Mapping of Complex Marine Environments using an Unmanned Surface Craft

by

Jacques Chadwick Leedekerken

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Computer Science and Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2011

Author ....
Department of Electrical Engineering and Computer Science
August 9, 2011

Certified by..
John Leonard
Professor of Mechanical and Ocean Engineering
Thesis Supervisor

Accepted by ....
Leslie A. Kolodziejski
Chairman, Department Committee on Graduate Theses

Mapping of Complex Marine Environments using an Unmanned Surface Craft

by

Jacques Chadwick Leedekerken

Submitted to the Department of Electrical Engineering and Computer Science
on August 9, 2011, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Computer Science and Engineering

Abstract

Recent technology has combined accurate GPS localization with mapping to build 3D maps in a diverse range of terrestrial environments, but the mapping of marine environments lags behind. This is particularly true in shallow water and coastal areas with man-made structures such as bridges, piers, and marinas, which can pose formidable challenges to autonomous underwater vehicle (AUV) operations. In this thesis, we propose a new approach for mapping shallow water marine environments, combining data from both above and below the water in a robust probabilistic state estimation framework. The ability to rapidly acquire detailed maps of these environments would have many applications, including surveillance, environmental monitoring, forensic search, and disaster recovery. Whereas most recent AUV mapping research has been limited to open waters, far from man-made surface structures, in our work we focus on complex shallow water environments, such as rivers and harbors, where man-made structures block GPS signals and pose hazards to navigation. Our goal is to enable an autonomous surface craft to combine data from the heterogeneous environments above and below the water surface – as if the water were drained, and we had a complete integrated model of the marine environment, with full visibility.

To tackle this problem, we propose a new framework for 3D SLAM in marine environments that combines data obtained concurrently from above and below the water in a robust probabilistic state estimation framework. Our work makes systems, algorithmic, and experimental contributions in perceptual robotics for the marine environment. We have created a novel Autonomous Surface Vehicle (ASV), equipped with substantial onboard computation and an extensive sensor suite that includes three SICK lidars, a Blueview MB2250 imaging sonar, a Doppler Velocity Log, and an integrated global positioning system/inertial measurement unit (GPS/IMU) device. The data from these sensors is processed in a hybrid metric/topological SLAM state estimation framework. A key challenge to mapping is extracting effective constraints from 3D lidar data despite GPS loss and reacquisition. This was achieved by developing a GPS trust engine that uses a semi-supervised learning classifier to ascertain the validity of GPS information for different segments of the vehicle trajectory. This eliminates the troublesome effects of multipath on the vehicle trajectory estimate, and provides cues for submap decomposition. Localization from lidar point clouds is performed using octrees combined with Iterative Closest Point (ICP) matching, which provides constraints between submaps both within and across different mapping sessions. Submap positions are optimized via least squares optimization of the graph of constraints, to achieve global alignment. The global vehicle trajectory is used for subsea sonar bathymetric map generation and for mesh reconstruction from lidar data for 3D visualization of above-water structures.

We present experimental results in the vicinity of several structures spanning or along the Charles River between Boston and Cambridge, MA. The Harvard and Longfellow Bridges, three sailing pavilions and a yacht club provide structures of interest, having both extensive superstructure and subsurface foundations. To quantitatively assess the mapping error, we compare against a georeferenced model of the Harvard Bridge using blueprints from the Library of Congress. Our results demonstrate the potential of this new approach to achieve robust and efficient model capture for complex shallow-water marine environments. Future work aims to incorporate autonomy for path planning of a region of interest while performing collision avoidance to enable fully autonomous surveys that achieve full sensor coverage of a complete marine environment.

Thesis Supervisor: John Leonard
Title: Professor of Mechanical and Ocean Engineering

Acknowledgments

This work was partially supported by the Office of Naval Research under Grants N00014-06-10043, N00014-05-10244 and N00014-07-11102, by the MIT Sea Grant College Program under grant 2007-R/RCM-20, and by Singapore's NRF via funding through the SMART / CENSAM Center (http://censam.mit.edu).

THIS PAGE INTENTIONALLY LEFT BLANK

Contents

List of Figures

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

List of Algorithms

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 1

Introduction

Recent technology has combined accurate GPS localization with mapping to build 3D maps in a diverse range of terrestrial environments, but the mapping of marine environments lags behind. This is particularly true in shallow water and coastal areas with man-made structures such as bridges, piers, and marinas, which can pose formidable challenges to autonomous underwater vehicle (AUV) operations. In this thesis, we propose a new approach for mapping shallow water marine environments, combining data from both above and below the water in a robust probabilistic state estimation framework. We describe an implemented system that performs large-scale Simultaneous Localization and Mapping (SLAM) using data streams from an autonomous surface vehicle equipped with GPS, sonar, and lidar sensing. Our results demonstrate the potential of this new approach to achieve robust and efficient model capture for complex shallow-water marine environments.

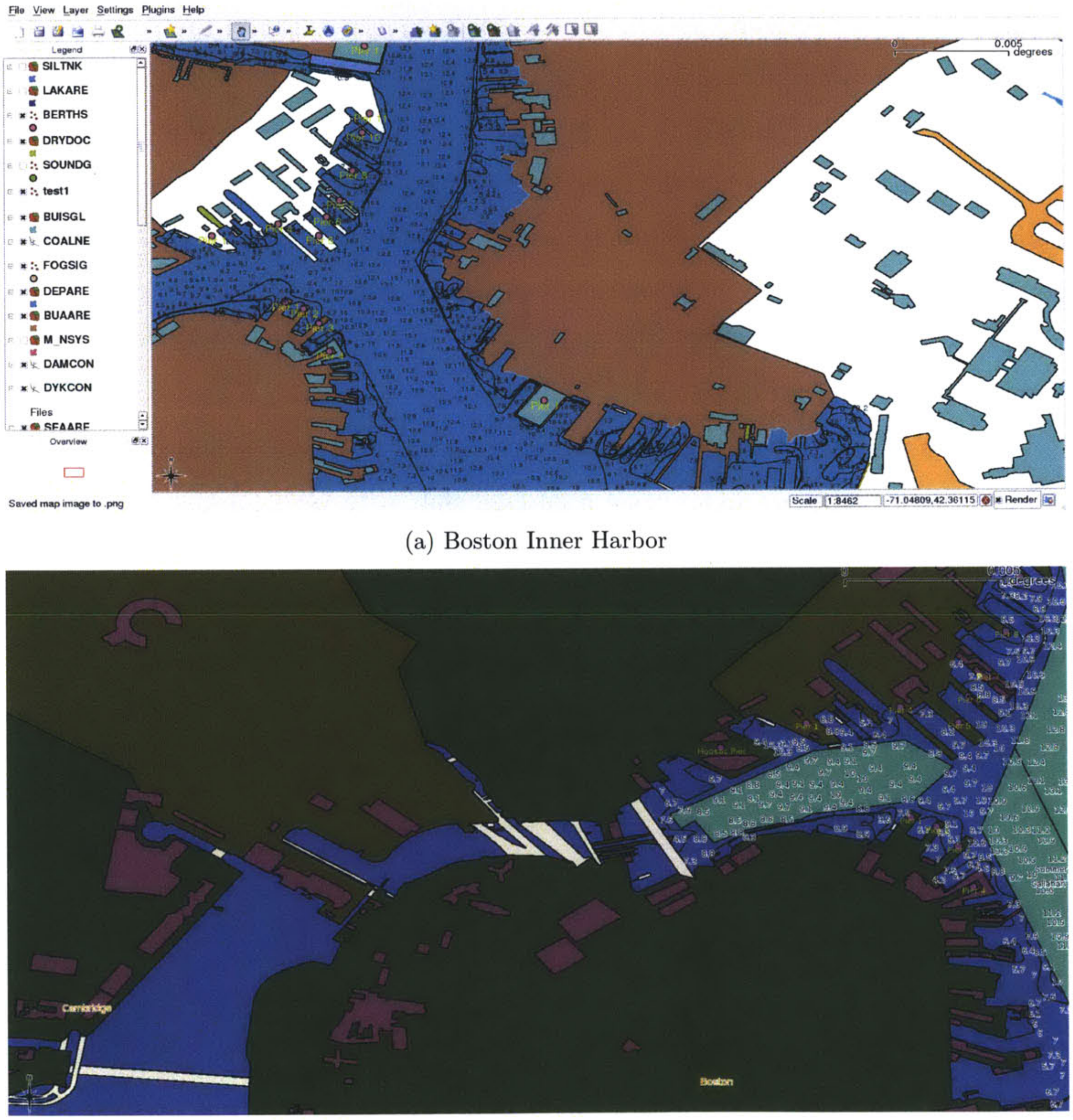
1.1 Research Objectives

Our goal is to enable an autonomous surface craft to create accurate 3D maps of complex marine environments, both above and below the water. Environments of particular interest are bridges, harbors, and other shallow marine environments with significant human use or construction. Typically such environments have no accurate digital map representation, nor are regular sonar surveys conducted. Underwater depth maps are often called bathymetry maps, and are analogous to terrain maps for terrestrial environments. Most bathymetric data are for major lakes, high-traffic coastal regions, and in the deep ocean beyond 100 meters depth. An example of a typical digital nautical chart (for Boston Harbor) is shown in Figure 1-1. While bathymetric data covers much of the world, the majority of undersea map data is decades old, and as shown in the figure, coverage is typically poor in coastal areas.

The environments that we consider in this thesis possess the following traits. The environment above the water should contain structures observable with the ranging sensors. The environment below the surface contains structures or bathymetry within the perceptual field of the platform sensors. For example, a river or harbor region with maximum depth of five meters and possessing surface structures, geological formations, or natural vegetation within or along the shore would be suitable for our platform (described below), which has a sonar with a maximum range of ten meters and surface laser ranging sensors with maximum range of thirty meters. Figure 1-2 illustrates examples of target environments within the domain of this research.

The ability to rapidly acquire detailed maps of these environments would have many applications, including surveillance, forensic search, and environmental monitoring. Unmanned vehicles may be used for environmental sampling [28], disaster response [106], forensic recovery [125], and surveillance [128]. Natural disasters motivate the need for marine robotic perception systems. Robotic systems may provide a means to locate human survivors and casualties, inspect structures for damage, and survey environments for recovery construction efforts without endangering humans. For example, Murphy et al. documented inspection of a bridge for damage with a robotic surface craft in the wake of a hurricane [106]. In such situations, small surface craft offer advantages of safety, maneuverability, and perception unavailable from other marine vehicles or aerial survey methods. While aerial surveys can provide shallow water bathymetry [163], the sampling is sparse and subject to occlusions such as bridges.

Our primary thesis claim is that bathymetric mapping may be improved relative to AUV bathymetry mapping using a surface craft with GPS sensing for accurate localization. Our work incorporates constraints from surface maps to provide greater qualitative consistency of surface maps, which demonstrate distortions from GPS errors when uncorrected. More generally, our method for improving localization of underwater maps applies to underwater sampling when the observations are inadequate for self-alignment. This could result from using a sonar with a limited field of view or for making point measurements of parameters such as temperature, salinity, or chemical concentrations.



Saved map image to .png

(a) Boston Inner Harbor

(b) Boston Inner Harbor to Charles River

Figure 1-1: Example views of multiple data layers of Digital Nautical Chart (DNC) data. Depth soundings are labeled in meters. (a) Part of Boston's inner harbor, including the North End, Charlestown, and East Boston; (b) Close-up of view of the Charles River near MIT and the inner harbor adjacent to the North End and Charlestown. The Longfellow Bridge is the linear structure in the lower left of the figure. Note the absence of depth data in the Charles River.

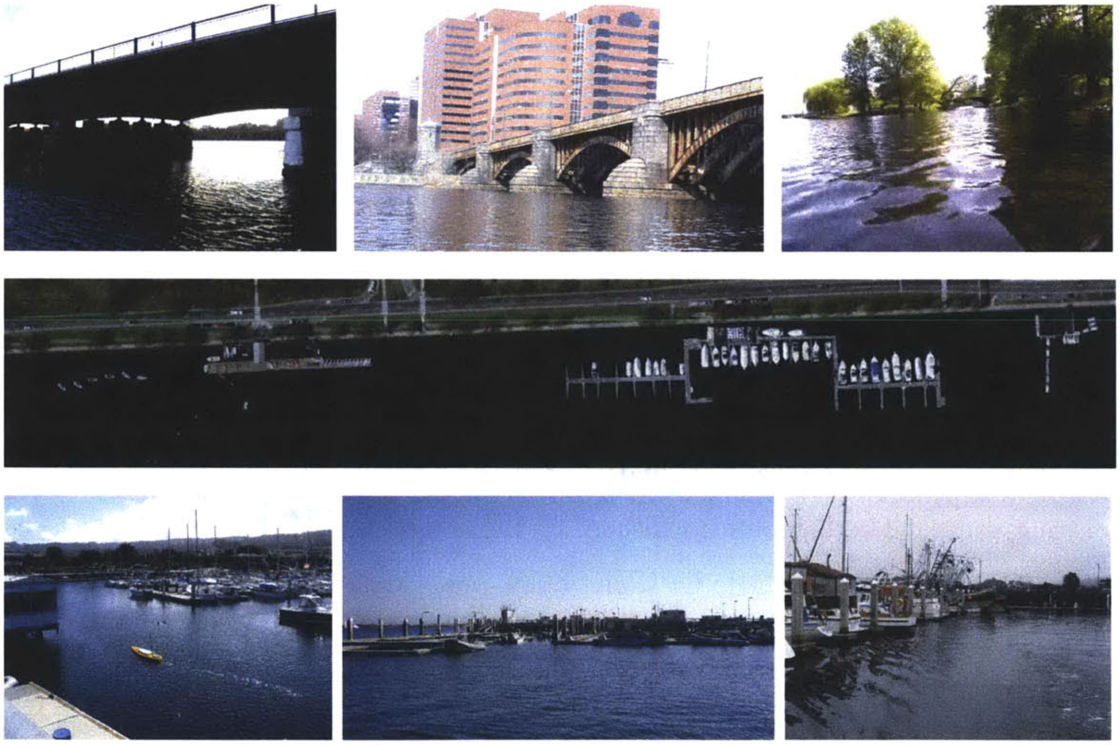


Figure 1-2: This work aims to map shallow marine environments such as those shown here: bridges spanning rivers and natural or man-made features along shores.

