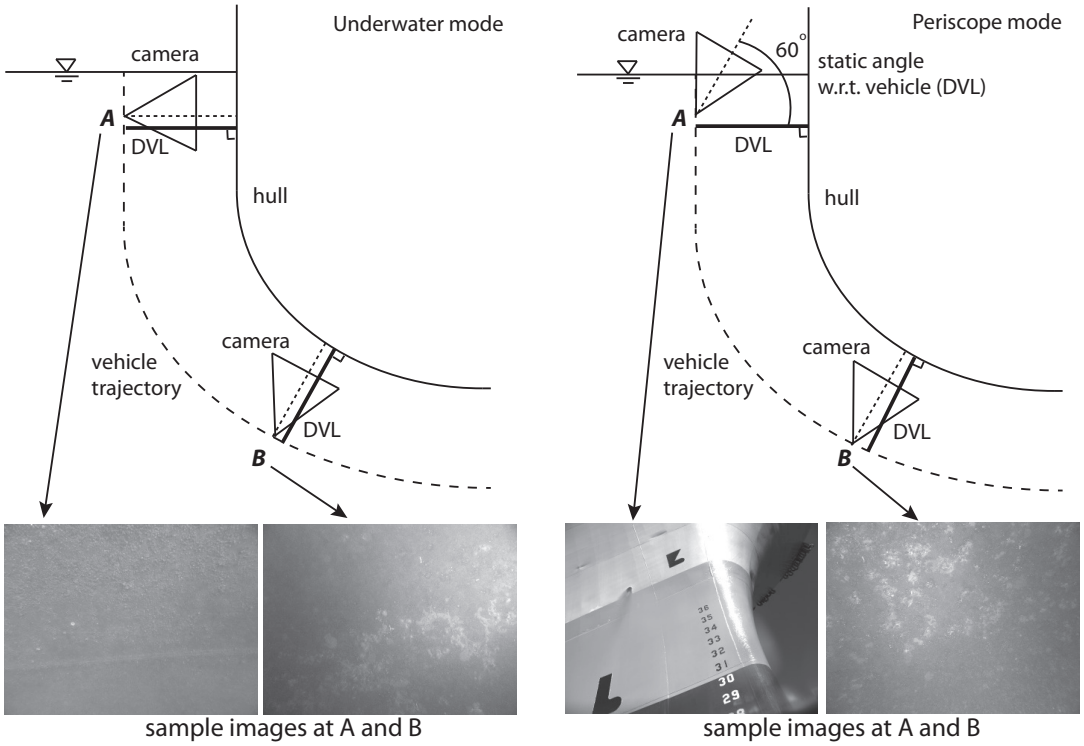
**Figure** A.2 Two different camera configurations for the HAUV: (a) "underwater" mode and (b) "periscope" mode . Sample imagery and illustration of the camera field-of-view in the two respective modes is shown in (c). The camera field-of-view is depicted by the triangle whereas the orientation of the DVL is plotted as a thick black line. The DVL is pitch servoed to always remain approximately orthogonal to the local hull curvature. In the underwater mode, the camera is servoed on this same axis, whereas in the periscope mode it has a fixed pitch of approximately 60 degrees. Sample images are depicted for each configuration, where $A$ and $B$ denote near-waterline and near-keel positions.




(a) HAUV in underwater mode




(b) HAUV in periscope mode




(c) Field-of-view of the camera and DVL in each mode

**Table A.1** Specifications of major HAUV components.

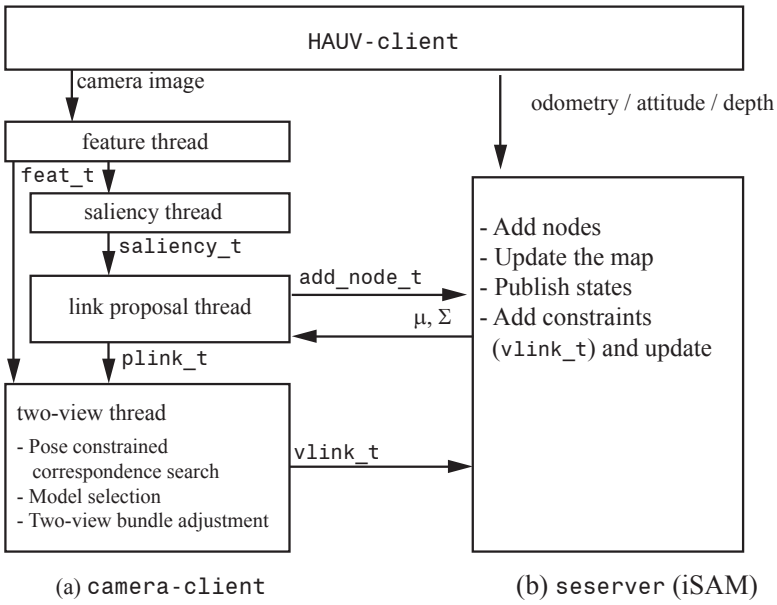
| | |
|---|---|
| **Dimensions** | 1 m × 1 m × 0.45 m (L × W × H ) |
| **Dry weight** | 79 kg |
| **Battery** | 1.5 kWh lithium-ion |
| **Thrusters** | 5, rotor-wound |
| **IMU sensor** | Honeywell HG1700 |
| **Depth sensor** | Keller pressure |
| **Imaging Sonar** | Sound Metrics 1.8 MHz DIDSON |
| **Doppler velocity** | RDI 1200 kHz Workhorse;<br>also provides four range beams |
| **Camera** | 1380 × 1024 pixel, 12-bit CCD |
| **Lighting** | 520 nm (green) LED |
| **Processor** | 650 MHz PC104 |
| **Optional Tether** | 150 m long, 5 mm dia. (fiber-optic) |

igates on the hull using DVL derived odometry collected at a constant standoff distance maintained using DVL measured range. The DVL is pitch-actuated so that it is always approximately orthogonal to the local hull surface. Our normal mode of operation is the underwater camera mode whereby the camera is mounted on the same servo tray as the DVL, and, therefore, also actuated to keep an approximately orthogonal view of the hull surface (Figure A.2(c)). This configuration allows for maximum underwater field-of-view of the hull from the camera throughout the mission, with a sparse scene depth prior provided from the four DVL range beams.

The second configuration we run is the periscope mode, which is helpful when water visibility is poor due to high turbidity. In this mode, the camera is mounted on top of the HAUV at a fixed angle of ∼60 degrees pointing upward. This configuration allows the camera to protrude above the waterline whenever the vehicle broaches the surface, and thereby captures imagery of the above-water superstructure of the ship as depicted in Figure A.2(b). This configuration allows the camera to provide meaningful hull-relative SLAM measurements regardless of the water turbidity, though its utility underwater in clear water is diminished as the camera field-of-view is not ideal for viewing the entire hull surface (Figure A.2(c)). When running in this configuration, the camera intrinsics are calibrated for both in-water and in-air (the appropriate calibration is chosen based upon pressure sensor depth).

In both configurations, underwater and periscope, the camera runs at 2–3 Hz, which typically provides approximately 50–70% sequential frame-to-frame overlap at nominal vehicle speeds. For cross-track overlap, missions are typically designed in either one of two ways. The first way is for 100% imaging sonar coverage, which means that since the camera field-of-view footprint is much smaller than the imaging sonar (Figure A.2(c))

**Figure A.3** Real-time SLAM software architecture. Depicted is the server-client software architecture using iSAM. The shared estimation server, `seserver`, listens for add node message requests, `add_node_t`, from the `camera-client`. Extracted features, `feat_t`, are published by the feature thread. The saliency thread subscribes to these `feat_t` messages and computes a visual saliency score, which gets published as a `saliency_t` message. This score is used in the link thread to determine node addition as well as link proposal events. Proposed link candidates are published as `plink_t` events, which the two-view thread then attempts to register. If successful, the camera thread then publishes the 5-DOF camera constraint as a verified link message, `vlink_t`, which then gets added to the graph by `seserver`.

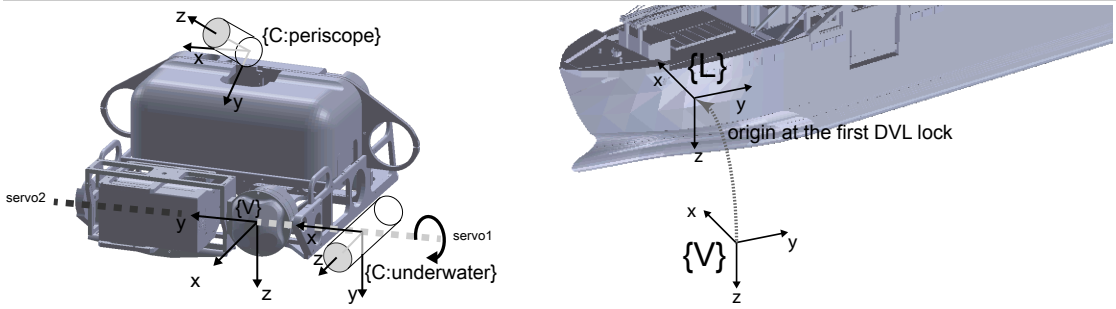



there is no camera-to-camera cross-track overlap on adjacent track-lines. In this scenario, the camera is primarily used for closing large-loops in our SLAM navigation framework; results for this configuration are reported in Hover et al. (2012). In the second operational scenario, we use the camera to provide 100% sensor coverage of the hull and thereby aim for about 30% side-to-side cross-track overlap between adjacent track-lines. In both cases the camera is used as a navigation sensor and for inspection.

## A.2 SLAM Software Architecture

Our real-time SLAM implementation is based on a publish/subscribe software architecture using the open-source LCM library (Huang et al., 2010) for inter-process communication. We run iSAM as a shared server process and each sensor client process can independently publish measurement constraints to add to the graph; Figure A.3 depicts an architectural block-diagram. The server process subscribes to messages from the HAUV

**Figure** A.4 Local, vehicle and sensor coordinate frames. (a) Camera sensor transformation with respect to the vehicle frame ($\{V\}$) for two different configurations, underwater and periscope. (b) Local frame ($\{L\}$) originates from the first lock of DVL measurement onto the hull.




(a) Vehicle and camera frames (b) Vehicle and local frames

vehicle client process to add DVL odometry constraints, absolute roll/pitch attitude measurements (from the IMU), and pressure depth observations.

Five DOF camera constraints are published to the server from the camera client process. The camera process is multi-threaded and organized into four main modules: a feature extraction thread, an image saliency thread, a link proposal thread, and a two-view image registration thread. The feature thread extracts robust features used for correspondence detection. The saliency thread then uses these extracted features to create a bag-of-words representation for the image and computes a visual saliency score. The link proposal thread uses the visual saliency metric along with a calculation of geometric information gain to (*i*) add only salient nodes to the graph and (*ii*) to propose visually informative candidates for registration. We call the process of hypothesizing possible loop-closure candidates "link proposal", because a measurement will act as a "link" (i.e., constraint) between two nodes in our pose-graph framework. The extracted features and proposed links are then fed to the two-view thread for attempted registration.

## A.3 Robot / Sensor Coordinate Frames

We define the vehicle coordinate frame following the SNAME[1] convention (Fossen, 1994), $x$ forward, $y$ starboard, and $z$ down, as in Figure A.4, and centered on the DVL sensor. The DVL and Dual frequency IDentification SONar (DIDSON) sonar are separately pitch-actuated with two different servos (Figure A.4(a)). As described in A.1, the DVL is pitch-actuated via servo to keep a orthogonal view to the hull throughout the mission.