1.2 Technical Approach

Our technical approach for developing this unique capability — concurrent mapping above and below the water surface — has the following key elements: (1) design of a novel autonomous marine platform, equipped with GPS, inertial, sonar, and lidar sensors; (2) development of a machine learning approach to classifying GPS errors that are encountered in close proximity to marine structures; and (3) the realization of a robust and efficient 3D simultaneous localization and mapping (SLAM) algorithm to process the streams of heterogeneous data generated by the vehicle's sensors.

Our platform is shown in Figure 1-3. For positioning and localization, the vehicle has an OXTS integrated IMU and GPS and an RDI Doppler Velocity Log (DVL). The terrestrial sensors include three SICK LIDARs, a low-cost web cam, and two Hokuyo LIDARs mounted on a shaft rotating at approximately 1Hz. Two of the SICK LIDARs are mounted to scan vertically to permit 3D reconstruction as the vehicle passes along features. For underwater sensing we mount a Blueview MB2250 imaging sonar. The computational core of the vehicle contains two motherboards carrying a quad-core 2.83 GHz CPU and a low-power dual-core 1.6 GHz CPU.

The platform evolved through an iterative design process, and sensors were added and removed through the life of the project. The camera and a rotating laser scanner shown in Figure 1-3 were logged but not used in the results reported in the thesis. Further details on the vehicle design and a discussion of the various sensor configurations that were evaluated during the course of the research is provided in Appendix B.

The data from these sensors is processed in a hybrid metric/topological SLAM state estimation framework. From an algorithmic perspective, tackling this problem forces us to address a number of difficult issues not encountered in previous SLAM research. Operation at the water surface presents unique challenges for robotics. Wind and current forces induce unpredictable platform motion, posing modeling and

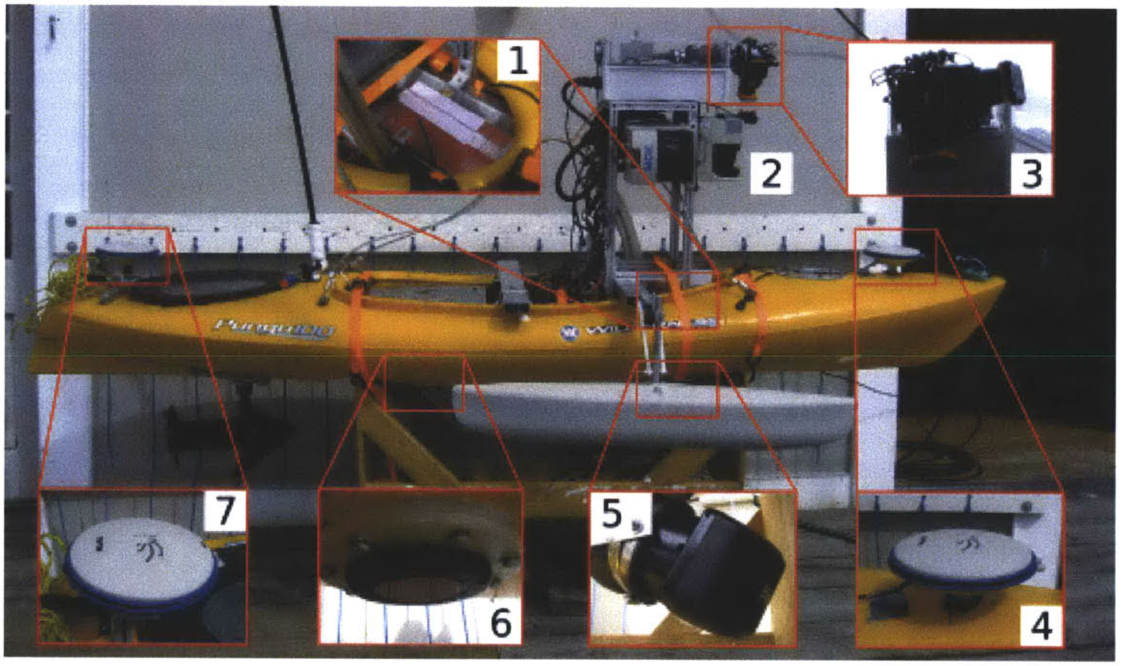


Figure 1-3: The marine mapping platform developed for the thesis research project, shown with close-up images of the primary sensors. These include the Oxford Inertial+ IMU and GPS (inset 1, 4, and 7), an RDI Doppler Velocity Log (6), three SICK LMS291 laser scanners (2), a Blueview MB2250 imaging sonar (5). Also shown are two Hokuyo UTM30LX laser scanners on a continuously rotating shaft (3), for which data was collected, but not processed in the thesis.

control challenges. Unstructured terrain, sparse landmarks, and GPS distortions near man-made structures make navigation difficult and increase uncertainty in control and coverage capabilities.

A key challenge to mapping is extracting effective constraints from 3D lidar data despite GPS loss and reacquisition. We wish to make effective use of GPS information, when appropriate, but to not be fooled by the GPS errors that inevitably occur in close proximity to marine structures. This was achieved by developing a GPS trust engine that uses a semi-supervised learning classifier to ascertain the validity of GPS information for different segments of the vehicle trajectory. This eliminates the troublesome effects of multipath on the vehicle trajectory estimate, and provides cues for submap decomposition. Localization from lidar point clouds is performed using octrees combined with Iterative Closest Point (ICP) matching, which provides constraints between submaps both within and across different mapping sessions. Submap positions are optimized via least squares optimization of the graph of constraints, to achieve global alignment. The global vehicle trajectory is used for subsea sonar bathymetric map generation and for mesh reconstruction from lidar data for 3D visualization of above water structures.

The primary target in the environment for our work is the Harvard Bridge (also known as the Mass Ave. Bridge), which spans the Charles River at MIT, connecting Boston and Cambridge. A particular challenge problem for this work was to obtain a complete model of the supporting structures of the Harvard Bridge, which would entail frequent transits in and out of GPS availability, posing a great challenge to sensor processing. The Harvard Bridge occupies a special place in MIT folklore, as described by a plaque placed on the bridge in 2008 by the MIT class of 1962. The plaque reads:

> "In October 1958 the span of this bridge was measured using the body of

> Oliver Reed Smoot MIT '62 and found to be 364.4 smoots +/- 1 ear. This provided a very useful metric for generations of bridge travelers since, and the Smoot has joined the angstrom, meter, and light-year as enduring standards of length."
>
> Dedicated to our Classmate Oliver Reed Smoot '62 on October 4, 2008 by the MIT Class of 1962.

1.3 Related Research

The idea of using computer vision and GPS measurements to capture large-scale urban environments has been the subject of considerable research in the past. A notable early system was developed by Teller and colleagues at MIT, in the City Scanning project [4, 151]. More recently, one of the most renowned applications of 3D mapping by a commercial enterprise is the Google Street View project [67]. The Google Street View project launched in 2007 using automobiles for environmental capture along navigable streets. With integration to their mapping technology, the project permits users a virtual street-level perspective of geographic locations. While many of the details of the Street View project are proprietary, the project clearly shows the tremendous utility to users of obtaining detailed map data for the world.

In an effort to capture more diverse environments beyond easily navigable streets, Google has constructed a variety of alternative platforms, such as bicycles and snowmobiles, shown in Figure 1-4. The configuration of laser range sensors on Google's platforms are quite similar to our vehicle, shown in Figure 1-6 and described in detail in Appendix B. The approach that we present in this thesis could provide a similar capability to Street View or Microsoft's Bing Maps, for shallow-water coastal marine environments. For example, significant portions of cities such as Venice and Amsterdam, with their substantial waterway systems, could be mapped in great detail using the 3D mapping framework that we describe in this work.

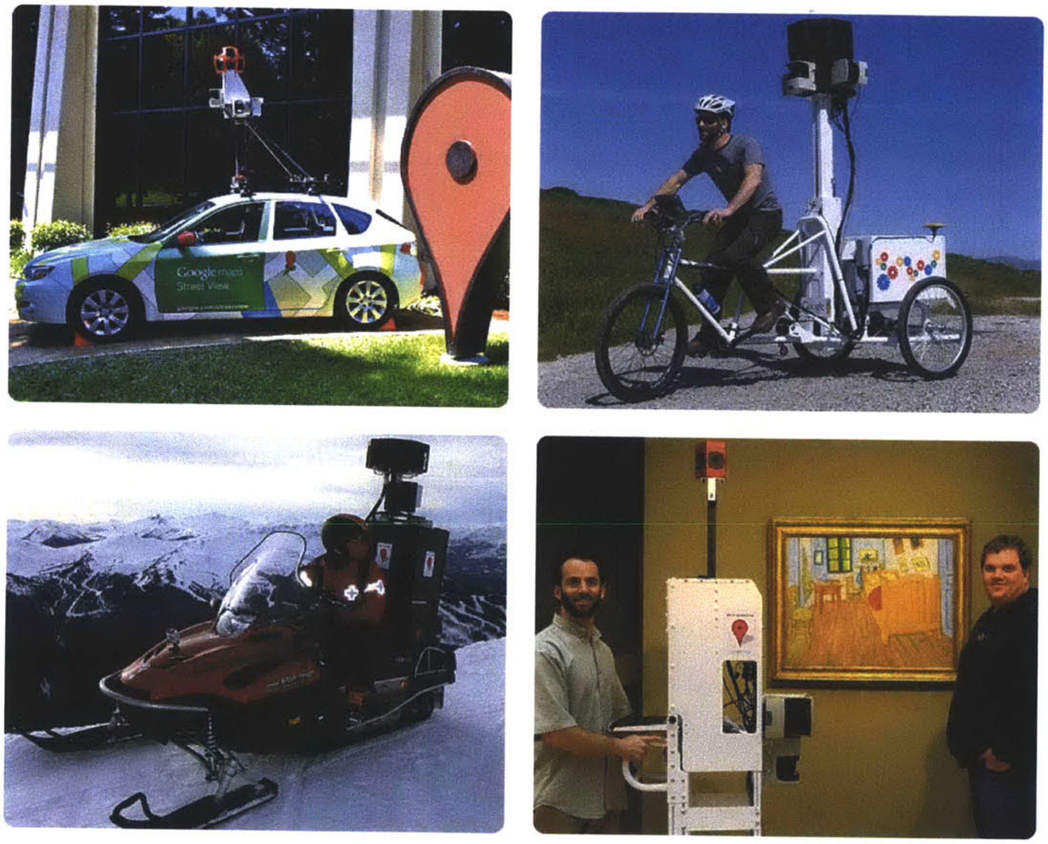


Figure 1-4: The Google Street View project employs a variety of platforms for capturing environments in 3D [67].

The work presented in this thesis touches upon a variety of research topics in robotics: perception, sensor fusion or SLAM optimization, marine applications, submapping, point registration, and surface reconstruction. There has been substantial research on these topics, and for terrestrial domains there exist deployed commercial systems that can efficiently capture large environments using airborne and land vehicles for data capture. However, the use of these technologies in the marine environment lags behind terrestrial applications, due to a number of difficult issues. These include platform motion, limited sensor range, massive data streams, limited onboard computation, perceptual aliasing due to repeated environmental structure, erroneous GPS readings, ambiguous data association, and outlier measurements. The elusive goal is robustness. When erroneous constraints are applied within conventional approaches, the result can easily lead to non-convergence or incomprehensible results.

Unlike terrestrial robots, for a marine robot the assumption of static support is invalidated, so vehicle motion is possible in the absence of control inputs. The robot moves within a fluid and is subject to current, wind, and wave disturbances. This makes the issue of modeling platform motion much more difficult than for wheeled land vehicles. Similar issues can be encountered with unmanned air vehicles [7]. Under these conditions a robot is rarely stationary even when not using any actuators. In fact station keeping is a challenge for many marine vehicles, especially when underactuated, i. e. unable to directly control motion of all degrees of freedom. What is more relevant for estimation when static support is invalidated is the predictive modelling of vehicle dynamics when a vehicle is subject to motion induced by forces unrelated to either control inputs or *a priori* parameters, such as gravity or predetermined buoyancy. The environmental forces weaken the accuracy of simpler predictive control models, and consequently reduce mapping accuracy. Ameliorating such error may be accomplished with increasingly complex models of environmental forces or a combination of more accurate motion sensors and robust estimation algorithms.

Underwater perception is difficult compared to terrestrial sensing. Part of this is due to technology limits, since sensor technology for vision and laser or radar ranging are in much higher demand. Underwater sensing is primarily limited to acoustics, although some underwater vision is possible at close range in calm, clear water. Acoustic sensing is a substantially different mechanism when compared to visual- or laser-based sensing [12,61]. While underwater mapping may benefit from improvements in sonar sensors or advanced processing techniques [100, 110, 130], we propose an alternative mechanism to aid underwater mapping using constraints derived from simultaneous surface sensing.

GPS can provide good position accuracy even with low-cost single antenna devices. Applications of GPS technology have recently increased, and GPS sensors may be found in consumer portable audio players, mobile phones, electronic readers, wrist watches, etc. The majority of applications do not require very high accuracy. For example, anti-theft GPS tracking devices in automobiles need only coarse positioning.

Professional surveys and military applications do require higher levels of accuracy, and the systems and technology required to achieve greater accuracy rapidly increase in cost. Even with expensive equipment, the underlying mechanisms of GPS are susceptible to many sources of error which can be difficult to model or detect. Structural conditions, such as canopies or canyons whether natural or man-made, corrupt GPS ranging through multipath or signal obstruction.

This thesis does not attempt to model complex GPS parameters, or to perform signal-level error detection. Instead, we present a framework to detect when common GPS hazards are most likely to occur, such as canopy or canyon situations, using other sensor data available to the robot. While there clearly has been a large body of work in SLAM in the past decade [9, 47, 153], the marine mapping task posed in this thesis presents several challenges for SLAM algorithm development not addressed in previous research. Integrating GPS with SLAM can be a surprisingly difficult issue [30], especially in our target environment, where GPS multipath errors are extremely common.

A common tradeoff in navigation and mapping system design is local versus global consistency [23]. Locally consistent maps provide accurate relative relations between the true features in the world, although a local region may be inaccurately positioned in the global frame. Local consistency is usually due to limited sensing range, the use of relative motion sensors such as inertial measurement units or wheel encoders, and predictive (relative) motion models [103]. One may observe gradual loss of global consistency as drifts and biases corrupt the estimated map; the resulting error is commonly referred to as odometry drift or dead-reckoning error. Efforts to achieve greater global consistency often disrupt local consistencies. Global corrections may manifest as loop closures, sparse GPS fixes, or sparse features with known data association. In incremental recursive estimators, global corrections can produce "kinks" in the trajectory, breaking consistency of the map before and after the correction event. Recent work in full trajectory and map optimization and smoothing attempts to distribute the error [43, 119].

Related platforms for this research include a wide range of autonomous surface vehicles (ASVs) and autonomous underwater vehicles (AUVs). Historically, ASV research has attracted less interest from academic researchers, in comparison to AUV development. However, in recent years, there has been a surge in interest in robotic surface vehicles. Bertram [21] and Manley [98] provide recent surveys of notable ASV platforms. Many military surface vehicles are manned platforms that have been modified for autonomous operation. The US Navy Unmanned Surface Vehicle Master Plan, published in 2007, lists a wide range of potential surface vehicle missions, including: mine countermeasures, UUV delivery, maritime security, surface warfare, special operations forces support, electronic warfare, and maritime interdiction operations support [109]. To our knowledge, the capability that we have set out to achieve – integrated mapping above and below the water with an ASV – has not previously been attempted in any of this literature.

Our platform is most similar to the SCOUT (Surface Craft for Oceanographic and Underwater Testing) vehicles [40]. The SCOUT vehicles offer relatively low-cost alternatives to traditional truly submersible robots. The low cost and use of commercial-off-the-shelf (COTS) components permitted rapid deployment of multiple vehicles, as demonstrated in [15]. The primary research applications of the SCOUT have been in testing marine autonomy [16,17] and navigation assistance for autonomous underwater vehicles [8,41,157]. Our platform inherits the hull design of the SCOUT vehicle, and our work extends the perceptual and computational capabilities used in prior work. While the SCOUT hull design is a COTS recreational kayak, other researchers have noted catamaran type hulls offer stability benefits [95,97,149].

In the most recent work with autonomous surface vehicles similar to our work, Huntsberger et al. present a robotic boat for navigation and target following [76]. Their system integrates GPS, IMU, stereo vision, radar sensors to generate maps of objects above the waterline. The objective in their work is to generate a two-dimensional hazard map for obstacle avoidance at relatively high speeds; precise map estimation for environmental reconstruction is not their stated objective. Underwater perception such as sonar mapping is beyond the scope of their application. They implement their perception and navigation system on several platform hulls with dimensions approximately 13 to 14 meters in length. The physical size is much larger than our platform (approximately 2 meters) and permits operation in rougher sea states and at higher speeds. Interestingly, the authors note laser ranging sensors to be ineffective for their application, and they state wave motion effects, limited range, and limited angular resolution at longer ranges as factors for not incorporating lasers into their sensor suite.

In an earlier vision-based navigation application, Subramanian et al. present results demonstrating shoreline tracking [149]. Their platform included typical navigational sensors such as a GPS receiver, an IMU, and Doppler Velocity Logger (DVL) in addition to a camera sensor for the shoreline tracking.

Recent research with significant similarity to ours in terms of application environment and the use of unmanned surface craft with inspection abilities is shown in [106, 107]. Their platform employed a twin-hull catamaran design. As the authors note, the catamaran design is more stable than the SCOUT hull in high currents, and our platform employed a pair of stabilizing pontoons. The primary objectives in Murphy's work were inspection and mapping of a bridge structure and debris field in the aftermath of a hurricane in a shallow coastal region. A unique aspect of Murphy's work is the experimental comparison of unmanned underwater vehicles (an ROV and AUV) with a surface vehicle in a cluttered environment. The authors report a collision rate of 40% with the underwater vehicles, and achieved their primary objective primarily with the surface craft. The tether on the ROV presented an additional hazard encountered experimentally. In their post experimental analysis on disaster response feasibility and human-robot interaction, the authors note the control challenge presented by GPS loss near the bridge structure and manual teleoperation required near the structure. Murphy provides further analysis on the human-robot interaction requirements in their application. Multiple displays and a team of experts required coordination to successfully deploy and monitor the status of the robot, interpret the sensor data for the inspection role, and navigate successfully near obstacles. Summarily, Murphy's research presented a real scenario motivating the use of robotic vehicles in cluttered marine environments, the challenges associated with the objective, and provided operational comparison of robotic surface and subsurface vehicles within the target.

Underwater and surface vehicle technology has evolved over the last two decades enough for autonomous surveys to be possible over a reasonably sized region and with reduced cost [3]. With the development of the SCOUT [40], costs can be further reduced using comparatively cheaper surface craft having greater flexibility in sensor attachment. Underwater vehicles face tighter engineering constraints limiting their flexibility to accommodate new sensor technology. Figure 1-5(a) shows two AUVs and a SCOUT vehicle en route to deployment for a mine hunting mission. The larger vehicle on the left is a REMUS model AUV, a vehicle developed for shallow water operations [2] with applications to coastal mine countermeasures [3,147]. The smaller AUV on the right is an iRobot Ranger, which is a less expensive vehicle developed for mine acquisition [36,64]. Due to the complex engineering of such vehicles, integrating new sensors such as the sonars on the AUVs shown in Figure 1-5(a) is a non-trivial task requiring trained experts. In contrast, surface craft provide much easier integration of sensors, such as the simple rig shown in Figure 1-5(b). Figure 1-5(b), shows a SCOUT vehicle equipped with two imaging sonars after a previous deployment with a sidescan sonar towfish.

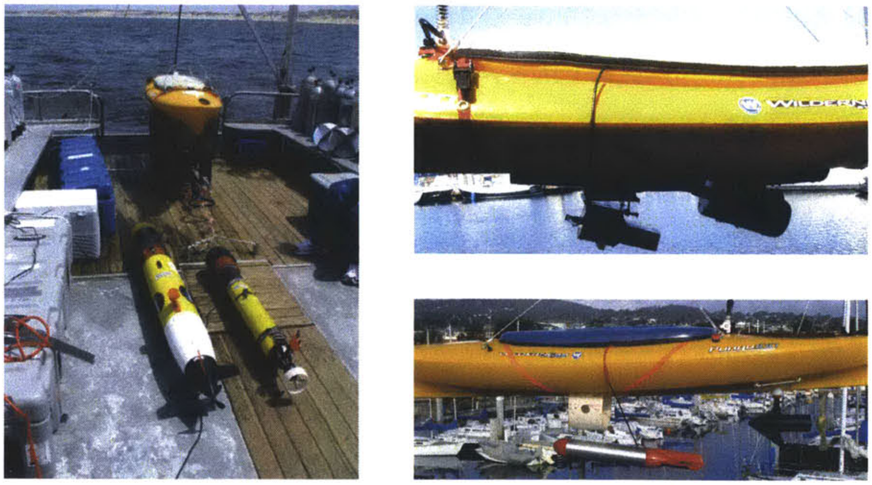


Figure 1-5: At left, two small autonomous underwater vehicles (AUVs) and a SCOUT autonomous surface craft (ASC) prior to deployment. A REMUS vehicle is shown to the left of the smaller lower cost iRobot Ranger. At right, simple mounting rigs enable the SCOUT autonomous surface craft (ASC) to rapidly deploy new sensors, such as a multiple imaging sonars (top) or a sidescan sonar (bottom).

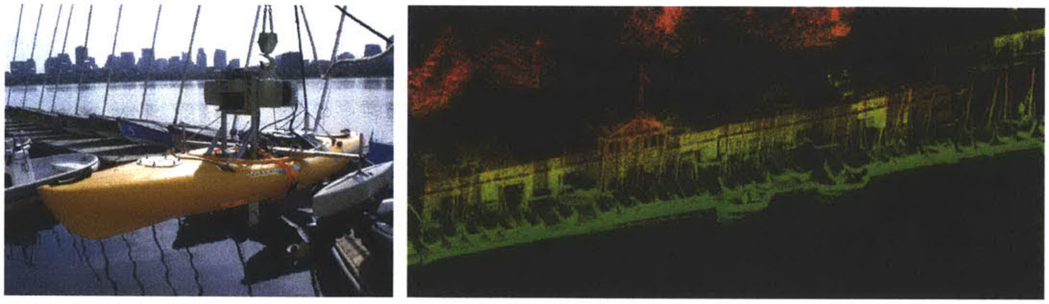


Figure 1-6: Integrating a rapidly deployable low-cost platform with accurate sensors and high performance computing motivates new opportunities in robotic mapping of marine environments. The platform at left permits the rapid survey of a sailing pavilion on the right.

1.4 Contributions

The primary contributions of this thesis are threefold. First, we have developed a novel approach to improving the quality of subsea mapping using surface sensor data, via creation of an autonomous platform with sensors both above and below the waterline. Second, we have developed a new approach for GPS trust assessment using perceptual data fed into a semi-supervised learning classifier. Third, we have implemented a large-scale hierarchical 3D mapping system that combines multiple mapping sessions into a network of submaps whose global positions are optimized with an efficient pose graph optimization algorithm. We have validated our experimental results by comparing with a historical model for the Harvard Bridge, providing a quantitative analysis of mapping error. The overall 3D mapping system that we have implemented compares well with other recent results from the 3D SLAM literature, and provides insights for a number of interesting future research areas, including the integration of perception with real-time motion control for coverage and obstacle avoidance.

Highlights of our results are shown in Figures 1-7 and 1-8, which demonstrate 3D mapping results for structures in the Charles River near MIT, with a focus on the Harvard Bridge. Critical to map consistency is the use of a GPS trust engine, described in Chapter 3, to adapt localization models as GPS hazards are encountered. Figure 1-8 illustrates a typical GPS hazard condition under a bridge and the utility of the classifier modes in trajectory optimization for addressing the hazard. The images show range data projected from the estimated trajectory with and without the aid of the classifier modes. The data projected with the aid of the classifier is more consistent with the true structure.

An additional use for the techniques developed in this thesis may be to provide quantitative assessment of subsea mapping algorithms. By mapping an environment both above and below the water surface, this thesis presents a novel approach to improving the accuracy of bathymetric maps when no ground truth is available due to the simultaneous capture of surface and subsurface maps. Accuracy of surface maps can be assessed quantitatively if ground truth is available. For some applications, qualitative assessment may be sufficient, as people are sensitive to even minor errors in map estimates of objects they can see themselves. An observer may inspect both surface and subsurface maps and readily infer the relative distribution of error. For example, a surface map of high accuracy but having small distortions in a small region would imply the bathymetry in that region is less accurate than other regions.