We use the Smith, Self, and Cheeseman (Smith et al., 1990) coordinate frame convention and describe pose $\mathbf{x}$ as a 6-tuple vector, where the first three elements indicate position

[1]The Society of Naval Architecture and Marine Engineers.

and the last three indicate Euler angle attitude in roll, pitch, and yaw,

$$\mathbf{x} = [x, y, z, \phi, \theta, \psi]^\top = [\mathbf{p}^\top, \mathbf{r}^\top]^\top, \tag{A.1}$$

where $\mathbf{p}$ indicates the positional components $[x, y, z]^\top$ and $\mathbf{r}$ refers to the orientation components $[\phi, \theta, \psi]^\top$. The pose-graph SLAM reference frame is defined with respect to a local coordinate frame that we initiate as the vehicle first obtains DVL lock on the ship hull. Originating from this point, the sensor measurements and optimized Cartesian 3D pose-graph are all described with respect to this local coordinate frame.

We also need to model the camera coordinate transform with respect to the vehicle frame. The two camera configurations (A.1) have different sensor coordinate transforms—one dynamic and the other static. When the camera is in underwater mode, the camera is located on the same servo as the DVL and, thus, dynamically changing as a function of the servo pitch angle ($p_{servo}$). When the camera is in periscope mode, it is fixed on top of the vehicle at a fixed angle resulting in a static coordinate transform with respect to the vehicle frame. The two sensor transform poses are as below.

$$\begin{aligned} \mathbf{x}_{vc,underwater} &= \mathbf{x}_{vs_1} \oplus \mathbf{x}_{s_1c} \\ &= [0, 0, 0, 0, p_{servo}, 0]^\top \oplus [0, -0.25\text{m}, 0, 90°, 0, 90°]^\top \\ \mathbf{x}_{vc,periscope} &= [-0.453\text{m}, 0.221\text{m}, -0.457\text{m}, (90+60)°, 0, 90°]^\top \end{aligned} \tag{A.2}$$

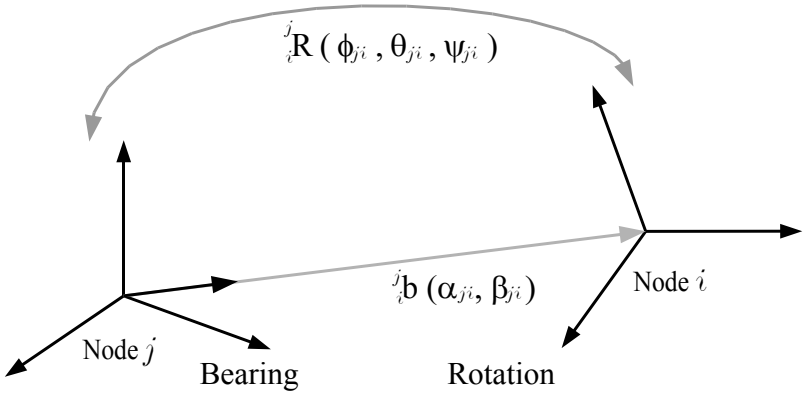
Here $\oplus$ is the head-to-tail operation by Smith et al. (1990). Note that the servo is applicable only to the underwater mode since the camera is oriented with fixed angle in the periscope mode.

## A.4 5-DOF Camera Measurement

The 5-DOF camera measurement used in this thesis, $\mathbf{z}_{ij}$, is between poses $\mathbf{x}_i$ and $\mathbf{x}_j$ and is the 6-DOF relative coordinate transform between cameras modulo scale[2]. The measurement consists of the relative bearing (azimuth $\alpha_{ji}$ and elevation angle $\beta_{ji}$) and the relative Euler orientation ($\phi_{ji}, \theta_{ji}, \psi_{ji}$) between the two camera poses $\mathbf{x}_{c_i} = [x_{c_i}, y_{c_i}, z_{c_i}, \phi_{c_i}, \theta_{c_i}, \psi_{c_i}]^\top$ and $\mathbf{x}_{c_j} = [x_{c_j}, y_{c_j}, z_{c_j}, \phi_{c_j}, \theta_{c_j}, \psi_{c_j}]^\top$ (Eustice et al., 2008, 2006b),

[2]The measurement is usually the relative bearing and orientation of the historic camera pose as seen by the most recent camera pose.

**Figure** A.**5** 5-DOF camera measurement model. The camera measurement is a bearing-only relative-pose between node $i$ and node $j$ that consists of azimuth, elevation and Euler angles.




| Bearing | | Rotation | |
|---|---|---|---|
| $\alpha_{ji}$ | azimuth | $\phi_{ji}$ | Euler roll |
| $\beta_{ji}$ | elevation | $\theta_{ji}$ | Euler pitch |
| | | $\psi_{ji}$ | Euler yaw |

$$\mathbf{z}_{ij} = h_{\mathsf{5dof}}(\mathbf{x}_i, \mathbf{x}_j) = \left[\alpha_{ji}, \beta_{ji}, \phi_{ji}, \theta_{ji}, \psi_{ji}\right]^\top, \tag{A.3}$$

where

$$\alpha_{ji} = \text{atan2}(y_{c_{ji}}, x_{c_{ji}}) \text{ and } \beta_{ji} = \text{atan2}(z_{c_{ji}}, \sqrt{x_{c_{ji}}^2 + y_{c_{ji}}^2}). \tag{A.4}$$

The Jacobian of $\mathbf{z}_{ij}$ with respect to $\mathbf{x}$ is sparse

$$\mathrm{H}_{\mathbf{x}} = \left[0 \quad \cdots \quad \frac{\partial \mathbf{h}_{\mathsf{5dof}}}{\partial \mathbf{x}_i} \quad \cdots \quad 0 \quad \frac{\partial \mathbf{h}_{\mathsf{5dof}}}{\partial \mathbf{x}_j} \quad \cdots \quad 0\right], \tag{A.5}$$

and the nonzero terms are computed using chain rule:

$$\frac{\partial \mathbf{h}_{\mathsf{5dof}}}{\partial \mathbf{x}_i} = \frac{\partial \mathbf{h}_{\mathsf{5dof}}}{\partial \mathbf{x}_{c_{ji}}} \cdot \frac{\partial \mathbf{x}_{c_{ji}}}{\partial \mathbf{x}_i}. \tag{A.6}$$

Here, $\mathbf{x}_{c_{ji}}$ is the 6-DOF relative pose between two camera poses $\mathbf{x}_{c_i}$ and $\mathbf{x}_{c_j}$ obtained by the tail-to-tail operation (Smith et al., 1990). The camera poses are computed from the vehicle pose and the vehicle to sensor transformation $\mathbf{x}_{vc}$, which is either dynamic (underwater mode) or static (periscope mode) as in (A.2). The sensor transformation is static and $\mathbf{x}_{vc_i} = \mathbf{x}_{vc}$ for all poses $i$ when the vehicle is in periscope mode. However, it is time varying otherwise such that $\mathbf{x}_{vc_i} \neq \mathbf{x}_{vc_j}$.

$$\mathbf{x}_{c_{ji}} = \ominus \mathbf{x}_{c_j} \oplus \mathbf{x}_{c_i} = \ominus(\mathbf{x}_j \oplus \mathbf{x}_{vc_j}) \oplus (\mathbf{x}_i \oplus \mathbf{x}_{vc_i}) \tag{A.7}$$

Each term in (A.6) is computed as

$$\frac{\partial \mathbf{h}_{\mathsf{5dof}}}{\partial \mathbf{x}_{c_{ji}}} = \begin{bmatrix} J_{\alpha\beta} & 0_{2\times 3} \\ 0_{3\times 3} & I_{3\times 3} \end{bmatrix}, \text{ where } J_{\alpha\beta} = \frac{\partial(\alpha_{ji}\beta_{ji})}{\partial(x_{ji}, y_{ji}, z_{ji})}, \text{ and} \tag{A.8}$$

$$\frac{\partial \mathbf{x}_{c_{ji}}}{\partial \mathbf{x}_i} = \frac{\partial \ominus (\mathbf{x}_j \oplus \mathbf{x}_{vc_j}) \oplus (\mathbf{x}_i \oplus \mathbf{x}_{vc_i})}{\partial \mathbf{x}_i}. \tag{A.9}$$

The camera measurements are published with the sensor coordinate transformation encoded ($x_{vc}$) for ease of use (e.g., the sensor transform being dynamic). Unlike other sensors, the published observation is with respect to the camera pose and the SLAM back-end assembles them appropriately using the embedded coordinate transformation. The same bearing angle in the camera frame can be transformed differently to the vehicle frame bearing depending on the baseline length. This baseline length will be optimized during SLAM and thus is not a fixed value. Without having a fixed baseline length, transforming the measurement into the vehicle frame will result in erroneous measurement. Note that, however, under the static sensor coordinate transformation, the measurement can be transformed to the vehicle frame due to the properties of the parallelogram. In our implementation, we decided to publish the measurement in the camera frame with the sensor transformation encoded for general application.

# APPENDIX B

# Survey Design using the CRLB

This appendix contains a detailed derivation for an approach that uses the Cramer Rao Lower Bound (CRLB) to measure trajectory uncertainty for survey preplanning, employing pose-graph SLAM.

## B.1 Cramer Rao Lower Bound

In the CRLB derivation, we assume that the AUV is equipped with a camera, a compass, and an odometry sensor (e.g., integrated DVL), and moves in a planar boustrophedon trajectory. We use the notation $\mathbf{p}_i = [x_i, y_i]^\top$ for the position of the vehicle along the trajectory at sample time $i$; and $\theta_i$ for the heading angle of the robot, where $\mathbf{x}_i = [\mathbf{p}_i^\top, \theta_i]^\top$ indicates the state of the vehicle. We use $\mathbf{Z}$ to indicate the stacked vector of all sensor measurements assembled from camera, $\mathbf{Z}_c$, odometry, $\mathbf{Z}_o$, and compass, $\mathbf{Z}_h$, measurements, and $\mathbf{X}$ to indicate the stacked vector of all trajectory samples, $\mathbf{x}_i$.

We use the CRLB (see Bar-Shalom et al. (2001) for standard reference)

$$\mathsf{E}\big[[\hat{\mathbf{X}}(\mathbf{Z}) - \mathbf{X}_0][\hat{\mathbf{X}}(\mathbf{Z}) - \mathbf{X}_0]^\top\big] \geq \mathrm{J}^{-1}, \tag{B.1}$$

as a measure of the conservative uncertainty bound for the pose-graph, where $\mathrm{J}$ in this equation is the Fisher information matrix,

$$\mathrm{J} = \mathsf{E}\big[[\nabla_{\mathbf{X}} \ln \Lambda(\mathbf{X})][\nabla_{\mathbf{X}} \ln \Lambda(\mathbf{X})]^\top\big]\Big|_{\mathbf{X}_0}, \tag{B.2}$$

for the measurement likelihood $\Lambda(\mathbf{X}) = p(\mathbf{Z}|\mathbf{X})$.