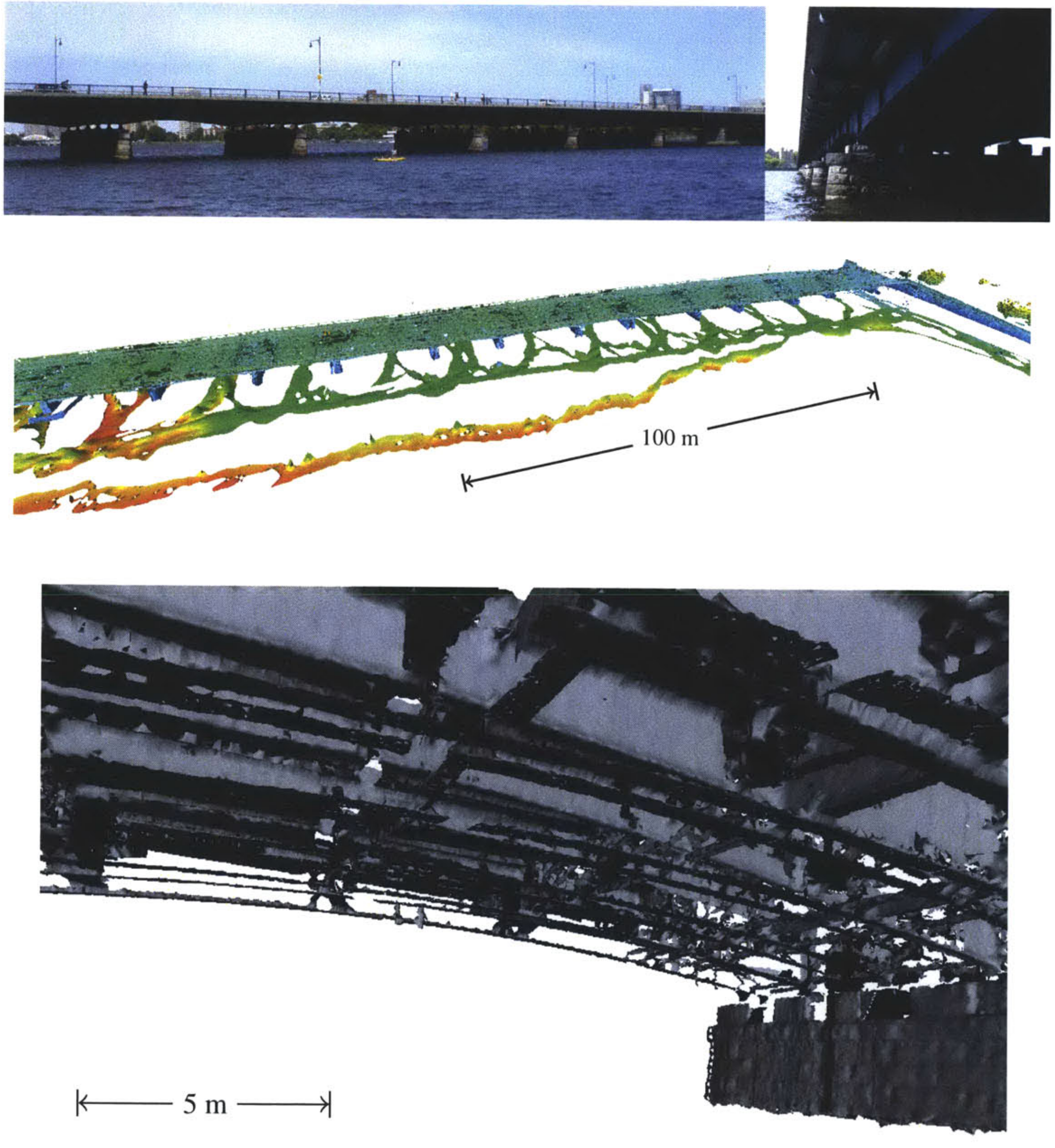
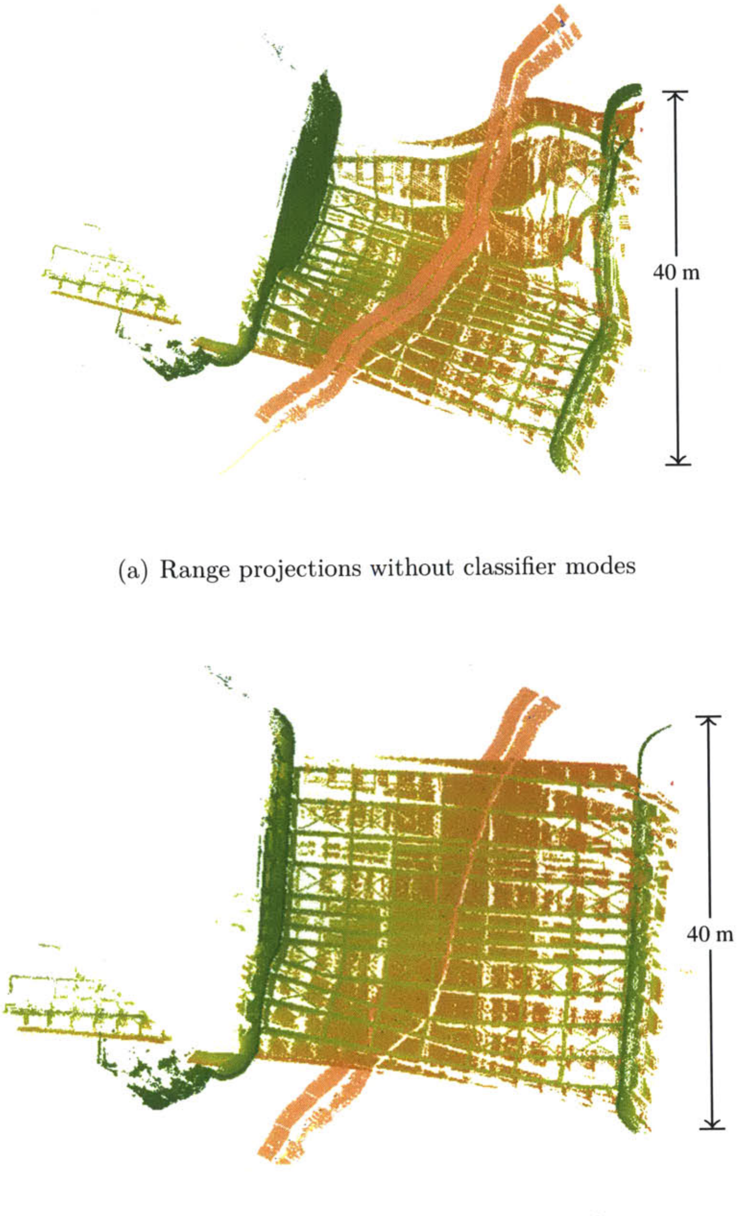


Figure 1-7: The camera images at top show the Harvard Bridge with views similar to the mapping results shown here. The center image shows mapping results with half of the bridge in view adjacent to the Cambridge stone retaining wall on the right side. The final image shows the map detail provided underneath one segment of the bridge.



(a) Range projections without classifier modes

(b) Range projections with classifier modes

Figure 1-8: The images compare projections of laser and sonar range data while the vehicle traversed under a bridge. The vehicle starts at the bottom of the image and travels toward the top along the track in the tan colored sonar points. Laser points are false colored by height. In the top image, poor GPS conditions distort the trajectory and the projected range data. In the lower image, the classifier-aided trajectory and range data are more consistent with the true bridge structure.

1.5 Summary

This chapter has motivated the autonomous marine mapping problem that is the focus of the thesis, presented our technical approach and the new robotic platform developed for this investigation, and has described some of the challenges of tackling this problem. The structure of the remainder of the thesis is as follows: Chapter 2 defines the 3D mapping problem, reviews previous research in 3D SLAM, and highlights open issues faced in the complex marine environments considered in this thesis. Chapter 3 describes a new technique for robustly incorporating GPS sensor measurements while operating in cluttered environments, using a semi-supervised learning classifier. Chapter 4 describes the major new algorithmic contributions of the thesis, presenting a hierarchical submap/pose-graph mapping algorithm that achieves robustness to degraded GPS and poor environmental observability with the aid of the GPS classification engine described in Chapter 3. Chapter 5 presents experimental results for multi-session mapping of part of the Charles River basin. Validation is performed against historical models of the Harvard Bridge piers. Chapter 6 summarizes our contributions and makes suggestions for future research. Appendices A and B provide information on the ground truth comparison for the Harvard Bridge and details of the construction of our marine sensing platform.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 2

The 3D Mapping Problem

This chapter defines the 3D marine mapping problem formally and reviews the relevant literature from the SLAM community. While the previous research in SLAM considered broadly is vast, the 3D mapping task domain presents great difficulties for current state-of-the-art algorithms. We give the probabilistic framework defining the 3D map estimation problem, and we discuss the solution of the estimation problem with emphasis on state-of-the-art least squares algorithms.

2.1 Probabilistic State Estimation for Mapping

At the core of many robotic estimation systems is a sensor fusion or Simultaneous Localization and Mapping (SLAM) framework. A standard approach is the use of a Kalman Filter [80] or Extended Kalman Filter (EKF). Numerous examples of EKF approaches to the sensor fusion and SLAM problems exist [31, 46, 94]. Julier and Uhlmann in [78] presented the Unscented Kalman Filter (UKF), which uses nonlinear transformations of selected sample points to avoid some issues with accumulated linearization error. Later work by Thrun [154] and Walter [158] emphasized the sparsity of the dual form of the EKF, the Extended Information Filter. Sparsity refers to the information matrix having mostly zero entries off the diagonal and is most apparent when robot poses are not marginalized. Although the state space grows over time leading to $O(n^2)$ growth of the covariance or information matrix, the sparse problem structure permits use of efficient representations and sparse optimization methods. Compared to marginalization in traditional recursive implementations such as the EKF, maintaining the entire trajectory permits recalculation of linearization points for nonlinear models. More recent research has focused on optimization of the sparse problem and providing more emphasis on graphical models of problem. Examples include Graphical SLAM [62], square root Smoothing and Mapping (SAM) [42], pose graph optimization [119], and gradient descent methods such as in [44,69].

Other mapping and localization research has focused on particle filtering, which is a stochastic approach to the problem. A key insight by Montemerlo in [101] was that given a robot pose, landmarks are conditionally independent. This allows landmarks to be represented by independent filters without maintaining covariances or co-information between landmarks. The efficiency tradeoff for particle filters is maintaining sufficient particle diversity and not maintain highly improbable particles. Resampling procedures prune particle sets at the risk of depletion, where the set shares a common ancestry. Depletion prevents new observations, especially loop closures, from affecting feature positions prior the common ancestry point. While particle filters reduce the dimensions for the state representation, the state must be replicated over the n particles. For small environments with landmarks of small dimension, one may use a very large number of particles for a system robust to non-Gaussian distributions. Map representation may be a critical factor in the ability to apply particle filtering to a problem. In evidence grid approaches the memory requirements to maintain particle set size for a non-trivial environment quickly become large. Intelligent approaches to maintaining an effective particle set were shown in [68]. Improving memory usage with clever use of data structures allowing sharing was shown in [51,52]. Overall particle filters provide a robust estimation framework using randomized sampling and work well on modest environment scales and low dimensional representations. As environments grow larger and map representation dimensionality increases, the cost of maintaining a sufficient set of particles without rapid depletion becomes computationally intractable. If trajectories contain many large or nested cycles, the ability to apply shared data structure optimizations fades. Shared data optimizations work best for particle sets having common ancestry, and particle filters are more robust when they exhibit particle diversity.

Probabilistic Formulation

As mentioned in Chapter 1, we have designed a vehicle equipped with sensors providing observability of vehicle position and partial views of the environment relative to the vehicle. Real sensors are imperfect, and measurements are corrupted by noise. To deal with noisy measurements, we introduce an estimation framework to approach the mapping problem from a probabilistic perspective.

The general probabilistic formulation for map estimation is as follows. The objective is to determine the values for estimated states maximizing the joint probability of all states. For the purposes of this thesis, we consider only Gaussian error distributions in the vehicle prediction and sensor models, which is a widely adopted assumption underlying most solution frameworks. Consider the set of estimated states to be X, where X typically includes vehicle poses, such as a 3D pose $x = [x, y, z, \phi, \theta, \psi]^T$. Implementations often include a representation of the map with landmarks and other estimated parameters such as bias states such that X may be a set containing landmarks m and bias estimates b : $X = \{x_0, x_1, \ldots, x_N, m_0, \ldots, m_N, b_0, \ldots, b_L\}$. Consider the observable inputs to the system to be the set C, where C includes sensor measurements and control inputs each having an uncertainty from additive Gaussian noise. The set C may be considered *constraints* upon the unknown states X through the respective sensor models and predictive motion models. The models are shown in Equations 2.1 and 2.2 with corrupting additive Gaussian noise represented by the variables v and w. The probability density functions are shown in Equations 2.3 and 2.4, where Q and R represent covariance matrices. The predictive motion model in Equation 2.1 provides a constraint between two sequential pose states x_{k-1} and x_k given a control input u_k and covariance Q. The measurement model in Equation 2.2 provides a constraint between a pose state x_k and a landmark state m_j through the observation $z_{j,k}$ and covariance R. The set C would contain the control inputs u_k and measurements $z_{j,k}$ as the necessary parameters to generate constraints on states in X. The actual constraints consist of the differences between model predictions current state given the input parameters: $x_k - \hat{x}_k$ and $z_{j,k} - \hat{z}_{j,k}$. Constraints also include prior estimates (often called priors). One example of a prior is an initial condition on the initial vehicle pose or landmark locations from a prior map.

$$\begin{aligned}
\hat{x}_k &= f(x_{k-1}, u_k) + v && (2.1)\\
\hat{z}_{j,k} &= h(x_k, m_j) + w && (2.2)\\
P(\hat{x}_k) &\sim \mathcal{N}(f(x_{k-1}, u_k), Q) && (2.3)\\
P(\hat{z}_{j,k}) &\sim \mathcal{N}(h(x_k, m_j), R) && (2.4)
\end{aligned}$$

The joint probability distribution for estimated states is given in Equation 2.5. Since the set C consists of known inputs rather than unknowns requiring estimation, the joint distribution is equal to the conditional distribution. Using the Markov assumption in the predictive vehicle model and the independence of measurements, the distribution may be written as a product of the individual probability density functions of the measurements and predicted vehicle motions. Let the number of poses be N, where each pose is represented as x_k for $k = 0 \ldots N$. Let the number of observations be M. Each observation is represented as $z_{j,k}$ with subscripts denoting the association of the measurement to an estimated map landmark j and observed from vehicle pose k.

$$\begin{aligned} P(X, C) &= P(X|C) \\ &= P(x_0) \prod_{k=1}^{N} P(x_k | x_{k-1}, u_0) \prod_{j=1}^{M} P(z_{j,k} | x_k, m_j) \end{aligned} \tag{2.5}$$

We wish to compute the estimated states that maximize the joint probability distribution given in Equation 2.5. In other words, we wish to find the maximum a posteriori (MAP) estimate of the states X (trajectory, landmarks, etc) given the uncertain constraints in C as shown in Equation 2.6. Given the Gaussian nature of the distributions comprising the terms within Equation 2.5, one may apply a monotonic log-likelihood transformation as shown in Equation 2.8. The transformation reduces the product of exponentials to a summation of inner products, and constant factors with no effect to optimization are removed. The result is a weighted non-linear least squares objective function shown in Equation 2.9, which is a well-known form for optimization with many solution methods available [43]. Here we use E to represent the value of the objective function (error) we wish to minimize, and clarify the source of error for the two summation terms.

$$\begin{aligned} X^* &= \arg\max_X P\left(X, C\right) \\ &= \arg\max_X P\left(X|C\right) \qquad (2.6) \\ X^* &= \arg\max_X P(x_0) \prod_{k=1}^{N-1} P(x_k|x_{k-1}, u_k) \prod_{j=1}^{M} P(z_{j,k}|x_k, m_j) \qquad (2.7) \end{aligned}$$

$$\begin{aligned} X^* &= \arg\min_X - \log P(X, C) \\ &= \arg\min_X - \log P(x_0) - \sum_{k=1}^{N-1} \log P(x_k|x_{k-1}, u_k) - \sum_{k=1}^{M} \log P(z_{j,k}|x_k, m_j) \\ &= \begin{array}{ll} \arg\min_X & \|x_0 - \bar{x}_0\|^2_{Q_0} + \ldots \\ & \sum_{k=1}^{N-1} \|x_k - f(x_k|x_{k-1}, u_k)\|^2_Q + \sum_{j=1}^{M} \|z_{j,k} - h(x_k, m_j)\|^2_R \end{array} \qquad (2.8) \end{aligned}$$

$$E = \overbrace{\|x_0 - \bar{x}_0\|^2_{Q_0}}^{prior} + \overbrace{\sum_{k=1}^{N-1} \|x_k - f(x_k|x_{k-1}, u_k)\|^2_Q}^{\text{predictive error}} + \overbrace{\sum_{j=1}^{M} \|z_{j,k} - h(x_k, m_j)\|^2_R}^{\text{measurement error}} \qquad (2.9)$$

In the preceding equations we have introduced the subscript notation for covariance weighted inner products, also known as Mahalanobis distances.

$$\|y\|^2_A \equiv y^T A^{-1} y \qquad (2.10)$$

A variety of techniques exist for computing the solution to the probabilistic formulation given in Equation 2.9. Early work formulated the problem graphically as Markov Random Fields [33, 82], and inference methods were developed to solve MRF problems. Later work in the SLAM research provided insights into the problem structure. Specifically, using the naturally sparse structure for information matrix [158, 159] for the *full*-SLAM optimization [43] and the well known matrix-graph duality, solutions using graphical methods emerged in the last decade. Examples include SEIF [153], Graphical SLAM [63], GraphSLAM [153], square root Smoothing and Mapping (SAM) [43] and variants [79,114], Pose Graph Optimization (PGO) [119], Sparse Bundle Adjustment (SBA) [85], etc. The smoothing approaches, such as GraphSLAM, SBA, PGO, and SAM, have the advantage of exactness, when using non-linear models, over the earlier filtering approaches, such as SEIF. Smoothing methods iteratively update linearizations of the non-linear models, and filtering approaches linearize once. The more recent graphical smoothing methods incorporate optimizations for speed, efficiency, and scalability. Square root smoothing and mapping (SAM) incorporates fast matrix column reordering to quickly solve the linear equations [43]. Kaess et al. extends SAM to allow incremental incorporation of new constraints in iSAM using Givens rotations to maintain near optimal column ordering [79]. Other approaches use gradient descent methods rather than matrix reordering for efficient optimization of the linear equations [44,85,119].

For clarity, consider a small estimation problem for a two-dimensional vehicle where there are three vehicle states and a single map landmark. In this example, our estimated state is $X = \{x_0, x_1, x_2, m\}$, the constraints consist of a prior on the initial pose and two control inputs for state propagation and three landmark observations to yield $C = \{p_0, u_0, u_1, z_0, z_1, z_2\}$. Graphically, the Bayes Net for this example is shown in Figure 2-1. An alternative graphical form of this example, called a factor graph, is provided in Figure 2-2. Factor graphs are undirected bipartite graphs, where links connect the variable nodes with factor nodes [38]. In Figure 2-2, the square nodes represent the factors and round nodes represent the estimated state variables. The factor graph representation has a clear connection to the mathematical problem formulation, where each factor node has an associated probability distribution factor in the total probability distribution. Mathematically, the factor nodes correspond to the terms in Equation 2.7.

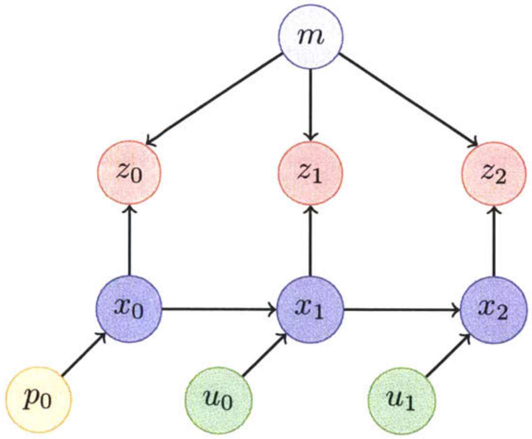


Figure 2-1: A graphical model of a small localization and mapping problem. There are three vehicle states, each of which make an observation of a map. The initial robot pose has a prior to establish the coordinate origin.

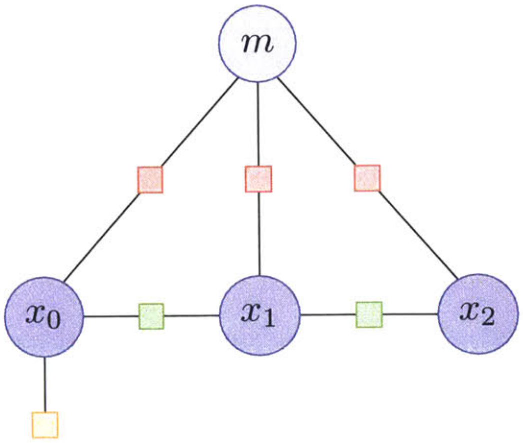


Figure 2-2: A factor graph representation of a mapping and localization problem also shown in Figure 2-1. Circles represent state nodes, and small squares represent factors. Relative measurements, such as vehicle prediction or landmark observation, result in binary factors connecting two nodes. The node colors are consistent with Figure 2-1.

2.2 Related Mapping Research

Given the generic 3D mapping problem formulation given in the previous section, this section describes research with robotic mapping with similarities to the work in this thesis. While the general formulation defines the problem, research results vary significantly in implementation. The research objective, platform, sensing capabilities, and environment present significant variability in research results.

First we discuss a selection of topological mapping methods to highlight the tradeoffs involved in applications placing greater importance on navigation than on global map consistency. Sibley et al. [140] convey several important considerations for designing and evaluating a SLAM system. Their work involved navigation outdoors in large-scale environments using primarily visual sensing. Aside from the technical achievements, the authors make several statements related to their design choices worth discussion. First the authors make the distinction between SLAM systems for surveying tasks and navigation tasks. A primary goal of surveying is generating a consistent map in a global coordinate frame. Navigation tasks do not necessarily require global consistency, as local consistency is sufficient for local navigation, and topological links may convey relations between local areas without requiring global consistency. The authors emphasize that their maps are constructed on Riemannian manifolds rather than a global Euclidean coordinate frame (note that Riemannian manifolds may be considered locally Euclidean). The authors' statement has relevance in other SLAM implementations. Namely, global map consistency is not always a necessity to the application. Conversely, surveying applications place high importance on global map consistency.

The ATLAS framework [27] employed a submap-based SLAM system with forced independence between submaps. A topological graph of submaps maintains edges between the submap nodes. Each local map could be considered locally Euclidean, and one must choose a root submap node to embed the graph in a global Euclidean frame. This is similar to the locally consistent mapping in [140], and the features and tradeoffs of ATLAS provide further support to the navigation versus surveying question. The local submaps of ATLAS indeed provide locally consistent maps, which is crucial to support one of its key features: traversal with multiple competing hypotheses. Each independent submap has an origin, whose uncertainty is captured in the uncertain edges of the topological graph. A robot may be well localized within a map while being highly uncertain of global position, so local navigation capabilities do not suffer from global uncertainties. Failed loop closures in ATLAS result in unnecessary new map generation, which has the advantage of robust real-time operation. Instead of experiencing navigation failure, the robot starts a new map and may continue operating.

2.2.1 Terrestrial 3D Mapping

There is an extensive literature on terrestrial 3D mapping, even within the domain of robotics or small perception vehicles. Two popular modes of perception for 3D mapping are laser ranging and computer vision. Laser ranging, or lidar, has the advantages of precise ranging and bearing to surfaces relative to the sensor. Monocular vision does not provide precise range sensing, but provides bearing and color for the environment visible within the sensor viewing frustum. Laser and vision provide information in different forms, and the literature reflects different strategies employed for 3D mapping for those two modes of perception.

3D Laser Mapping

Research in recent years has increasingly incorporated 3D laser mapping results. The progression towards three dimensions was a logical extension to the 2D laser mapping seen extensively in the early 2000's and a necessity for many outdoor robotic applications. A key distinction in prior research with 3D laser mapping is the technique employed for extending observability from planar range scanners to 3D range scans. The techniques were: (1) fixed lasers - typically mounted to scan perpendicular to platform motion (sidescan), (2) actuated lasers - planar scanners with an actuation mechanism to either rotate or "nod" the sensor, and (3) 3D scanners. Full 3D scanners are a relatively new and expensive technology for robotic applications with scarce literature to merit discussion in depth. A noted development for 3D laser scanners was the Velodyne scanner, which debuted on an autonomous car in the DARPA Urban Challenge in 2007 [93]. More recent results with specific application to mapping in outdoor urban environments are shown in [104].

One approach to mapping in 3D with laser scanners is mounting the sensor vertically, or orthogonal to the robot's motion plane, and exploiting the motion of the robot to observe the environment. Example implementations include [152] [75] [102]. One example of commercial use may be seen in the platforms for Google Street View project shown in Figure 1-4.

Research in 3D mapping with actuated planar range scanners first began with servo actuation to tilt or "nod" the sensor up-and-down for 3D observability. Examples of such work are Thrun et al. [155], Harrison and Newman [72], Pfaff [124] Cole and Newman [37], Surmann [150] and Nüchter [115]. The nodding scanner approach is limited in observability due to tilt speed and limits. As described in [37] and [115], tilting lasers may be used to simulate 3D scanners with a *stop-scan-move* collection strategy. The collection strategy decouples the robot motion from the sensor actuation, which simplifies registration to sequences of full 3D scans. An alternative yet similar approach to nodding is shown in [134], where Ryde et al. employed a rotating mirror in front of a scanner to provide a nearly hemispherical perception field. The continuously rotating mirror may be advantageous to tilting scanners by having lower power requirements, constant velocity control for actuation, and accurate angular encoding. Continuous rotating scanners provide full hemispherical observability with smooth continuous motion and do not suffer practical issues such as limited tilt angles or motor backlash at tilt limits. Bosse and Zlot in [25] presented a system employing a continuously rotating laser on a Bobcat vehicle for outdoor SLAM.

A commonality among the different 3D mapping applications is the use of registration methods to align the 2D or 3D scans with a registration method. Research in vision developed many of the three-dimensional point registration algorithms, such as least squares [5] and Iterative Closest Point (ICP) [22, 165]. Within the robotics community, sensor technology arguably influenced the point registration methods in robotics research. Early work by Lu and Milios [96] on range scan alignment from planar laser sensors became widely adopted as "scan-matching" [13, 70, 118] for 2D robot applications. Later as researchers constructed actuated planar scanners for three dimensional mapping, three dimensional registration methods were required. The dominant registration method for 3D point data has been ICP or a close variation of ICP. Details and further discussion of 3D registration are provided in Section 4.3.2.

Visual Reconstruction

Similar to global optimization in SLAM frameworks are Bundle Adjustment (BA) and Structure from Motion (SFM) methods in computer vision research. Several noteworthy recent publications demonstrate 3D reconstructions from images using SFM methods. Snavely's dissertation topic concerned 3D reconstruction from collections of internet photos [144]. He also presented graphical insights into the algorithmic approach [145]. 3D reconstructions of Rome from digital photo collections was presented in later work by Agarwal et al. [1]. Ni demonstrated three dimensional mapping of St. Peter's Basilica from images using bundle adjustment [113]. Although using images as inputs for the mapping, the mapping algorithm employed is similar to the approach in this work, where the author uses a smoothing and mapping (SAM) method and submaps. Ni details his mapping framework, dubbed Tectonic-SAM (T-SAM), in [114]. T-SAM outlines a hierarchical state-space for defining local maps relative to a map base node or anchor node. The relative representation isolates many constraints to affect only variables relative to the base node and relatively few constraints between variables defined relative to different base nodes. T-SAM takes advantage of the locality to optimize submaps independently. T-SAM introduces sparsity through re-parameterization of a given problem without ignoring information, so it maintains exactness rather than forcing sparsity through approximation. The primary difference between Ni's work and prior submapping SLAM approaches is the method integrates the exactness of global SAM with a divide-and-conquer method to bound in-core memory requirements, thereby allowing problems to scale beyond available memory limitations. In contrast, submap implementations such as Atlas [27] or SegSLAM [57] assume independence of submaps to maintain scalability.