When we assume that all sensor measurements are independent, then the likelihood factorizes to a simple product that, after taking its logarithm, becomes a summation of all

sensor log-likelihoods

$$\begin{aligned} \Lambda(\mathbf{X}) &= p\big(\mathbf{Z}\big|\mathbf{X}\big) \\ &= p\big(\mathbf{Z}_c\big|\mathbf{X}\big)p\big(\mathbf{Z}_o\big|\mathbf{X}\big)p\big(\mathbf{Z}_h\big|\mathbf{X}\big) \\ \ln\Lambda(\mathbf{X}) &= \ln p\big(\mathbf{Z}_c\big|\mathbf{X}\big) + \ln p\big(\mathbf{Z}_o\big|\mathbf{X}\big) + \ln p\big(\mathbf{Z}_h\big|\mathbf{X}\big). \end{aligned} \tag{B.3}$$

We model the sensor measurements as corrupted by Gaussian noise with zero mean and covariances $\Sigma_{c_{ij}}$, $\Sigma_{o_i}$, and $\sigma^2_{h_i}$, respectively:

$$\begin{aligned} \text{Camera:} \quad \mathbf{Z}_{c_{ij}} &= \mathbf{g}(\mathbf{x}_i, \mathbf{x}_j) + \mathbf{W}_{c_{ij}}, \mathbf{W}_{c_{ij}} \sim \mathcal{N}(0, \Sigma_{c_{ij}}), \\ \text{Odometry:} \quad \mathbf{Z}_{o_i} &= (\mathbf{p}_i - \mathbf{p}_{i-1}) + \mathbf{W}_{o_i}, \mathbf{W}_{o_i} \sim \mathcal{N}(0, \Sigma_{o_i}), \\ \text{Compass:} \quad Z_{h_i} &= \theta_i + W_{h_i}, W_{h_i} \sim \mathcal{N}(0, \sigma^2_{h_i}), \end{aligned}$$

where we assume a standard deviation of $\pm 1.2$ cm/s in DVL velocity when computing the integrated odometry measurement (Teledyne RD Instruments, 2008) and a fixed $\pm 1^\circ$ for compass heading uncertainty (i.e., $\sigma_{h_i} = \sigma_h = \pm 1^\circ$).

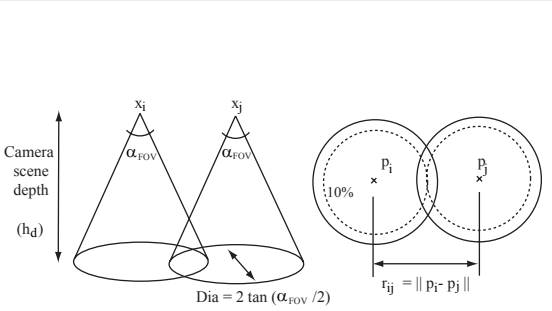
## B.2 Modeling of the Camera Measurement

Camera measurements are modeled using a reduced two-DOF camera measurement model $\mathbf{g}(\mathbf{x}_i, \mathbf{x}_j)$ for the planar survey, which measures the azimuth, $\alpha_{ij}$, and relative orientation, $\beta_{ij}$, between nodes $i$ and $j$. This measurement model treats the camera as a bearing sensor making both sequential and non-sequential links in the graph

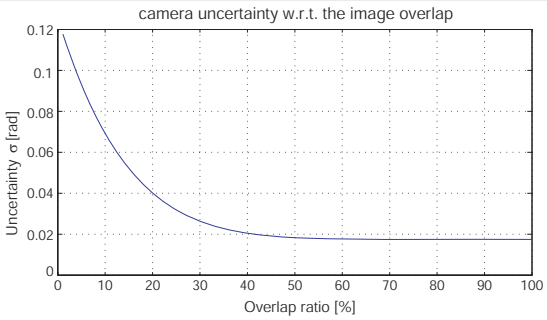
$$\mathbf{g}(\mathbf{x}_i, \mathbf{x}_j) = \begin{bmatrix} \alpha_{ij} \\ \beta_{ij} \end{bmatrix} = \begin{bmatrix} \text{atan2}(y_j - y_i, x_j - x_i) \\ \theta_j - \theta_i \end{bmatrix}.$$

The camera sensor measurement uncertainty, $\Sigma_{c_{ij}}$, is modeled so as to depend on the distance between two nodes as a function of their overlap ratio. The size of the overlap region is commensurate with the proximity of node $i$ and $j$; the closer nodes $i$ and $j$, the larger the overlap region, resulting in less uncertainty in the camera measurement. As can be seen from Figure B.1(a), with a fixed camera FOV and scene depth $h_d$, the overlap ratio becomes a function of the horizontal distance of two nodes only. This fixed scene depth assumption is made considering a survey that maintains constant distance to the target surface. For simplicity, we assume the footprint of the camera measurement to be a circle as illustrated in Figure B.1(a), and define the camera measurement threshold $C_{thresh} = 2h_d \tan(\text{FOV}/2)$ and the overlap ratio as $\gamma = \frac{||\mathbf{p}_i - \mathbf{p}_j||}{C_{thresh}}$. Only when the distance $r_{ij}$ is

**Figure** B.1 Camera uncertainty model. Camera measurement noise is modeled as a function of the image overlap ratio between an idealized pairwise camera measurement.




(a) Pairwise overlap model




(b) Standard deviation of the camera measurement noise as a function of the overlap ratio

smaller than $C_{thresh}$, is the camera measurement meaningful and the information gain is proportional to the distance $r_{ij}$. The function that defines the uncertainty of the camera measurement,

$$\sigma = \pm 1^{\circ}\gamma^{5}, \tag{B.4}$$

is shown in Figure B.1(b). When the overlap ratio falls below $10\%$, the uncertainty increases significantly, and for greater than $50\%$ it remains relatively flat. We assume the camera FOV to be $40^{\circ}$, which gives a $C_{thresh}$ of $0.73$ m at a scene depth of $1$ m, for example.

## B.3 Fisher Information Matrix

As previously shown in (B.2), the Fisher Information matrix, J, is simply the expectation of the squared gradient of the log-likelihood evaluated at the true parameters, $\mathbf{X}_0$. Using the assumption of sensor measurement independence, the expectation of the gradient from two different sensor measurement likelihoods is zero (i.e., $\mathsf{E}\big[[\nabla_{\mathbf{X}} \ln p\big(\mathbf{Z}_m\big|\mathbf{X}\big)]$ $[\nabla_{\mathbf{X}} \ln p\big(\mathbf{Z}_n\big|\mathbf{X}\big)]^{\top}\big] = 0$ when $m \neq n$). Therefore, J reduces to the three simple sensor terms below:

$$\begin{aligned} \mathrm{J} &= \mathsf{E}\big[[\nabla_{\mathbf{X}} \ln p\big(\mathbf{Z}_c\big|\mathbf{X}\big)][\nabla_{\mathbf{X}} \ln p\big(\mathbf{Z}_c\big|\mathbf{X}\big)]^{\top}\big] \\ &+ \mathsf{E}\big[[\nabla_{\mathbf{X}} \ln p\big(\mathbf{Z}_o\big|\mathbf{X}\big)][\nabla_{\mathbf{X}} \ln p\big(\mathbf{Z}_o\big|\mathbf{X}\big)]^{\top}\big] \\ &+ \mathsf{E}\big[[\nabla_{\mathbf{X}} \ln p\big(\mathbf{Z}_h\big|\mathbf{X}\big)][\nabla_{\mathbf{X}} \ln p\big(\mathbf{Z}_h\big|\mathbf{X}\big)]^{\top}\big]. \end{aligned}$$

$$
\begin{aligned}
\Lambda_o + \Lambda_h &= \mathsf{E}\big[[\nabla_{\mathbf{X}} \ln p(\mathbf{Z}_o|\mathbf{X})][\nabla_{\mathbf{X}} \ln p(\mathbf{Z}_o|\mathbf{X})]^\top\big] + \mathsf{E}\big[[\nabla_{\mathbf{X}} \ln p(\mathbf{Z}_h|\mathbf{X})][\nabla_{\mathbf{X}} \ln p(\mathbf{Z}_h|\mathbf{X})]^\top\big] \\
&= \left[\begin{array}{cc|cc|c|cc|cc}
\Sigma_{o_2}^{-1} & 0_{2\times1} & -\Sigma_{o_2}^{-1} & 0_{2\times1} & 0_{3\times3} & \cdots & & 0_{3\times3} & \\
0_{1\times2} & 1/\sigma_{h_1}^2 & 0_{1\times2} & 0 & & & & & \\
\hline
-\Sigma_{o_2}^{-1} & 0_{2\times1} & \Sigma_{o_2}^{-1}+\Sigma_{o_3}^{-1} & 0_{2\times1} & 0_{3\times3} & \cdots & & 0_{3\times3} & \\
0_{1\times2} & 0 & 0_{1\times2} & 1/\sigma_{h_2}^2 & & & & & \\
\hline
0_{3\times3} & & 0_{3\times3} & & \ddots & \vdots & \vdots & \vdots & \vdots \\
\hline
\vdots & \vdots & \vdots & \vdots & \cdots & \Sigma_{o_{n-1}}^{-1}+\Sigma_{o_n}^{-1} & 0_{2\times1} & -\Sigma_{o_n}^{-1} & 0_{2\times1} \\
 & & & & \cdots & 0_{1\times2} & 1/\sigma_{h_{n-1}}^2 & 0_{1\times2} & 0 \\
\hline
0_{3\times3} & & 0_{3\times3} & & \cdots & -\Sigma_{o_n}^{-1} & 0_{1\times2} & \Sigma_{o_n}^{-1} & 0_{1\times2} \\
 & & & & \cdots & 0_{1\times2} & 0 & 0_{1\times2} & 1/\sigma_{h_n}^2
\end{array}\right] \\
\Lambda_c &= \mathsf{E}\big[[\nabla_{\mathbf{X}} \ln p(\mathbf{Z}_c|\mathbf{X})][\nabla_{\mathbf{X}} \ln p(\mathbf{Z}_c|\mathbf{X})]^\top\big] \\
&= \left[\begin{array}{c|c|c|c|c}
\sum_{k=1}^{n} \mathrm{D}_{1k} & -\mathrm{D}_{12} & -\mathrm{D}_{13} & \cdots & -\mathrm{D}_{1n} \\
\hline
-\mathrm{D}_{12} & \sum_{k=1}^{n} \mathrm{D}_{2k} & -\mathrm{D}_{23} & \cdots & -\mathrm{D}_{2n} \\
\hline
\vdots & \vdots & \ddots & \vdots & \vdots \\
\hline
-\mathrm{D}_{1(n-1)} & -\mathrm{D}_{2(n-1)} & \cdots & \sum_{k=1}^{n} \mathrm{D}_{(n-1)k} & -\mathrm{D}_{(n-1)n} \\
\hline
-\mathrm{D}_{1n} & -\mathrm{D}_{2n} & \cdots & -\mathrm{D}_{(n-1)n} & \sum_{k=1}^{n} \mathrm{D}_{nk}
\end{array}\right]
\end{aligned}
\tag{B.6}
$$

By defining information matrices $\Lambda_c$, $\Lambda_o$, and $\Lambda_h$, to correspond to the three terms above, respectively, the Fisher information, J, can be written as the summation of the three information matrices as contributed from each sensor,

$$\mathrm{J} = \Lambda_o + \Lambda_h + \Lambda_c.$$

The detailed general structure for the three information matrices is given in equation (B.6).

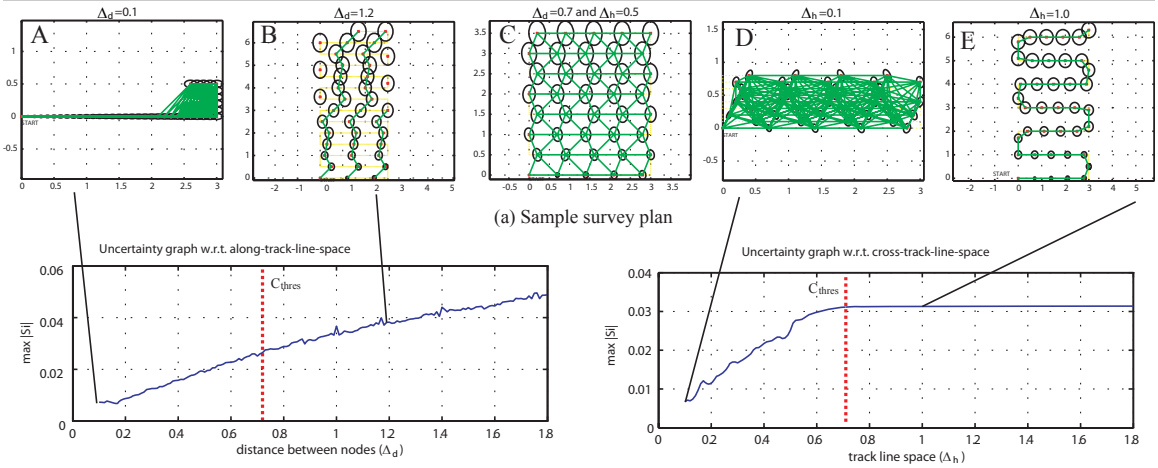
Unlike the odometry and heading sensors, the camera information matrix, $\Lambda_c$, is a function of geometry and can be written in terms of the camera observation model gradient as

$$\mathrm{D}_{ij} = \frac{\partial \mathbf{g}(\mathbf{x}_i, \mathbf{x}_j)}{\partial \mathbf{x}_i} \Sigma_{c_{ij}}^{-1} \frac{\partial \mathbf{g}(\mathbf{x}_i, \mathbf{x}_j)}{\partial \mathbf{x}_j}^\top. \tag{B.5}$$

While the information matrix of the odometry and heading sensors depend only on the sensor uncertainty and integration interval, (B.5) makes it clear that the camera information is a function of the sensor uncertainty and the trajectory geometry. In particular, $\Lambda_c$ will exhibit nonzero elements in the off diagonal of the Fisher information matrix since the camera measurement can result from non-sequential nodes in the graph.