2.2.2 Marine Mapping

Prior research for mapping in marine environments focuses primarily on exclusively mapping underwater [60, 126, 127, 130]. Simultaneous construction of surface maps is usually not an objective considered when using AUVs and ROVs operating in water far from land. Robotic surface crafts have much less history of publicized research, especially with respect to mapping. With recent emergence of robotic surface vehicles as research platforms, the research applications tend to focus upon autonomy, behavior, and navigational or communication aids to underwater vehicles [17, 41]. Historical applications of underwater mapping required deep submergence of vehicles for surveys to overcome limited perceptual range. Accuracy of underwater maps is highly contingent upon vehicle localization accuracy. With GPS unavailable, vehicle localization usually depends upon inertial sensors, acoustic beacons, and local features. Using expensive but highly accurate inertial sensors improves localization dramatically, but error accumulates with distance traveled [83]. Acoustic beacons such as those used in Long Base-Line (LBL) [161] and Short Base-Line (SBL) [143] provide artificial landmarks, or fiducials, at known locations to allow the robot to compute its position relative to those features [92, 121]. This thesis presents an alternative approach by simultaneously mapping both above and below the surface to improve the accuracy of the underwater map. Assuming fixed surface structures, map accuracy can exceed GPS accuracy. Variations of GPS biases over time become apparent when comparing multi-session data.

Fairfield, Wettergreen, Kantor, and Stone demonstrated underwater sonar mapping with octree evidence grids in a series of publications [58, 59, 66, 148]. They mapped underwater tunnels and sinkholes with a submersible robot equipped with sonar sensors. Their work is unique in objective and application. They map the walls of a sinkhole rather than create typical ocean bathymetry maps or perform bottom target detection. We adopt a similar strategy in underwater mapping with evidence grids and efficient octree data structures.

Several demonstrations of successful vision-based underwater reconstruction have been published in recent years [11, 54, 77]. In work by Eustice et al. [54], cameras on a Remotely Operated Vehicle (ROV) gathered images. The authors' final result was a reconstruction of significant portions of the sunken *Titanic* vessel. Their work employed features extracted from images for landmarks in their navigation system. They used the Exactly Sparse Extended Information Filter (ESEIF) as the algorithmic approach for trajectory and map estimation. The ESEIF algorithm is one of the earlier frameworks for the full SLAM problem with clear graphical and sparsity motivations.

In [11], Beall et al. demonstrates high quality 3D underwater reconstruction using stereo vision camera. Although the experimental data was of limited scale (approximately 7 meters), their work incorporates recent advances in sparse full trajectory smoothing and mapping using Dellaert's SAM estimation framework. Additionally, they produce a mesh reconstruction of an underwater surface from the optimized map points with Delaunay-based triangulation. They achieve impressive visualizations of their results with texture mapping of camera frames onto the reconstructed mesh surface. Unlike other related work in underwater mapping, they constructed maps using solely imagery without any other navigational sensors. They focused on applying visual odometry methods to a new environment. The experiments did not involve an actual robotic platform, but rather a "rig" to aid a potential AUV or propelled by a diver or strapped to a boat as in their experiments.

Research by Roman et al. in [129,131] involved creation of consistent bathymetric maps, where the authors essentially used submaps generated from multibeam sonar over short periods of time and registered the submaps against each other. Roman's work is similar to this work in concept, although the work presented in this thesis involves registration of surface maps to improve underwater bathymetry maps. Their work involved bathymetry mapping in the deep ocean and could not benefit from simultaneous surface sensing precision to improve their maps. The sonar sensor employed in their work provided a large sensing field of view resulting in high coverage of the operating area for their predefined survey route. The high coverage permitted submap registrations by providing adequate point cloud density, and the high sensor coverage is unlike the applications such as ours and others [11,53,77].

Johnson-Roberson et al. [77] employed an underwater vehicle for lengthy surveys, and generated a large scale reconstruction from stereo imagery of the environment covered. Their work is similar in application to Beall et al.'s work in terms of underwater 3D mapping with vision, but differs in terms of physical scale with [77] by covering regions extending kilometers rather than meters. The sensing footprint of their vehicle is small in comparison to the scale of their operating environment, which results in sparse coverage within the bounding area of vehicle trajectories.

2.3 GPS-aided Mapping

For our target mission, GPS is typically available in open-water locations with a clear view of the horizon. In the mobile robotic community, mapping is typically posed without recourse to GPS. Using GPS might seem to make the problem "too easy" but in practice, applying GPS constraints properly with a SLAM framework is fraught with difficulty. For marine mapping application, view of the horizon is often poor, and GPS data is often corrupted near marine structures.

Of previous work in robotics that has considered GPS errors, the work of Carlson stands out [30]. Carlson demonstrates GPS-based Smoothing and Mapping (SAM) in a large scale urban environment using planar laser scanners on an automobile and uses corner features as landmarks in constructing a two dimensional map. Carlson provides details concerning how GPS-based SAM differs from the previously published use cases, which predominantly utilize relative constraints between nodes in an arbitrary coordinate system. Carlson's work included an extensive analysis of GPS error, and the estimation state space explicitly included states related to GPS biases, such as receiver offset and satellite bias.

Our work also applies Smoothing and Mapping (SAM) in a global GPS coordinate system, and has a few key differences. In contrast to Carlson's work, we do not employ a landmark map representation, since reducing the environment to the data matching templates of a small alphabet of landmarks prevents reconstruction by ignoring data not recognized as a landmark. In our work, raw pseudoranges are unavailable from our sensor to permit applying the model presented by Carlson. Additionally, lack of a reference station precludes integration of differential constraints.

2.4 Open Issues

The 3D mapping research described in the previous section represents a large and diverse body of work that addresses some of the issues of our target application, but no previous research has encompassed the scope of this problem.

A key challenge for robotic map creation is achieving consistency and accuracy for large scale maps over multiple sessions. Recent research has made progress in this domain [86,87,111]. In contrast to the previously mentioned methods, our work attempts to generate multi-session maps in a previously established global coordinate frame. The authors in [87] specifically mention absolute global error may be large when compared to ground truth, and such a statement would be true for systems lacking observability of an external reference frame. For surveying applications, creation of a map in an external frame known to the user is a primary goal. Examples of external reference frames for outdoor applications are coordinate systems derived from the Global Positioning System, which use standardized datums such as NAD83, WGS84, or EGM2008.

As precision robotic mapping applications scale to larger environments, manifold approximation error becomes non-negligible. Manifold approximation error arises from Euclidean assumptions of local topology, while the actual planetary surface is not flat. The GPS coordinate datums mentioned previously provide an elliptical approximation to Earth to account for the curved topology. Due to the difficulty of working in elliptical reference frames and the infinitesimal error at small scales, linearizations to local Euclidean coordinates have been sufficient for most robotic applications. For very large maps, the issue of maintaining map consistency at both local and global scales remains open. Promising work on adaptive learning to account for manifold curvature [166] could supplement SLAM estimation frameworks.

2.5 Summary

This chapter has posed the 3D robotic mapping problem with a general probabilistic formulation for map and trajectory estimation and discussion of algorithmic methods for solving the estimation problem. This chapter discusses selected research with relevance to robotic 3D map estimation, both in the terrestrial and underwater environments. Our work spans both the terrestrial and underwater domains of marine environments and addresses the challenges of both domains. In the following chapters, we describe the representations and algorithmic approach we developed to address the total marine 3D mapping problem.

Chapter 3

Perception-driven GPS Trust Engine

This chapter introduces a framework for negotiating the difficulties of using Global Positioning System (GPS) sensors in cluttered environments. We start with an overview of the GPS system, clarifying the principles behind GPS-aided navigation and the hazards to using GPS. We present a classifier framework to determine when GPS failure is likely. Our framework utilizes a feature set of the platform sensors in determining whether estimation may place high or low trust in the GPS sensor. The classifier framework presented in this chapter is a key component of our work in providing good constraints in the optimization presented in Chapter 4.

3.1 GPS Aided Navigation

The Global Positioning System (GPS) is an extremely useful resource for navigation information in a variety of outdoor robotic applications. GPS may be used for navigation, mapping, surveying, and time synchronization [41, 55, 56].

The current global navigation satellite system (GNSS) used for GPS consists of the NAVSTAR system of the United States and the *Glonass* system of Russia, although other systems are under development. The NAVSTAR system, originally 24 satellites, currently consists of 31 satellites. The basic functionality is similar to landmark navigation with known data association. Each satellite transmits a unique identifier, its pseudo random number (PRN), and orbital parameters allowing localization of the satellite by a receiver. Receivers attempt to compute positions from the satellite signals using psuedo ranges, or ranges based upon a signal's time of flight and the speed of light. Clock synchronization is necessary to use one-way ranging for time of flight signals, so the clock offset between the satellite and receiver must be estimated. The basic pseudo range model for a receiver at $\mathbf{x}_r$, a satellite with PRN of k at position $\mathbf{x}_{s[k]}$, speed of light c, clock offset δt, and error ν is shown in Equation 3.1.

$$\rho_k = ||\mathbf{x}_r - \mathbf{x}_{s[k]}|| + c\delta t + \nu \tag{3.1}$$

Position estimation using the model given in Equation 3.1 requires solving for the four unknown position and clock parameters, so at least four satellites are required [39]. Non-terrestrial applications might have to account for baseline ambiguity, but here we assume the solution is the one between the earth and satellites and not on the far side of the satellites. More pseudo ranges from satellites, ideally dispersed throughout the sky relative to the receiver, may improve accuracy. GPS receivers may also provide velocity by observing the Doppler shift of the satellite signals relative to known carrier frequency. The shift is modeled as the inner product of the relative velocity and relative direction of the satellite from the receiver corrupted with noise, as shown in Equation 3.2. Let $d_{s[k]}$ represent the Doppler shift and w represent a corrupting noise input.

$$d_{s[k]} = \left(\dot{\mathbf{x}}_{s[k]} - \dot{\mathbf{x}}_r\right) \cdot \frac{\mathbf{x}_{s[k]} - \mathbf{x}_r}{|\mathbf{x}_{s[k]} - \mathbf{x}_r|} + w \qquad (3.2)$$

Characterizing GPS error is challenging, and there are many sources of error. Error may be introduced at several stages within the system. Each satellite broadcasts its ephemeris (orbital parameter values from a predictive model), which introduces some modeling error at the signal source. Satellite clock offsets also introduce small but not insignificant error at the source. Errors introduced at the satellite source may be reduced offline using more accurate ephemeris and clock data [108], but such errors remain during online processing. Errors are also introduced during signal transmission due to atmospheric effects. Refraction in the ionosphere and passage through the troposphere introduce delays in the signal, especially when a satellite is low on the horizon relative to the receiver, where the signal path includes more of the atmosphere.

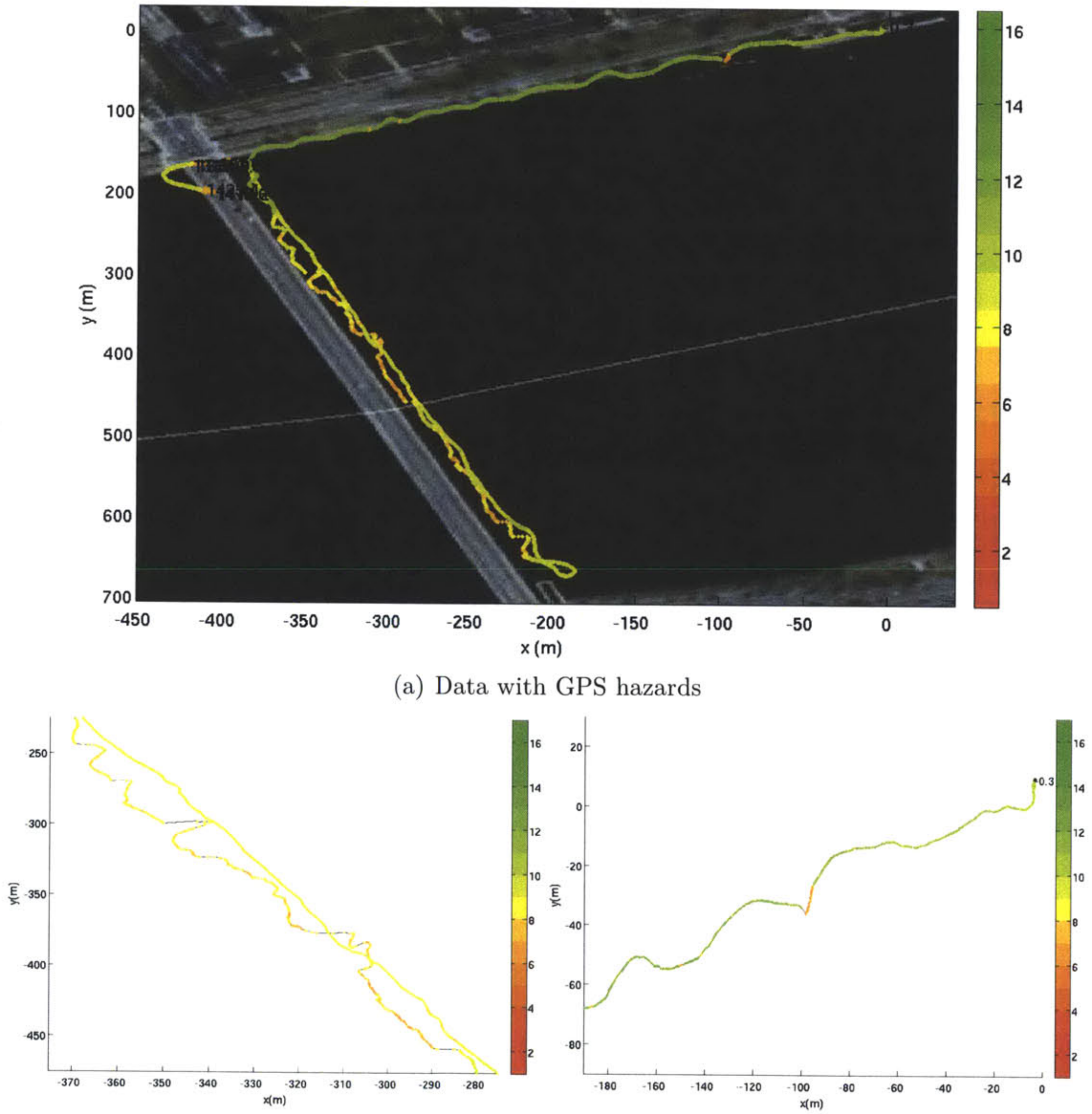
Multipath error is one of the most troubling error sources. Multipath errors result when a signal has more than one path from satellite to receiver, usually from a reflection off a surface such as a building. Receivers often detect and disregard multipath when a signal arrives at two different times, but when multipath is not rejected, significant errors are introduced. Corridor or canyon environments such as tall buildings in a city pose multipath hazards, where a signal can have several paths to the receiver and possibly no line-of-sight path. In combination with a small number of observed satellites, a receiver may not provide a robust solution. Dual-antenna systems are highly susceptible to multipath errors since one or both antennas may receive a multipath signal and the error corrupts jointly estimated parameters. The effect of multipath errors in positioning can lead to results with abrupt jumps in an otherwise smooth vehicle trajectory and temporary biases and drifts in position. Differentiating multipath from other errors is a challenge especially under simultaneous constellation changes. Example scenarios include the canopy and corridor effects of entering or exiting a covered bridge having some corrupted satellite signals available near the entrance or exit.

Multipath is particularly hazardous for velocity derived from Doppler shift observations. The frequency of composite echo and direct signals will have different frequency, and the observed shift is different. When only echoed signals are observed, the direction of the shift can also have the wrong sign.

The satellite constellation observed by a receiver and its geometry also affect accuracy. Solving for a three-dimensional position and receiver clock offset requires a minimum of four satellite observations. A larger constellation may provide more measurements, which can yield a more robust position solution. In the ideal case, the satellite constellation would be distributed throughout the sky. If the observed satellites are clustered in one portion of the sky, the poor geometry diminishes accuracy.

Details of GPS usage in robotics varies widely and is often application specific. Many commercial receivers have internal filters and provide some information about the implementation within manuals. Most receivers provide output in binary or National Marine Electronics Association (NMEA) format [6] for basic standardized information, but not all provide pseudo range values, satellite PRN, or intermediate parameters that could permit optimization. While GPS error is not strictly a time-invariant Gaussian noise process, many applications model it as such without severe consequences. Devices integrating other information from accelerometers, rate gyros, magnetometers, or encoders often report raw GPS and other sensor data in addition to their computed navigation solution.

Figure 3-1 illustrates several GPS hazards encountered experimentally. The figure shows GPS position measurements for a vehicle travelling near a bridge. In the upper left region of Figure 3-1(a) the vehicle went under the bridge, and the canopy condition resulted in too few satellites visible to estimate position and a discontinuity in the trajectory. Not only did the GPS sensor lose satellites under structures as expected, but it was sufficient for the vehicle to be near a structure for positioning to have discontinuities. In the central portion of Figure 3-1(a) and in greater detail in Figure 3-1(b), effect of vehicle proximity to the bridge structure on GPS position is shown. The vehicle traveled southeast along the bridge and returned in the northwest direction at a closer distance to the bridge. Although no ground-truth trajectory is available, the actual vehicle trajectory is smooth in both directions, but GPS measurements only support a smooth trajectory when the vehicle traveled at a greater distance from the bridge. On the northwest route, the GPS trajectory shows several discontinuities and distortions. Multipath and poor visible satellite geometry are often the cause of the lateral "warping" seen in the figure both along the bridge and the retaining wall.



(a) Data with GPS hazards

(b) Distortion near bridge

(c) Multipath near wall

Figure 3-1: GPS multipath and signal loss during an experiment. Measurements are colored by the number of satellites reported by the GPS device, which is a crude indicator of position error. The top image shows the overall trajectory and the lower images show selected views. See text for discussion.

3.2 Discriminative Classifier

Within the estimation framework for our system, localization accuracy has great impact on the mapping quality. Without the use of sequentially observed landmarks, the burden of localization falls upon the position, motion, and orientation sensors. While characterizing the sensors for mapping in marine environments, two different modes of operation were observed: one when GPS quality is high, and the second when GPS quality degrades. The two modes correspond to different management of sensor measurements.

A common problem in estimation frameworks is creating accurate models with low complexity. Violation of assumptions is not always catastrophic, and there are many cases where both the estimation algorithm and application are robust to minor violations. For example, a time-varying bias in a GPS sensor likely would not cause an EKF to fail. For precision surveying applications, the insufficiency of the models can lead to poor results.

One alternative is to explicitly model parameters like satellite biases and receiver clock offsets using the pseudoranges from the satellites. The literature contains many resources on modelling at the pseudorange level [30,32,39]. We propose an alternative approach to improve localization by adaptively tuning parameters and changing models to current conditions. Methods such as Interacting Multiple Models [10, 99] are examples, but such approaches adapt only after recognizing increases in estimation error.

In our framework, two different vehicle modes are used to improve vehicle trajectory estimates. The primary mode applies to conditions of good GPS accuracy, implying good visibility of satellites and few obstructions to block signals or cause multipath. The secondary mode applies to conditions with poor satellite visibility or when signal multipath hazards are likely. Examples of poor conditions are while under a bridge or in close proximity to objects. Such conditions exhibit canopy and canyon effects in similar fashion to the urban canyon effect observed in cities between large buildings. The hazards due to multipath or changes in the constellation of tracked satellites can lead to observations violating the Gaussian error assumption in the measurement model to a degree where noticeable discontinuities or biases appear in the resulting path. Improvement is seen with a smoothing SLAM solution; however visual inspection of projected range data shows noticeable warping distortions. To mitigate the warping effects, a preemptive measure to discount the sensor in question is required. A classifier is designed and trained to detect when the vehicle is likely in a GPS-degraded situation, and change models for vehicle state and observation prediction.

The primary vehicle mode S_0 incorporates GPS position updates, IMU reported velocities, and DVL velocities when both are available and valid. The secondary mode S_1 increases GPS rejection, decreases DVL uncertainty (when available), and switches from updates with IMU velocities to integrating IMU accelerometer data within the prediction step. A particular difference between the models is that one uses the GPS/IMU derived velocities, and the other uses only the IMU accelerometer data. The IMU internally filters accelerometer data when reporting velocities. It also uses differential GPS from the two GPS receivers, and in poor conditions the reported velocities are corrupted. The primary mode does not incorporate accelerometer data within the predictive step to avoid overconfidence and double counting of information.

The secondary mode uses the accelerometer rates within the predictive step, and as a consequence is more susceptible to accumulated error over time but not the hazards of velocity measurements with a GPS-derived component. The secondary mode is intended for short duration, where error accumulation is limited. The primary mode places relatively high trust in GPS data and much lower trust in DVL data, and the secondary mode trusts the DVL with low trust in GPS position and rejection of GPS based velocity.

To minimize the effect of GPS hazards, we design a discriminative classifier to adapt parameters and models to current conditions. The purpose is to create a framework for inferring the existence of hazards to GPS-derived data from the available sensor data and adapting the estimation algorithm. There are several considerations in the design to address. First, the available sensor data has high dimension, so dimensionality reduction is desirable. Second, the algorithm must generalize to new data. We demonstrate the use of semi-supervised methods for learning a discriminative classifier which performs the mode selection, which is the adaptive tuning of parameters and models.

To determine which vehicle mode applies to a given time, a linear classifier is trained on a subset of the data with manually labeled classes according to times the vehicle was underneath a bridge determined using the camera sensor. The classifier model accepts an input vector $\mathbf{v}$ in a feature space $\mathcal{V}$ and outputs the binary label as shown in Equation 3.7. The weight vector $\mathbf{w}$ must be determined empirically by training the classifier. First, the feature space must be chosen and defined. The space of data available to the robot from all sensors has high dimension, but may be reduced by functions mapping the inputs to a low dimensional feature space $\mathcal{V}$.

Simple linear regression takes the form shown in Equation 3.4. Simplifying to vector notation often requires pre-pending a constant to the input space to accommodate the offset or bias term w_0, so the vector $\mathbf{x}$ is treated as $[1 \;\; x_1 \ldots x_n]^T$.

$$f : \mathcal{R}^n \mapsto \mathcal{R} \tag{3.3}$$

$$\begin{aligned} f(\mathbf{x}; \mathbf{w}) &= w_0 + w_1 x_1 + \ldots w_n x_n \\ &= \mathbf{w}^T \mathbf{x} \end{aligned} \tag{3.4}$$

More generally, the input space may be transformed while maintaining a regression function that is linear in the weight vector $\mathbf{w}$ through the use of basis functions, or features, of the original input space of $\mathbf{x}$. For one-dimensional input data and basis functions $\phi_k(x) = x^k$, the result is polynomial regression. Considering higher dimensional inputs, basis functions may be a selector function where $\phi_k(\mathbf{x}) = x_k$, which would result in the simple linear regression shown in Equation 3.4.

$$\begin{aligned} \phi : \mathcal{R}^n &\mapsto \mathcal{R} && (3.5) \\ f(\mathbf{x};\mathbf{w}) &= w_0 + \sum_{k=1}^{n} w_k \phi_k(\mathbf{x}) && (3.6) \end{aligned}$$

Basis functions provide a convenient mechanism for applying simple and robust linear regression models to a large variety of problems. The trick becomes the selection of the basis functions. Good basis functions transform the input data into a space which a regressive discriminator finds separable. When using high-dimensional data, basis functions may compress the input into a small space of sufficient statistics for discrimination. For our application, the input space consists of available sensor data. Much of the sensor data is not informative for discriminating whether GPS conditions are poor, and a selection of basis functions can make the problem suitable for low-dimensional classifiers.

$$S(\mathbf{v}) = \begin{cases} S_0 & \mathbf{v}^T\mathbf{w} \leq 0 \\ S_1 & \mathbf{v}^T\mathbf{w} > 0 \end{cases} \qquad (3.7)$$

The input to the classifier for a given time are a vector containing features chosen as reasonable indicators of reliability of GPS signals. The features are as follows:

- The reported velocity error of the IMU/GPS.
- The number of reported GPS satellites tracked by the device.
- The reported dilution of precision (DOP) by the GPS device.
- The mean inverse range of the horizontal lidar for a field of view of 30 degrees on the right side (0 to 30 degrees). This figure is derived from the green arc on the left side of Figure 3-2.
- The mean inverse range of the horizontal lidar for a field of view of 30 degrees on the left side (150 to 180 degrees). This figure is derived from the green arc on the right side of Figure 3-2.
- The mean inverse range for the vertical lidars within 15 degrees of vertical. This figure is derived from the two arcs extending upward in Figure 3-2.

Figure 3-3 shows example features derived from real data. The robot made multiple traversals under bridges between approximate times 1000-3000 seconds and after 5000 seconds. The high correlation between the features and the robot in a GPS-degraded situation suggests the utility of the selected features as input to a classifier.

The reported velocity error of the IMU/GPS grows when the unit is in a GPS-degraded situation. Multipath effects are most troublesome for velocity observations, and such effects are probable upon approach of an obstruction. The dilution of precision (DOP) of position estimate by the IMU/GPS also grows in poor conditions from loss of visible satellites and multipath effects. While the device will report fewer visible satellites in GPS-degraded situations, it will typically hold a few satellites, and the delay between signal loss or device internal rejection and the device reporting such is undocumented. The features using GPS/IMU reported statistics are signals whose response lags the actual transitions between good and poor GPS conditions.

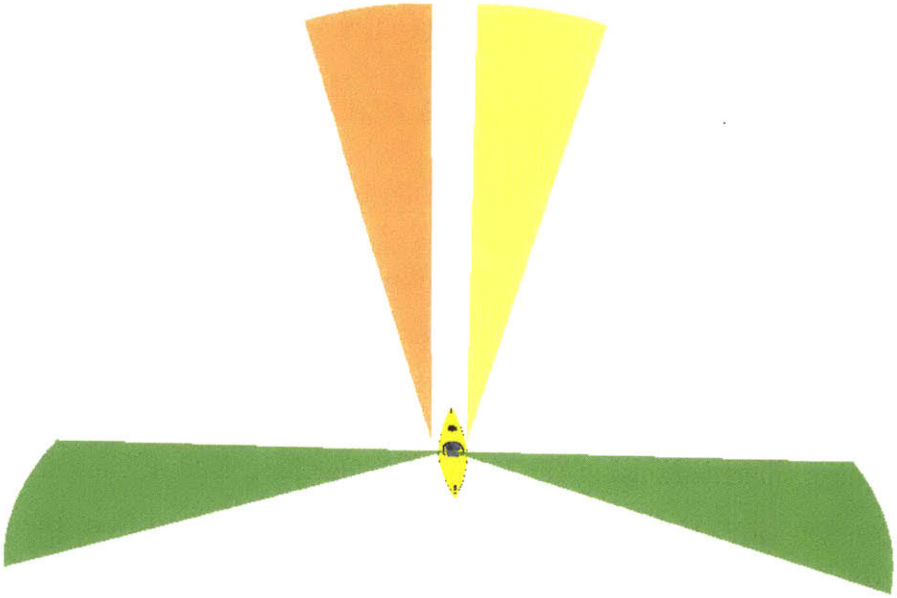
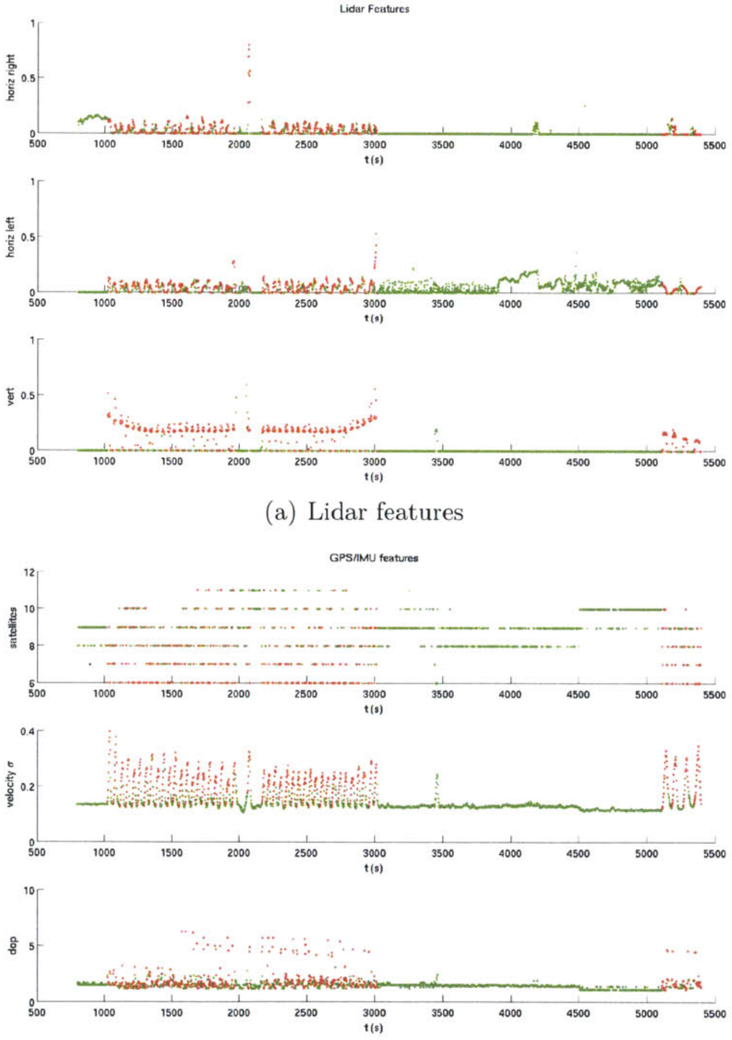