## B.4 Effect of the Design Parameters

**Figure** B.2 Example trajectory design paths. Change of maximum uncertainty with respect to the altitude normalized along-track-line-space, $\Delta_d$, and the cross-track-line-space, $\Delta_h$. Example trajectory design paths are depicted in (a).




(a) Sample survey plan

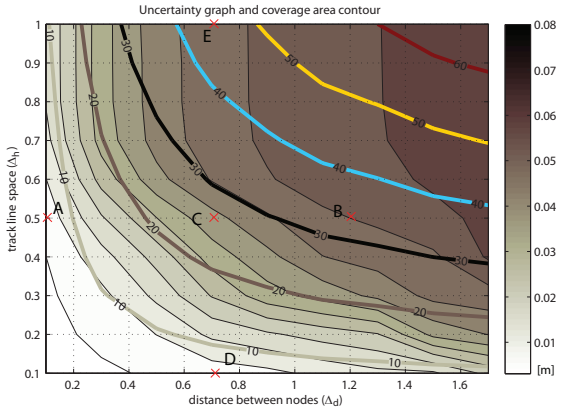
(b) Evaludated uncertainty w.r.t. d and h

We examine the CRLB by evaluating the inverse of the Fisher Information matrix, which gives the covariance matrix, S, of the overall graph. The determinant of each covariance matrix sub-block, $\mathrm{S}_{ii}$, reveals the uncertainty of each node in the graph. We evaluate the maximum $x, y$ standard deviation for all nodes in the graph as $\max_i \sqrt[4]{|\mathrm{S}_{xy_{ii}}|}$ by looking at the $S_{xy}$ sub-block for each node and use this as a metric for the level of $x, y$ uncertainty in the entire trajectory. Figure B.2 and B.3 depict this survey metric for a more extended case considering a 40 node trajectory.

With respect to the odometry sensor uncertainty on the $i^{th}$ node, for example, $\Sigma_{o_i}$ will increase as the along-track-line-space, $\Delta_d$, increases because it accumulates DVL uncertainty over longer distances. However, the reduction of the cross-track-line-space, $\Delta_h$, will enhance the overall pose-graph link structure since the camera can make a larger number of non-sequential links to other track-lines. In terms of the camera measurements, small $\Delta_d$ and $\Delta_h$ values enable a higher density of camera measurements, thereby reducing the uncertainty; however, this also reduces the size of the survey coverage area, as depicted in plans A, and D of Figure B.2(a). To achieve an optimal combination of these parameters, we examine the relation of the uncertainty and the survey coverage with respect to the design parameters to achieve an optimal balance given these competing considerations.

The two plots in Figure B.2(b) show the relation of the camera measurement with the two independent design parameters, $\Delta_d$ and $\Delta_h$. As can be seen in Figure B.2(b), the maximum pose uncertainty does not rapidly increase until it passes the camera measurement threshold, $C_{thresh}$. Once the cross-track-line-space exceeds the camera measurement threshold, the resulting pose-graph (Figure B.2(b) Survey E) no longer makes cross-track

**Figure** B.3 CRLB based contour plot. The contour plot of the maximum uncertainty and the coverage area. The letters from A to E indicate each sample plan in Figure B.2. The curved lines with numbers indicate the altitude normalized coverage area of the plan, while the contours underneath show the altitude normalized maximum standard deviation from the CRLB. The color bar on the left side represents the uncertainty level.




links, which significantly increases the maximum trajectory uncertainty. Therefore, without considering the coverage area, it would be reasonable to select the values of $\Delta_d$ and $\Delta_h$ according to the camera measurement threshold, $C_{thresh}$, which is a function of the camera's intrinsic parameters, field of view, and scene altitude. However, since the uncertainty and coverage area should be evaluated simultaneously, a contour plot containing both factors is plotted as in Figure B.3, which can be used to analyze and optimally preplan the survey.

In the contour plot, each sample path is marked with capital letters from A to E. Path C has been selected to be optimal in that it achieves the minimum uncertainty for coverage of the largest area among the parameters for the same uncertainty level. Under plan C ($\Delta_d = 0.7$ and $\Delta_h = 0.5$), an altitude normalized[1] max standard deviation of 0.017 is estimated for an altitude normalized total coverage area of 10.4.

## B.5 Conclusion

In this appendix, we described a preplanning method for improving the localization performance of a pose-graph SLAM methodology using sequential/non-sequential relative-pose constraints. A conservative static uncertainty bound was examined by calculating the CRLB in the AUV survey design phase, and thereby determining the uncertainty level of the graph. A graphical design tool based upon contour plots of the coverage area and maximum uncertainty was developed to interpret the effects of the design parameters on

[1]The results are normalized with altitude (the scene depth $h_d$).

the survey coverage and uncertainty. Using this tool, design parameters were proposed to meet the purpose of the survey as a preplanning instrument.

# APPENDIX C

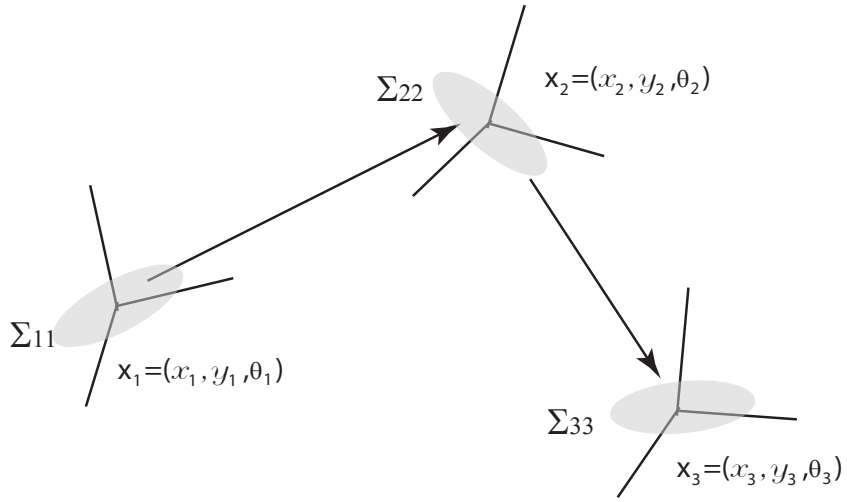
# Monotonicity of Covariance Propagation

This chapter shows that the determinant of the propagated covariance matrix in dead-reckoned motion increases monotonically for all cases, whereas the trace may fail to hold this property, and may actually decrease depending upon the robot's motion. Usually, it is thought that the trace and determinant hold monotonicity even though any one sub element may decrease due to the motion of a robot (Kelly, 2004). Kelly (2004) proved that the reduction of any one element in the propagated covariance matrix results in an increase in the other elements. For example, if we examine a subblock of the covariance matrix corresponding to position, it may decrease when there is a significant orientation change. In this case, however, the reduction in position uncertainty has been transferred to the orientation uncertainty increase, and thus, the total uncertainty is growing monotonically for all cases. Therefore, two metrics, the trace and determinant, are considered to be proper metrics for evaluation of the monotonicity property. However, in this chapter, we show this might not always be the case and justify using the determinant as the evaluation criteria.

The following example illustrates a case when the trace of the covariance propagation fails to hold monotonicity, while the determinant conserves it. Consider the following simple three-step propagation example in 2D as in Figure C.1. Initially starting from $\mathbf{x}_1 = (x_1, y_1, \theta_1)$, a robot moves to $\mathbf{x}_2 = (x_2, y_2, \theta_2)$ and then to $\mathbf{x}_3 = (x_3, y_3, \theta_3)$. The robot starts with initial state uncertainty $\Sigma_{11}$ and control noise $Q$. We will show how the determinant and the trace of the covariance changes through this two-step motion, following a similar method as in Smith et al. (1988). In general, the initial covariance, $\Sigma_{11}$, is very small (i.e., we know our position very well at the beginning). However, note that, in perception-driven navigation (PDN), the propagation starts from the current pose, thus this initial covariance ($\Sigma_{11}$) could be large. The Jacobian for the first step of propagation can be written as $J_{12}$, following Smith et al. (1988),

$$J_{12} = \left[\begin{array}{ccc|ccc} 1 & 0 & -(y_2 - y_1) & \cos\theta_1 & -\sin\theta_1 & 0 \\ 0 & 1 & (x_2 - x_1) & \sin\theta_1 & \cos\theta_1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{array}\right] = [T_1|R_1]. \tag{C.1}$$

**Figure C.1** 2D Covariance propagation example of three poses.




For easy derivation, we define $J_{i,i+1} = [T_i|R_i]$, where $T_i$ represents the translational part of the Jacobian and $R_i$ represents the rotational part of the Jacobian. Note that $T_i$ is a upper triangular matrix and $R_i$ is an orthonormal matrix. Using this Jacobian, the second node uncertainty can be written as

$$\Sigma_{22} = J_{12} \begin{bmatrix} \Sigma_{11} & 0 \\ 0 & Q \end{bmatrix} J_{12}^{\top} = [T_1|R_1] \begin{bmatrix} \Sigma_{11} & 0 \\ 0 & Q \end{bmatrix} \begin{bmatrix} T_1^{\top} \\ R_1^{\top} \end{bmatrix} = T_1 \Sigma_{11} T_1^{\top} + R_1 Q R_1^{\top}. \tag{C.2}$$

In the above equation, the first term indicates the propagated initial uncertainty, whereas the second term indicates the additive control noise. If we take another step of propagation, the covariance for the third node becomes

$$\begin{aligned} \Sigma_{33} &= J_{23} \begin{bmatrix} \Sigma_{22} & 0 \\ 0 & Q \end{bmatrix} J_{23}^{\top} = [T_2|R_2] \begin{bmatrix} \Sigma_{22} & 0 \\ 0 & Q \end{bmatrix} \begin{bmatrix} T_2^{\top} \\ R_2^{\top} \end{bmatrix} = T_2 \Sigma_{22} T_2^{\top} + R_2 Q R_2^{\top} \\ &= T_2 (T_1 \Sigma_{11} T_1^{\top} + R_1 Q R_1^{\top}) T_2^{\top} + R_2 Q R_2^{\top} \\ &= T_2 T_1 \Sigma_{11} T_1^{\top} T_2^{\top} + T_2 R_1 Q R_1^{\top} T_2^{\top} + R_2 Q R_2^{\top}. \end{aligned}$$

For simplicity, we can write the general propagated covariance after $n$ steps as,

$$\Sigma_{nn} = \overbrace{(\prod_{i=n}^{1} T_i)\Sigma_{11}(\prod_{i=n}^{1} T_i)^\top}^{\text{initial uncertainty}} + \overbrace{\sum_{i=1}^{n} \left( (\prod_{j=n}^{n-i} T_j) R_i \right) Q \left( (\prod_{j=n}^{n-i} T_j) R_i \right)^\top}^{\text{additive uncertainty}}. \tag{C.3}$$

Note that, $\prod_{i=n}^{1} T_i := T^{(n)}$ is a upper triangular matrix, which is simply the difference between the final and the initial pose,

$$\prod_{i=n}^{1} T_i := T^{(n)} = \begin{bmatrix} 1 & 0 & -(y_{n+1} - y_1) \\ 0 & 1 & (x_{n+1} - x_1) \\ 0 & 0 & 1 \end{bmatrix}. \tag{C.4}$$

For example, the $T_2 T_1$ term results in the following form,

$$T_2 T_1 = \begin{bmatrix} 1 & 0 & -(y_3 - y_2) \\ 0 & 1 & (x_3 - x_2) \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -(y_2 - y_1) \\ 0 & 1 & (x_2 - x_1) \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -(y_3 - y_1) \\ 0 & 1 & (x_3 - x_1) \\ 0 & 0 & 1 \end{bmatrix}. \tag{C.5}$$

Then, we can rewrite (C.3) as

$$\Sigma_{nn} = T^{(n)} \Sigma_{11} {T^{(n)}}^\top + \sum_{i=1}^{n} \left( T^{(n-i)} R_i \right) Q \left( T^{(n-i)} R_i \right)^\top.$$

Consider the example again in the case where $\mathbf{x}_1 = \mathbf{x}_3$; i.e., we moved to the second node and came back to the initial pose on the third step. When there is a significant order difference between the eigenvalues of $\Sigma_{11}$ and $Q$ as in PDN, this difference becomes the major source of broken monotonicity in the trace of the covariance matrix. Because $tr(A + B) = tr(A) + tr(B)$, we can rewrite the trace of $\Sigma_{22}$ and $\Sigma_{33}$ as

$$\begin{aligned} tr(\Sigma_{11}) &= tr(\Sigma_{11}), \\ tr(\Sigma_{22}) &= tr(T^{(2)} \Sigma_{11} {T^{(2)}}^\top) + tr(R_1 Q R_1^\top), \\ tr(\Sigma_{33}) &= tr(T^{(3)} \Sigma_{11} {T^{(3)}}^\top) + tr(T_2 R_1 Q R_1^\top T_2^\top) + tr(R_1 Q R_1^\top). \end{aligned}$$

$R$ is a orthonormal matrix and thus does not change the eigenvalues, i.e., $tr(R_1 Q R_1) = tr(Q)$. Furthermore, due to the order difference between $\Sigma_{11}$ and $Q$, the dominant term is

the first term in each equation.

$$
\begin{aligned}
T^{(n)}\Sigma_{11}T^{(n)^\top} &= \begin{bmatrix} 1 & 0 & \alpha \\ 0 & 1 & \beta \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \alpha & \beta & 1 \end{bmatrix} \\
&= \begin{bmatrix} \sigma_{11} + 2\alpha\sigma_{13} + \alpha^2\sigma_{33} & * & * \\ * & \sigma_{22} + 2\beta\sigma_{23} + \beta^2\sigma_{33} & * \\ * & * & \sigma_{33} \end{bmatrix}.
\end{aligned}
$$

Therefore, $tr(\Sigma_{22})$ results in some extra terms additional to the initial covariance trace,

$$
\begin{aligned}
tr(\Sigma_{22}) &= tr(T^{(2)}\Sigma_{11}T^{(2)^\top}) + tr(R_1QR_1), \\
&= \sigma_{11} + \sigma_{22} + \sigma_{33} + 2\alpha\sigma_{13} + \alpha^2\sigma_{33} + 2\beta\sigma_{23} + \beta^2\sigma_{33} + tr(Q), \\
&= tr(\Sigma_{11}) + tr(Q) + \underbrace{\beta^2\sigma_{33} + \alpha^2\sigma_{33}}_{>0} + \underline{2\alpha\sigma_{13} + 2\beta\sigma_{23}}.
\end{aligned}
$$

For the third covariance, though, if we take a step back to the initial pose ($\mathbf{x}_1$), $T^{(3)}$ becomes identity as $\mathbf{x}_1$ and $\mathbf{x}_3$ are the same position,

$$
\begin{aligned}
tr(\Sigma_{11}) &= tr(\Sigma_{11}), \\
tr(\Sigma_{22}) &= tr(T^{(2)}\Sigma_{11}T^{(2)^\top}) + tr(R_1QR_1^\top), \\
&\sim tr(\Sigma_{11}) + \beta^2\sigma_{33} + \alpha^2\sigma_{33} + 2\alpha\sigma_{13} + 2\beta\sigma_{23}, \\
tr(\Sigma_{33}) &= tr(T^{(3)}\Sigma_{11}T^{(3)^\top}) + tr(T_2R_1QR_1^\top T_2^\top) + tr(R_1QR_1^\top), \\
&\sim tr(\Sigma_{11}) \quad \text{when } tr(\Sigma_{11}) >> tr(Q).
\end{aligned}
$$

Therefore, when the eigenvalues of $Q$ are relatively smaller as compared to the initial covariance, $tr(\Sigma_{33}) \sim tr(\Sigma_{11})$, does not hold the monotonicity of the covariance trace. However, the determinant is always monotone. Using the inequality $det(A + B) \geq det(A) + det(B)$ for non-negative Hermitian matrices $A$ and $B$ (Marcus and Minc, 1964),

$$
\begin{aligned}
det(\Sigma_{nn}) &= det(T\Sigma_{n-1,n-1}T^\top + RQR^\top), \\
&\geq det(T\Sigma_{n-1,n-1}T^\top) + det(RQR^\top), \\
&= det(T)det(\Sigma_{n-1,n-1})det(T) + det(Q), \\
&= det(\Sigma_{n-1,n-1}) + det(Q), \\
det(\Sigma_{nn}) &\geq det(\Sigma_{n-1,n-1}) + det(Q) > det(\Sigma_{n-1,n-1}),
\end{aligned}
$$

which proves its monotonicity. Because monotonicity holds no matter what form the initial covariance takes, we chose the determinant in defining the reward function for PDN.

# BIBLIOGRAPHY

# BIBLIOGRAPHY


Acar, E. U., H. Choset, Y. Zhang, and M. Schervish (2003), Path planning for robotic demining: Robust sensor-based coverage of unstructured environments and probabilistic methods, *International Journal of Robotics Research*, *22*(7-8), 441–466.

Akaike, H. (1974), A new look at the statistical model identification, *IEEE Transaction on Automatic Control*, *19*(6), 716–723.

Alcocer, A., P. Oliveira, and A. Pascoal (2006), Underwater acoustic positioning systems based on buoys with GPS, in *Proceedings of the IEEE European Conference on Underwater Acoustics*, pp. 1–8, Portugal.

Angeli, A., D. Filliat, S. Doncieux, and J.-A. Meyer (2008), Fast and incremental method for loop-closure detection using bags of visual words, *IEEE Transaction on Robotics*, *24*(5), 1027–1037.

Austin, T., D. Hosom, and D. Kuchta (1984), Long baseline acoustic navigation—a flexible approach to custom applications, in *OCEANS 1984*, vol. 16, pp. 69–74.

Baek, S., T.-K. Lee, O. H. Se-Young, and K. Ju (2011), Integrated on-line localization, mapping and coverage algorithm of unknown environments for robotic vacuum cleaners based on minimal sensing, *Advanced Robotics*, *25*(13–14), 1651–1673.

Bajcsy, R. (1988), Active perception, *Proceedings of the IEEE*, *76*(8), 996–1005.

Bar-Shalom, Y., X. Rong Li, and T. Kirubarajan (2001), *Estimation with applications to tracking and navigation*, John Wiley & Sons, Inc., New York.

Barnard, K., P. Duygulu, D. Forsyth, N. de Freitas, D. M. Blei, and M. I. Jordan (2003), Matching words and pictures, *Journal of Machine Learning Research*, *3*, 1107–1135.

Batalin, M., and G. S. Sukhatme (2007), The design and analysis of an efficient local algorithm for coverage and exploration based on sensor network deployment, *IEEE Transaction on Robotics*, *23*(4), 661–675.

Bay, H., T. Tuytelaars, and L. Van Gool (2006), SURF: Speeded-up robust features, in *Proceedings of the European Conference on Computer Vision*, pp. 404–417, Graz, Austria.

Bay, H., A. Ess, T. Tuytelaars, and L. Van Gool (2008), Speeded-up robust features (SURF), *Computer Vision and Image Understanding*, *110*(3), 346–359.

Belcher, E., W. Hanot, and J. Burch (2002), Dual-frequency identification sonar (DIDSON), in *Proceedings of the International Symposium on Underwater Technology*, pp. 187–192, Tokyo, Japan.

Bohlander, G. S., G. Hageman, F. S. Halliwell, R. H. Juers, and D. C. Lynn (1992), Automated underwater hull maintenance vehicle, *Tech. Rep. ADA261504*, Naval Surface Warfare Center Carderock Division, Bethesda, MD.

Bosse, M., and R. Zlot (2008), Map matching and data association for large-scale two-dimensional laser scan-based SLAM, *International Journal of Robotics Research*, *27*(6), 667–691.

Bosse, M., P. Newman, J. Leonard, and S. Teller (2004), An Atlas framework for scalable mapping, *International Journal of Robotics Research*, *23*, 1113–1139.

Bourgault, F., A. A. Makarenko, S. B. Williams, B. Grocholsky, and H. F. Durrant-Whyte (2002), Information based adaptive robotic exploration, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 540–545.

Bowen, A. D., et al. (2009), Field trials of the Nereus hybrid underwater robotic vehicle in the challenger deep of the Mariana Trench, in *Proceedings of the IEEE/MTS OCEANS Conference and Exhibition*, pp. 1–10, Biloxi, USA.

Bradbeer, R., S. Harrold, F. Nickols, and L. Yeung (1997), An underwater robot for pipe inspection, in *Proceedings of the International Conference on Mechatronics and Machine Vision in Practice*, pp. 152–156.

Bradley, A., M. Feezor, H. Singh, and F. Sorrell (2001), Power systems for autonomous underwater vehicles, *IEEE Journal of Oceanic Engineering*, *26*(4), 526–538.

Brokloff, N. (1994), Matrix algorithm for Doppler sonar navigation, in *Proceedings of the IEEE/MTS OCEANS Conference and Exhibition*, vol. 3, pp. 378–383, Brest, France.

Brown, M., and D. G. Lowe (2005), Unsupervised 3D object recognition and reconstruction in unordered datasets, in *Proceedings of the International Conference on 3-D Digital Imaging and Modeling*, pp. 56–63, IEEE Computer Society, Washington, DC, USA.

Brumley, B., P. Heuchling, R. Koehler, and E. Terray (1987), Coded pulse-coherent Doppler sonar, in *Proceedings of the IEEE/MTS OCEANS Conference and Exhibition*, pp. 89–92.

Bryson, M., and S. Sukkarieh (2005), An information-theoretic approach to autonomous navigation and guidance of an uninhabited aerial vehicle in unknown environments, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3770–3775.

Campbell, J., R. Sukthankar, I. Nourbakhsh, and A. Pahwa (2005), A robust visual odometry and precipice detection system using consumer-grade monocular vision, in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3421–3427.

Carlevaris-Bianco, N., and R. M. Eustice (2011), Multi-view registration for feature-poor underwater imagery, in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 423–430, Shanghai, China.

Carlevaris-Bianco, N., A. Mohan, and R. M. Eustice (2010), Initial results in underwater single image dehazing, in *Proceedings of the IEEE/MTS OCEANS Conference and Exhibition*, pp. 1–8, Seattle, WA.

Carroll, K., S. McClaran, E. Nelson, D. Barnett, D. Friesen, and G. William (1992), AUV path planning: An A* approach to path planning with consideration of variable vehicle speeds and multiple, overlapping, time-dependent exclusion zones, *Proceedings of the Symposium on Autonomous Underwater Vehicle Technology*, pp. 79–84.

Carvalho, A., L. Sagrilo, I. Silva, J. Rebello, and R. Carneval (2003), On the reliability of an automated ultrasonic system for hull inspection in ship-based oil production units, *Applied Ocean Research*, *25*, 235–241.

Chaumette, F., and S. Hutchinson (2006), Visual servo control. I. Basic approaches, *IEEE Robotics and Automation Magazine*, *13*(4), 82–90.

Chen, C., and H. Wang (2006), Appearance-based topological bayesian inference for loop-closing detection in a cross-country environment, *International Journal of Robotics Research*, *25*(10), 953–983.

Cheng, Y. (1995), Mean shift, mode seeking, and clustering, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *17*(8), 790–799.

Choset, H. (2001), Coverage for robotics: A survey of recent results, *Annals of Mathematics and Artificial Intelligence*, *31*, 113–126, 10.1023/A:1016639210559.

Choset, H. (2005), *Principles of Robot Motion: Theory, Algorithms, and Implementation*, Intelligent Robotics and Autonomous Agents, MIT Press.

Chow, C. K., and C. N. Liu (1968), Approximating discrete probability distributions with dependence trees, *IEEE Transactions on Information Theory*, *14*, 462–467.

Chum, O., J. Philbin, and A. Zisserman (2008), Near duplicate image detection: min-hash and tf-idf weighting, in *Proceedings of the British Machine Vision Conference*, pp. 493–502.

Connolly, C. (1985), The determination of next best views, in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2, pp. 432–435.

Corke, P., C. Detweiler, M. Dunbabin, M. Hamilton, D. Rus, and I. Vasilescu (2007), Experiments with underwater robot localization and tracking, in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 4556–4561.

Csurka, G., C. R. Dance, L. Fan, J. Willamowski, and C. Bray (2004), Visual categorization with bags of keypoints, in *Proceedings of the European Conference on Computer Vision*, pp. 1–22.

Cummins, M., and P. Newman (2008), FAB-MAP: Probabilistic localization and mapping in the space of appearance, *International Journal of Robotics Research*, *27*(6), 647–665.

Cummins, M., and P. Newman (2009), Highly scalable appearance-only SLAM - FAB-MAP 2.0, in *Proceedings of the Robotics: Science & Systems Conference*, Seattle, USA.

Curti, H., G. Acosta, and O. Calvo (2005), Autonomous underwater pipeline inspection in AUTOTRACKER PROJECT: The simulation module, in *Proceedings of the IEEE OCEANS-Europe Conference and Exhibition*, vol. 1, pp. 384–388.

Daszykowski, M., B. Walczak, and D. L. Massart (2001), Looking for natural patterns in data part 1: Density-based approach, *Chemometrics and Intelligent Laboratory Systems*, *56*(2), 83–92.

Davison, A. J., I. D. Reid, N. D. Molton, and O. Stasse (2007), MonoSLAM: Real-time single camera SLAM, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, *29*, 1052–1067.

D'Errico, J. (2010), Surface fitting using gridfit, `http://www.mathworks.com/matlabcentral/fileexchange/8998`.

Desert Star Systems (2002), Ship hull inspections with aquamap, `http://www.desertstar.com`.

Dudek, G., and D. Jugessur (2000), Robust place recognition using local appearance based methods, in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1030–1035.

Duntley, S. (1963), Light in the sea, *Journal of the Optical Society of America*, *53*(2), 214–233.

Escolano, F., B. Bonev, P. Suau, W. Aguilar, Y. Frauel, J. Saez, and M. Cazorla (2007), Contextual visual localization: Cascaded submap classification, optimized saliency detection, and fast view matching, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1715–1722.

Ester, M., H. Kriegel, J. Sander, and X. Xu (1996), A density-based algorithm for discovering clusters in large spatial databases with noise, in *International Conference on Knowledge Discovery and Data Mining*, pp. 226–231.

Eustice, R. M., H. Singh, and J. J. Leonard (2006a), Exactly sparse delayed-state filters for view-based SLAM, *IEEE Transaction on Robotics*, *22*(6), 1100–1114.

Eustice, R. M., H. Singh, J. J. Leonard, and M. R. Walter (2006b), Visually mapping the RMS Titanic: Conservative covariance estimates for SLAM information filters, *International Journal of Robotics Research*, *25*(12), 1223–1242.

Eustice, R. M., O. Pizarro, and H. Singh (2008), Visually augmented navigation for autonomous underwater vehicles, *IEEE Journal of Oceanic Engineering*, *33*(2), 103–122.

Fairfield, N., G. A. Kantor, D. Jonak, and D. Wettergreen (2008), DEPTHX autonomy software: Design and field results, *Tech. Rep. CMU-RI-TR-08-09*, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.

Fattal, R. (2008), Single image dehazing, in *Proceedings of the International Conference and Exhibition on Computer Graphics and Interactive Techniques*, pp. 1–9, ACM, New York, NY, USA.

Feder, H. J. S., J. J. Leonard, and C. M. Smith (1999), Adaptive mobile robot navigation and mapping, *International Journal of Robotics Research*, *18*(7), 650–668.

Fei-Fei, L., and P. Perona (2005), A bayesian hierarchical model for learning natural scene categories, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 524–531.

Fossen, T. (1994), *Guidance and Control of Ocean Vehicles*, Wiley.

Frintrop, S., and P. Jensfelt (2008), Active gaze control for attentional visual SLAM, in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3690–3697.

Fukunaga, K., and L. Hostetler (1975), The estimation of the gradient of a density function, with applications in pattern recognition, *IEEE Transactions on Information Theory*, *21*(1), 32–40.

Furgale, P., and C. H. Tong (2010), Speeded Up SURF: A GPU implementation of speeded up robust features (SURF), `http://asrl.utias.utoronto.ca/code/gpusurf/`.

Gheissari, N., and A. Bab-Hadiashar (2005), Detecting cylinders in 3D range data using model selection criteria, in *Proceedings of the International Conference on 3-D Digital Imaging and Modeling*, pp. 158–163, IEEE Computer Society, Los Alamitos, CA, USA.

Gonzalez-Banos, H. H., and J.-C. Latombe (2002), Navigation strategies for exploring indoor environments, *International Journal of Robotics Research*, *21*(10–11), 829–848.

Gracias, N., S. van der Zwaan, A. Bernardino, R. Bernardino, and J. Santos-Victor (2003), Mosaic based navigation for autonomous underwater vehicles, *IEEE Journal of Oceanic Engineering*, *28*(4), 609–624.

Grisetti, G., D. Lodi Rizzini, C. Stachniss, E. Olson, and W. Burgard (2008), Online constraint network optimization for efficient maximum likelihood map learning, in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1880–1885, Pasadena, CA.

Grisetti, G., R. K, C. Stachniss, and C. Hertzberg (2010), Hierarchical optimization on manifolds for online 2d and 3d mapping, *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 273–278.

Gustafson, E., B. Jalving, ystein Engelhardtsen, and N. Burchill (2011), HUGIN 1000 Arctic class AUV, in *The Polar Petroleum Potential Conference & Exhibition*, pp. 714–721.

Haralick, R. (1994), Propagating covariance in computer vision, in *Proceedings of the International Conference Pattern Recognition*, vol. 1, pp. 493–498, Jerusalem, Israel.

Harris, C., and M. Stephens (1988), A combined corner and edge detector, in *Proceedings of the Alvey Vision Conference*, pp. 147–151, Manchester, U.K.

Harris, S., and E. Slate (1999), Lamp Ray: Ship hull assessment for value, safety and readiness, in *Proceedings of the IEEE/MTS OCEANS Conference and Exhibition*, vol. 1, pp. 493–500, Seattle, WA.

Hartley, R., and A. Zisserman (2000), *Multiple View Geometry in Computer Vision*, Cambridge University Press.

Hert, S., S. Tiwari, and V. Lumelsky (1996), A terrain-covering algorithm for an AUV, *Autonomous Robots*, *3*, 91–119.

Ho, K. L., and P. Newman (2007), Detecting loop closure with scene sequences, *International Journal of Computer Vision*, *74*(3), 261–286.

Hollinger, G., U. Mitra, and G. Sukhatme (2012), Active and adaptive dive planning for dense bathymetric mapping, in *Proceedings of the International Conference on Experimental Robotics*, Accepted, To Appear.

Horn, B. (1991), Relative orientation revisited, *Journal of the Optical Society of America A*, *8*(10), 1630–1638.

Hover, F. S., R. M. Eustice, A. Kim, B. Englot, H. Johannsson, M. Kaess, and J. J. Leonard (2012), Advanced perception, navigation and planning for autonomous in-water ship hull inspection, *International Journal of Robotics Research*, In Press.

Hu, M.-J., C.-H. Li, Y.-Y. Qu, and J.-X. Huang (2009), Foreground objects recognition in video based on bag-of-words model, in *Proceedings of the Chinese Conference on Pattern Recognition*, pp. 1–5.

Huang, A. S., E. Olson, and D. C. Moore (2010), LCM: Lightweight communications and marshalling, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4057–4062, Taipei, Taiwan.

Huang, S., N. M. Kwok, G. Dissanayake, Q. P. Ha, and G. Fang (2005), Multi-step look-ahead trajectory planning in SLAM: Possibility and necessity, in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1091–1096.

Ila, V., J. Porta, and J. Andrade-Cetto (2010), Information-based compact pose SLAM, *IEEE Transaction on Robotics*, *26*(1), 78–93.

Itti, L., and C. Koch (2001), Computational modeling of visual attention, *Nature Reviews Neuroscience*, *2*(3), 194–203.

Jaffe, J., K. Moore, J. McLean, and M. Strand (2001), Underwater optical imaging: Status and prospects, *Oceanography*, *14*(3), 66–76.

Jegou, H., M. Douze, and C. Schmid (2009), On the burstiness of visual elements, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1169–1176.

Johannsson, H., M. Kaess, B. Englot, F. Hover, and J. J. Leonard (2010), Imaging sonar-aided navigation for autonomous underwater harbor surveillance, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4396–4403, Taipei, Taiwan.

Johnson-Roberson, M. (2010), Large-scale multi-sensor 3D reconstructions and visualizations of unstructured underwater environments, Ph.D. thesis, The University of Sydney.

Johnson-Roberson, M., O. Pizarro, S. B. Williams, and I. Mahon (2010), Generation and visualization of large-scale three-dimensional reconstructions from underwater robotic surveys, *Journal of Field Robotics*, *27*(1), 21–51.

Jones, K. S. (1972), A statistical interpretation of term specificity and its application in retrieval, *Journal of Documentation*, *28*, 11–21.

Julesz, B. (1981), Textons, the elements of texture perception, and their interactions, *Nature*, *290*(5802), 91–97.

Julian, B. J., M. Angermann, M. Schwager, and D. Rus (2012), Distributed robotic sensor networks: An information theoretic approach, *International Journal of Robotics Research*, In Press.

Kadir, T., and M. Brady (2001), Saliency, scale and image description, *International Journal of Computer Vision*, *45*(2), 83–105.

Kaelbling, L. P., M. L. Littman, and A. R. Cassandra (1995), Planning and acting in partially observable stochastic domains, *Artificial Intelligence*, *101*, 99–134.

Kaess, M. (2008), Incremental smoothing and mapping, Ph.D. thesis, Georgia Institute of Technology.

Kaess, M., and F. Dellaert (2009), Covariance recovery from a square root information matrix for data association, *Robotics and Autonomous Systems*, *57*, 1198–1210.

Kaess, M., A. Ranganathan, and F. Dellaert (2008), iSAM: Incremental smoothing and mapping, *IEEE Transaction on Robotics*, *24*(6), 1365–1378.

Kaess, M., K. Ni, and F. Dellaert (2009), Flow separation for fast and robust stereo odometry, in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3539–3544.

Kaess, M., H. Johannsson, and J. Leonard (2010), Open source implementation of iSAM, `http://people.csail.mit.edu/kaess/isam`.

Kaess, M., H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert (2012), iSAM2: Incremental smoothing and mapping using the Bayes tree, *International Journal of Robotics Research*, *31*, 217–236.

Kanatani, K., and Y. Kanazawa (1999), Stabilizing image mosaicing by the geometric AIC, in *Proceedings of the Workshop on Information-Based Induction Sciences*, pp. 181–186.

Kavraki, L., P. Svestka, J. Latombe, and M. Overmars (1996), Probabilistic roadmaps for path planning in high-dimensional configuration spaces, *IEEE Transaction on Robotics and Automation*, *12*, 566–580.

Kavraki, L. E., and S. M. LaValle (2008), Motion planning, in *Springer Handbook of Robotics*, pp. 109–131, Springer.

Kawewong, A., N. Tongprasit, S. Tangruamsub, and O. Hasegawa (2010), Online and Incremental Appearance-based SLAM in Highly Dynamic Environments, *International Journal of Robotics Research*.

Kelly, A. (2004), Linearized error propagation in odometry, *International Journal of Robotics Research*, *23*(2), 179–218.

Khotanzad, A., and Y. H. Hong (1990), Invariant image recognition by Zernike moments, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, *12*(5), 489–497.

Kim, A., and R. M. Eustice (2009), Pose-graph visual SLAM with geometric model selection for autonomous underwater ship hull inspection, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1559–1565, St. Louis, MO.

Klein, G., and D. Murray (2007), Parallel tracking and mapping for small AR workspaces, in *Proceedings of the IEEE/ACM International Symposium on Mixed and Augmented Reality*, pp. 1–10, Nara, Japan.

Knopp, J., J. Sivic, and T. Pajdla (2010), Avoding confusing features in place recognition, in *Proceedings of the European Conference on Computer Vision*, pp. 748–761.

Kollar, T., and N. Roy (2008), Trajectory optimization using reinforcement learning for map exploration, *International Journal of Robotics Research*, *27*(2), 175–196.

Konolige, K. (2004), Large-scale map-making, in *Proceedings of the AAAI National Conference on Artifical Intelligence*, pp. 457–463, San Jose, CA.

Konolige, K. (2005), SLAM via variable reduction from constraint maps, in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 667–672.

Konolige, K., and M. Agrawal (2008), FrameSLAM: From bundle adjustment to real-time visual mapping, *IEEE Transaction on Robotics*, *24*(5), 1066–1077.

Konolige, K., M. Agrawal, and J. Solà (2007), Large scale visual odometry for rough terrain, in *Proceedings of the International Symposium on Robotics Research*, pp. 201–212.

Konolige, K., J. Bowman, J. D. Chen, P. Mihelich, M. Calonder, V. Lepetit, and P. Fua (2009), View-based maps, in *Proceedings of the Robotics: Science & Systems Conference*, Seattle, USA.

Kragic, D., and H. I. Christensen (2002), Survey on visual servoing for manipulation, *Tech. Rep. TRITA-NA-P02/01*, Computer vision and active perception laboratory, KTH Royal Institute of Technology, Stockholm, Sweden.

Kretzschmar, H., G. Grisetti, and C. Stachniss (2010), Lifelong map learning for graph-based SLAM in static environments, *KI – Künstliche Intelligenz*, *24*, 199–206.

Kruger, D., R. Stolkin, A. Blum, and J. Briganti (2007), Optimal AUV path planning for extended missions in complex, fast-flowing estuarine environments, in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 4265–4270.

Kuipers, B., and P. Beeson (2002), Bootstrap learning for place recognition, in *Proceedings of the AAAI National Conference on Artifical Intelligence*, pp. 174–180, Menlo Park, CA, USA.

Kunz, C., et al. (2009), Toward extraplanetary under-ice exploration: Robotic steps in the Arctic, *Journal of Field Robotics*, *26*(4), 411–429.

Kurniawati, H., D. Hsu, and W. Lee (2008), SARSOP: Efficient point-based POMDP planning by approximating optimally reachable belief spaces, in *Proceedings of the Robotics: Science & Systems Conference*.

Larlus, D., J. Verbeek, and F. Jurie (2010), Category level object segmentation by combining bag-of-words models with dirichlet processes and random fields, *International Journal of Computer Vision*, *88*(2), 238–253.

LaValle, S., and J. Kuffner, J.J. (1999), Randomized kinodynamic planning, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, pp. 473–479.

Lazic, N., and P. Aarabi (2007), Importance of feature locations in bag-of-words image classification, in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, vol. 1, pp. 641–644.

Lee, Y.-J., and J.-B. Song (2010), Autonomous salient feature detection through salient cues in an HSV color space for visual indoor simultaneous localization and mapping, *Autonomous Robots*, *24*(11), 1595–1613.

Leonard, J., and H. Feder (2001), Decoupled stochastic mapping, *IEEE Journal of Oceanic Engineering*, *26*(4), 561–571.

Leung, T., and J. Malik (2001), Representing and recognizing the visual appearance of materials using three-dimensional textons, *International Journal of Computer Vision*, *43*(1), 29–44.

Levine, D. S. (2010), Information-rich path planning under general constraints using rapidly-exploring random trees, Master's thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, Cambridge MA.

Li, F., and J. Kosecka (2006), Probabilistic location recognition using reduced feature set, in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3405–3410.

Li, W., and C. Cassandras (2005), Distributed cooperative coverage control of sensor networks, in *Proceedings of the IEEE Conference on Decision and Control*, pp. 2542–2547.

Li, Y. F., and Z. G. Liu (2005), Information entropy-based viewpoint planning for 3-d object reconstruction, *IEEE Transaction on Robotics*, *21*(3), 324–337.

Lowe, D. (2004), Distinctive image features from scale-invariant keypoints, *International Journal of Computer Vision*, *60*(2), 91–110.

Lu, F., and E. Milios (1997), Globally consistent range scan alignment for environment mapping, *Autonomous Robots*, *4*, 333–349.

Luo, J., A. Pronobis, B. Caputo, and P. Jensfelt (2007), Incremental learning for place recognition in dynamic environments, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 721–728.

Lynn, D., and G. Bohlander (1999), Performing ship hull inspections using a remotely operated vehicle, in *Proceedings of the IEEE/MTS OCEANS Conference and Exhibition*, vol. 2, pp. 555–562.

MacKay, D. J. C. (2003), *Information Theory, Inference, and Learning Algorithms*, Cambridge University Press.

Maimone, M. W., Y. Cheng, and L. Matthies (2007), Two years of visual odometry on the Mars exploration rovers, *Journal of Field Robotics*, *24*(3), 169–186.

Makarenko, A. A., S. B. Williams, F. Bourgault, and H. F. Durrant-Whyte (2002), An experiment in integrated exploration, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 534–539.

Maki, T., T. Ura, and T. Sakamaki (2012), AUV navigation around jacket structures II: Map based path-planning and guidance, *Marine Technology Society Journal*, pp. 1–9.

Manning, C. D., P. Raghavan, and H. Schtze (2008), *Introduction to Information Retrieval*, Cambridge University Press, New York, NY, USA.

Marchesotti, L., C. Cifarelli, and G. Csurka (2009), A framework for visual saliency detection with applications to image thumbnailing, in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2232–2239.

Marcus, M., and H. Minc (1964), *A Survey of Matrix Theory and Matrix Inequalities*, Courier Dover Publications.

Matas, J., O. Chum, M. Urban, and T. Pajdla (2004), Robust wide-baseline stereo from maximally stable extremal regions, *Image and Vision Computing*, *22*(10), 761–767.

Medagoda, L., S. B. Williams, O. Pizarro, and M. V. Jakuba (2011), Water column current profile aided localisation combined with view-based SLAM for autonomous underwater vehicle navigation, in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3048–3055.

Menegaldo, L., M. Santos, G. Ferreira, R. Siqueira, and L. Moscato (2008), SIRUS: A mobile robot for floating production storage and offloading (FPSO) ship hull inspection, in *IEEE Int. Workshop Advanced Motion Control*, pp. 27–32.

Menegaldo, L., G. Ferreira, M. Santos, and R. Guerato (2009), Development and navigation of a mobile robot for floating production storage and offloading ship hull inspection, *IEEE Transactions on Industrial Electronics*, *56*(9), 3717–3722.

Mikolajczyk, K., and C. Schmid (2004), Scale and affine invariant interest point detectors, *International Journal of Computer Vision*, *60*(1), 63–86.

Mikolajczyk, K., T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. V. Gool (2005), A comparison of affine region detectors, *International Journal of Computer Vision*, *65*, 43–72.

Mittleman, J., and L. Swan (1993), Underwater inspection for welding and overhaul, *Naval Engineers Journal*, *105*(5), 37–42.

Mittleman, J., and D. Wyman (1980), Underwater ship hull inspection, *Naval Engineers Journal*, *92*(2), 122–128.

Molton, N. D., A. J. Davison, and I. D. Reid (2004), Locally planar patch features for real-time structure from motion, in *Proceedings of the British Machine Vision Conference*, pp. 1–10, BMVC.

Moulin, C., C. Barat, and C. Ducottet (2010), Fusion of tf-idf weighted bag of visual features for image classification, in *Proceedings of the International Workshop on Content-Based Multimedia Indexing*, pp. 1–6.

Mouragnon, E., M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd (2009), Generic and real-time structure from motion using local bundle adjustment, *Image and Vision Computing*, *27*(8), 1178–1193.

Mutambara, A. G. (1999), Information based estimation for both linear and nonlinear systems, in *American Control Conference*, pp. 1329–1333, San Diego, CA.

Negahdaripour, S., and P. Firoozfam (2006), An ROV stereovision system for ship hull inspection, *IEEE Journal of Oceanic Engineering*, *31*(3), 551–546.

Neira, J., and J. Tardos (2001), Data association in stochastic mapping using the joint compatibility test, *IEEE Transaction on Robotics and Automation*, *17*(6), 890–897.

Nicosevici, T., and R. Garcia (2009), On-line visual vocabularies for robot navigation and mapping, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 205–212.

Nister, D., and H. Stewenius (2006), Scalable recognition with a vocabulary tree, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2161–2168.

Nister, D., O. Naroditsky, and J. Bergen (2004), Visual odometry, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 652–659.

Nister, D., O. Naroditsky, and J. Bergen (2006), Visual odometry for ground vehicle applications, *Journal of Field Robotics*, *23*, 3–20.

Nowak, E., F. Jurie, and B. Triggs (2006), Sampling strategies for bag-of-features image classification, in *Proceedings of the European Conference on Computer Vision*, vol. 3954, edited by A. Leonardis, H. Bischof, and A. Pinz, chap. 38, pp. 490–503, Springer Berlin Heidelberg, Berlin, Heidelberg.

Olds, R. B. (2003), Marine mammals systems in support of force protection, in *SSC San Diego Biennial Review 2003*, chap. Chapter 3: Intelligence, Surveillance, and Reconnaissance, pp. 131–135, Space and Naval Warfare Systems Center, San Diego, San Diego, CA.

Olson, E., and P. Agarwal (2012), Inference on networks of mixtures for robust robot mapping, in *Proceedings of the Robotics: Science & Systems Conference*, Sydney, Australia.

Olson, E., J. Leonard, and S. Teller (2007), Spatially-adaptive learning rates for online incremental SLAM, in *Proceedings of the Robotics: Science & Systems Conference*, Atlanta, GA, USA.

Oskiper, T., Z. Zhu, S. Samarasekera, and R. Kumar (2007), Visual odometry system using multiple stereo cameras and inertial measurement unit, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8.

Ozuysal, M., M. Calonder, V. Lepetit, and P. Fua (2010), Fast keypoint recognition using random ferns, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, *32*(3), 448–461.

Paim, P., B. Jouvencel, and L. Lapierre (2005), A reactive control approach for pipeline inspection with an AUV, in *Proceedings of the IEEE/MTS OCEANS Conference and Exhibition*, pp. 201–206.

Pito, R. (1999), A solution to the next best view problem for automated surface acquisition, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, *21*(10), 1016–1030.

Pizarro, O., R. M. Eustice, and H. Singh (2009), Large area 3-D reconstructions from underwater optical surveys, *IEEE Journal of Oceanic Engineering*, *34*(2), 150–169.

Prentice, S., and N. Roy (2009), The belief roadmap: Efficient planning in linear POMDPs by factoring the covariance, *International Journal of Robotics Research*, *8*(11-12), 1448–1465.

Pronobis, A., B. Caputo, P. Jensfelt, and H. I. Christensen (2010), A realistic benchmark for visual indoor place recognition, *Robotics and Autonomous Systems*, *58*(1), 81–96.

Redheffer, R. (1946), *Elementary Theory of Transmission and Reelection: Fundamental Relations and Geometry*, Defense Technical Information Center.

Reed, M. K., and P. K. Allen (2000), Constraint-based sensor planning for scene modeling, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, *22*(12), 1460–1467.

Ridao, P., M. Carreras, D. Ribas, and R. Garcia (2010), Visual inspection of hydroelectric dams using an autonomous underwater vehicle, *Journal of Field Robotics*, *27*(6), 759–778, special Issue: State of the Art in Maritime Autonomous Surface and Underwater Vehicles, Part 1.

Rife, J., and S. Rock (2001), Visual tracking of jellyfish in situ, *Proceedings of the International Conference on Image Processing*, *1*, 289–292.

Robertson, S. (2004), Understanding inverse document frequency: On theoretical arguments for idf, *Journal of Documentation*, *60*, 503–520.

Roman, C., and H. Singh (2005), Improved vehicle based multibeam bathymetry using sub-maps and SLAM, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2422–2429, Edmonton, Alberta, Canada.

Rossell, H., L. Chapman, S. of Naval Architects, and M. E. (U.S.) (1941), *Principles of Naval Architecture*, no. v. 1 in Principles of Naval Architecture, Society of Naval Architects and Marine Engineers.

Rosten, E., R. Porter, and T. Drummond (2010), Faster and better: A machine learning approach to corner detection, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, *32*, 105–119.

Rublee, E., V. Rabaud, K. Konolige, and G. Bradski (2011), ORB: An efficient alternative to SIFT or SURF, in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2564–2571.

Russell, S. J., and P. Norvig (2003), *Artificial Intelligence: A Modern Approach*, Pearson Education.

Saez, J., A. Hogue, F. Escolano, and M. Jenkin (2006), Underwater 3D SLAM through entropy minimization, in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3562–3567.

Saigol, Z., R. Dearden, J. Wyatt, and B. Murton (2009), Information-lookahead planning for AUV mapping, *Proceedings of the International Joint Conference on Artifical Intelligence*, pp. 1831–1836.

Salton, G., and C. S. Yang (1973), On the specification of term values in automatic indexing, *Journal of Documentation*, *29*, 351–372.

Shahbazi, H., and H. Zhang (2011), Application of locality sensitive hashing to realtime loop closure detection, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1228–1233.

Shannon, C. E. (1948), A mathematical theory of communication, *The Bell System Technical Journal*, *27*, 379–423, 623–656.

Shi, J., and C. Tomasi (1994), Good features to track, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 593–600.

Shwartz, S., E. Namer, and Y. Y. Schechner (2006), Blind haze separation, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, *2*, 1984–1991.

Silva, V. D., and J. B. Tenenbaum (2002), Global versus local methods in nonlinear dimensionality reduction, *Advances in Neural Information Processing Systems*, pp. 705–712.

Sim, R. (2005), Stable exploration for bearings-only SLAM, in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2411–2416.

Sim, R., and N. Roy (2005), Global a-optimal robot exploration in SLAM, in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 661–666, Barcelona, Spain.

Singh, H., J. Howland, and O. Pizarro (2004), Advances in large-area photomosaicking underwater, *IEEE Journal of Oceanic Engineering*, *29*(3), 872–886.

Sivic, J., and A. Zisserman (2003), Video google: A text retrieval approach to object matching in videos, in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1470–1477.

Smith, R., M. Self, and P. Cheeseman (1988), A stochastic map for uncertain spatial relationships, in *Proceedings of the International Symposium on Robotics Research*, pp. 467–474, MIT Press.

Smith, R., M. Self, and P. Cheeseman (1990), Estimating uncertain spatial relationships in robotics, in *Autonomous Robot Vehicles*, edited by I. Cox and G. Wilfong, pp. 167–193, Springer-Verlag.

Snavely, N., S. M. Seitz, and R. Szeliski (2006), Photo tourism: Exploring photo collections in 3D, in *Proceedings of the International Conference and Exhibition on Computer Graphics and Interactive Techniques*, pp. 835–846, ACM, New York, NY, USA.

Stachniss, C., G. Grisetti, and W. Burgard (2005), Information gain-based exploration using rao-blackwellized particle filters, in *Proceedings of the Robotics: Science & Systems Conference*, Cambridge, MA, USA.

Strasdat, H., J. M. M. Montiel, and A. J. Davison (2012), Visual SLAM: Why filter?, *Image Vision Comput.*, *30*(2), 65–77.

Snderhauf, N., and P. Protzel (2011), BRIEF-Gist—closing the loop by simple means., in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1234–1241.

Teledyne RD Instruments (2008), Explorer DVL, `http://www.rdinstruments.com/explorer.html`.

Tenenbaum, J. B., V. D. Silva, and J. C. Langford (2000), A global geometric framework for nonlinear dimensionality reduction, *Science*, *290*(5500), 2319–2323.

Thrun, S., Y. Liu, D. Koller, A. Ng, Z. Ghahramani, and H. Durrant-Whyte (2004), Simultaneous localization and mapping with sparse extended information filters, *International Journal of Robotics Research*, *23*(7-8), 693–716.

Thrun, S., W. Burgard, and D. Fox (2005), *Probabilistic Robotics*, The MIT Press.

Toldo, R., U. Castellani, and A. Fusiello (2009), A bag of words approach for 3d object categorization, in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 116–127, Springer-Verlag, Berlin, Heidelberg.

Tomasi, C., J. Zhang, and D. Redkey (1995), Experiments with a real-time structure-from-motion system, in *Proceedings of International Symposium on Experimental Robotics*, pp. 197–203.

Torr, P. (1998), Geometric motion segmentation and model selection, *Philosophical Transactions of the Royal Society of London*, *356*, 1321–1340.

Torr, P. (1999), Model selection for two view geometry: A review, in *Shape, Contour and Grouping in Computer Vision*, pp. 277–301, Springer.

Triggs, B., P. McLauchlan, R. Hartley, and A. Fitzgibbon (2000), Bundle adjustment – a modern synthesis, in *Vision Algorithms: Theory and Practice*, edited by W. Triggs, A. Zisserman, and R. Szeliski, LNCS, pp. 298–375, Springer-Verlag.

Trimble, G., and E. Belcher (2002), Ship berthing and hull inspection using the CetusII AUV and MIRIS high-resolution sonar, in *Proceedings of the IEEE/MTS OCEANS Conference and Exhibition*, vol. 2, pp. 1172–1175.

Tuytelaars, T., and L. Van Gool (2004), Matching widely separated views based on affine invariant regions, *International Journal of Computer Vision*, *59*(1), 61–85.

Urick, R. (1983), *Principles of Underwater Sound*, McGraw-Hill, Inc.

Vaganay, J., M. Elkins, S. Willcox, F. Hover, R. Damus, S. Desset, J. Morash, and V. Polidoro (2005), Ship hull inspection by hull-relative navigation and control, in *Proceedings of the IEEE/MTS OCEANS Conference and Exhibition*, pp. 761–766, Washington, D.C.

Vaganay, J., M. Elkins, D. Esposito, W. O'Halloran, F. Hover, and M. Kokko (2006), Ship hull inspection with the HAUV: U.S. Navy and NATO demonstrations results, in *Proceedings of the IEEE/MTS OCEANS Conference and Exhibition*, pp. 1–6, Boston, MA.

Vaganay, J., L. Gurfinkel, M. Elkins, D. Jankins, and K. Shurn (2009), Hovering autonomous underwater vehicle — system design improvements and performance evaluation results, in *Proceedings of the International Symposium on Unmanned UntetheredSubmersible Technology*, Durham, NH.

Valencia, R., J. Andrade-Cetto, and J. Porta (2011), Path planning in belief space with pose SLAM, in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 78–83.

Varma, M., and A. Zisserman (2009), A statistical approach to material classification using image patch exemplars, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, *31*(11), 2032–2047.

Vidal-Calleja, T., A. Davison, J. Andrade-Cetto, and D. Murray (2006), Active control for single camera SLAM, in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1930–1936.

Walter, M., F. Hover, and J. Leonard (2008), SLAM for ship hull inspection using exactly sparse extended information filters, in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1463–1470, Pasadena, CA.

Warren, C. (1990), A technique for autonomous underwater vehicle route planning, *IEEE Journal of Oceanic Engineering*, *15*(3), 199–204.

Weiss, L. G. (2011), Autonomous robots in the fog of war, *IEEE Spectrum*, *48*(8), 30–36.

Whaite, P., and F. Ferrie (1997), Autonomous exploration: Driven by uncertainty, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, *19*(3), 193–205.

Whitcomb, L., D. Yoerger, H. Singh, and J. Howland (1999), Advances in underwater robot vehicles for deep ocean exploration: Navigation, control and survery operations, in *Proceedings of the International Symposium on Robotics Research*, pp. 346–353, Snowbird, UT, USA.

Willcox, J., Y. Zhang, J. Bellingham, and J. Marshall (1996), AUV survey design applied to oceanic deep convection, in *Proceedings of the IEEE/MTS OCEANS Conference and Exhibition*, vol. 2, pp. 949–954.

Williams, S., and I. Mahon (2004), Simultaneous localisation and mapping on the Great Barrier Reef, in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1771–1776.

Wu, C. (2007), SiftGPU: A GPU implementation of scale invariant feature transform (SIFT), `http://cs.unc.edu/~ccwu/siftgpu`.

Wu, L., S. Hoi, and N. Yu (2010), Semantics-preserving bag-of-words models and applications, *IEEE Transactions on Image Processing*, *19*(7), 1908–1920.

Yang, Y., and O. Brock (2006), Elastic roadmaps: Globally task-consistent motion for autonomous mobile manipulation in dynamic environments, in *Proceedings of the Robotics: Science & Systems Conference*, Philadelphia, USA.

Zhang, Z., and Y. Shan (2001), Incremental motion estimation through local bundle adjustment, *Technical Report MSR-TR-01-54*, Microsoft Research.

Zheng, Y.-T., M. Zhao, Y. Song, H. Adam, U. Buddemeier, A. Bissacco, F. Brucher, T.-S. Chua, and H. Neven (2009), Tour the World: building a web-scale landmark recognition engine, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1085–1092.

Zuiderveld, K. (1994), Contrast limited adaptive histogram equalization, in *Graphics Gems IV*, vol. IV, edited by P. Heckbert, pp. 474–485, Academic Press, Boston.