Figure 3-2: Regions within the lidar sensors' field of view used to generate features for the classifier are highlighted. Note the colors indicate the sensor source and not the feature. For instance the two horizontal green regions are separate features, and the vertical regions from separate sensors are grouped into a single feature.

Figure 3-6 shows a typical evolution of reported uncertainties by the IMU under transition between ideal and canopy GPS conditions. Due to internal filtering on the device, the reported uncertainty increases with a delay. We address this by supplementing with features derived from the lidar sensors designed to detect canopy and corridor situations. The relevant regions of the lidar sensors' field of view are illustrated in Figure 3-2. Figure 3-4 and Figure 3-5 show the features based upon the GPS/IMU or lidar sensors, respectively, for a shorter interval of the data also shown in Figure 3-3. In those two figures, the vehicle made repeated traversals under the Harvard Bridge resulting in canopy hazards. The features based upon horizontal lidar data were chosen as metrics for corridor conditions, and the vertical lidar feature is a metric for canopy conditions. For the environments targeted in this thesis, the chosen combination of features lead to a vehicle mode selector with substantial improvement over use of a single vehicle model.



(a) Lidar features

(b) GPS/IMU features

Figure 3-3: Classifier input features for a dataset. Red color indicates degraded GPS conditions. Figure 3-3(a) shows features derived from laser scanners. The upper two plots show features designed for detecting corridor conditions from the horizontal lidar, and the lower plot shows a feature designed to detecting canopy conditions. Figure 3-3(b) shows features derived from the GPS and IMU sensors: the number of satellites in use, velocity uncertainty, and dilution of precision (DOP).

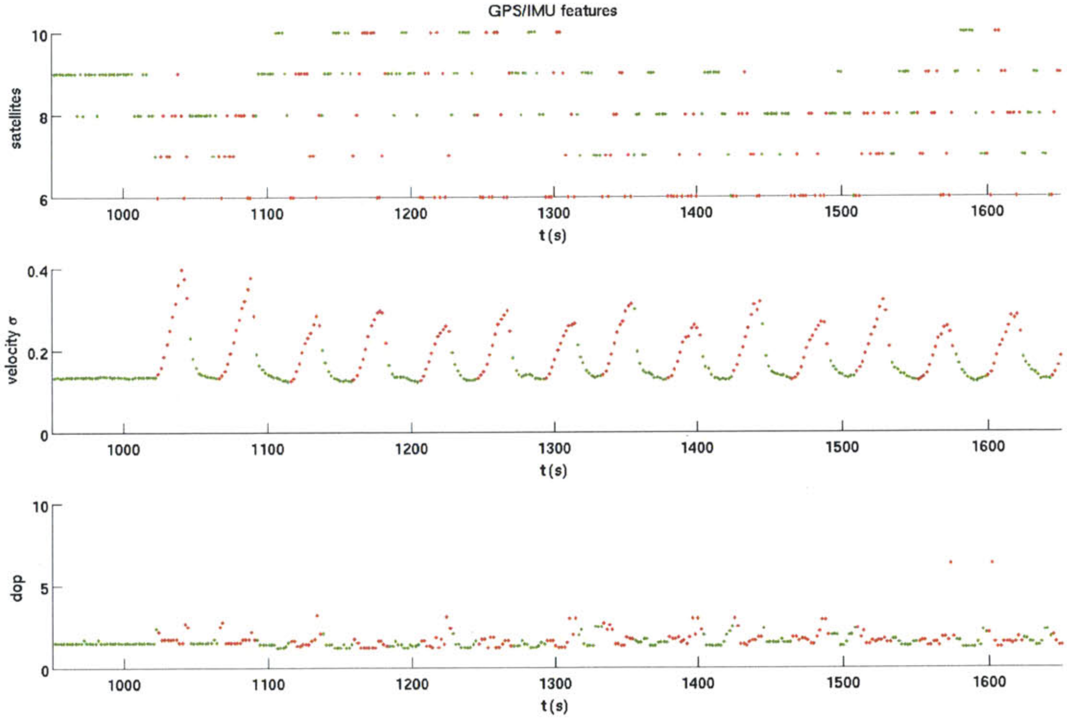


Figure 3-4: The classifier input features derived from the GPS and IMU sensors is shown for a selected time interval. Red color indicates poor conditions. The vehicle encountered repeated canopy conditions for the time sequence shown. The clearest indicator of the changing conditions is the reported velocity uncertainty (middle). The reported satellite count (top) and dilution of precision (bottom) are also features correlated with the changing conditions.

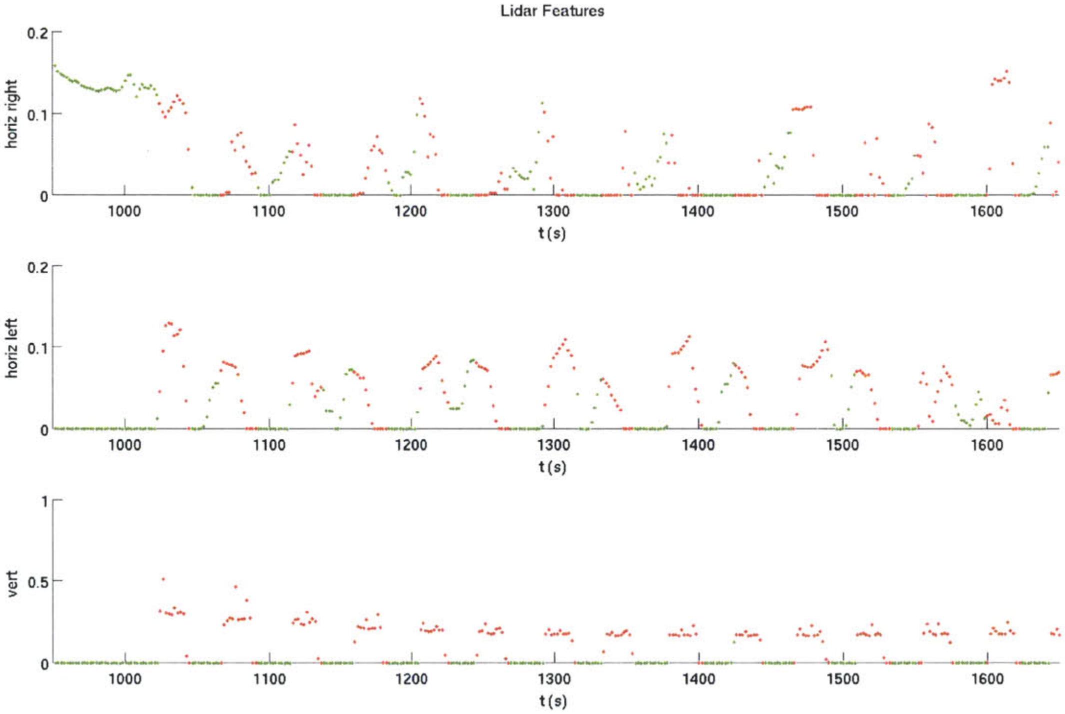


Figure 3-5: The classifier input features derived from laser sensors is shown for a selected time interval. Red color indicates poor conditions. The vehicle encountered repeated canopy conditions for the time sequence shown. The feature using the vertical lidar (bottom) was specifically designed for canopy conditions, and the features using the horizontal lidar (top and center) were designed as a metric for corridor conditions.

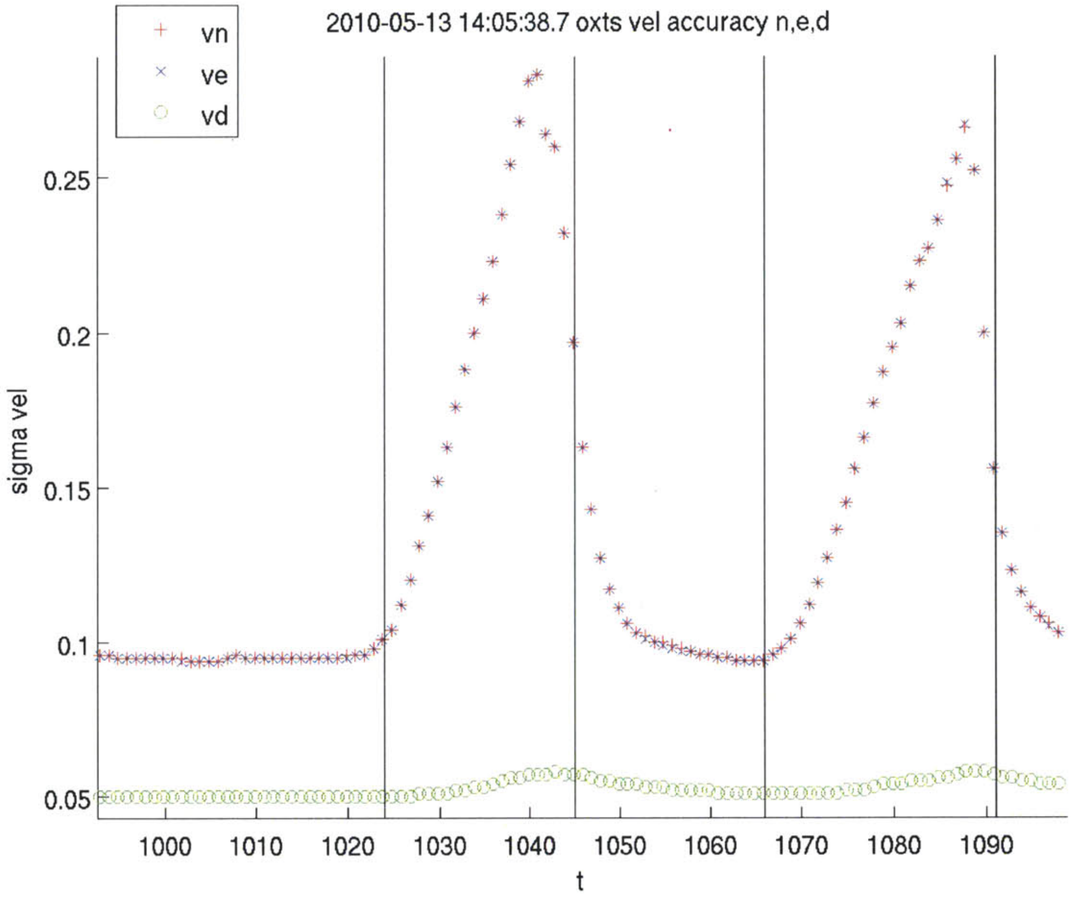


Figure 3-6: This figure shows the evolution of reported velocity uncertainty by the GPS/IMU sensor for a training data set. The robot begins in the ideal condition. Vertical black lines indicate transitions between ideal GPS conditions and canopy conditions. The velocity uncertainty (both blue and red) grows under poor conditions, but only after a delay.

3.3 Classifier Performance

Training with subsets of random samples from the labeled data resulted in a mean overall classification error of 4.6%. Figure 3-7 shows results of the trained classifier with errors shown in red. Note the labels were given manually for the training data, and some human error remains a contributing factor to the model's classification error. Errors in the manual labels would appear at transition points. The approximate percentage of time in GPS poor conditions in the labeled data set is 25%, amounting to 35 minutes. Most of the classification errors occurred at the transition points as shown in Figures 3-7 and 3-7(b). Empirically, labels are locally stable and do not exhibit hysteresis near transitions. In Figure 3-7, the apparently misclassified labels occurring near time 3400 seconds depict a situation where the vehicle briefly entered a GPS poor region that was not a traversal under a bridge and was not labeled as poor. The classifier in this case is arguably more correct than the prior labels.

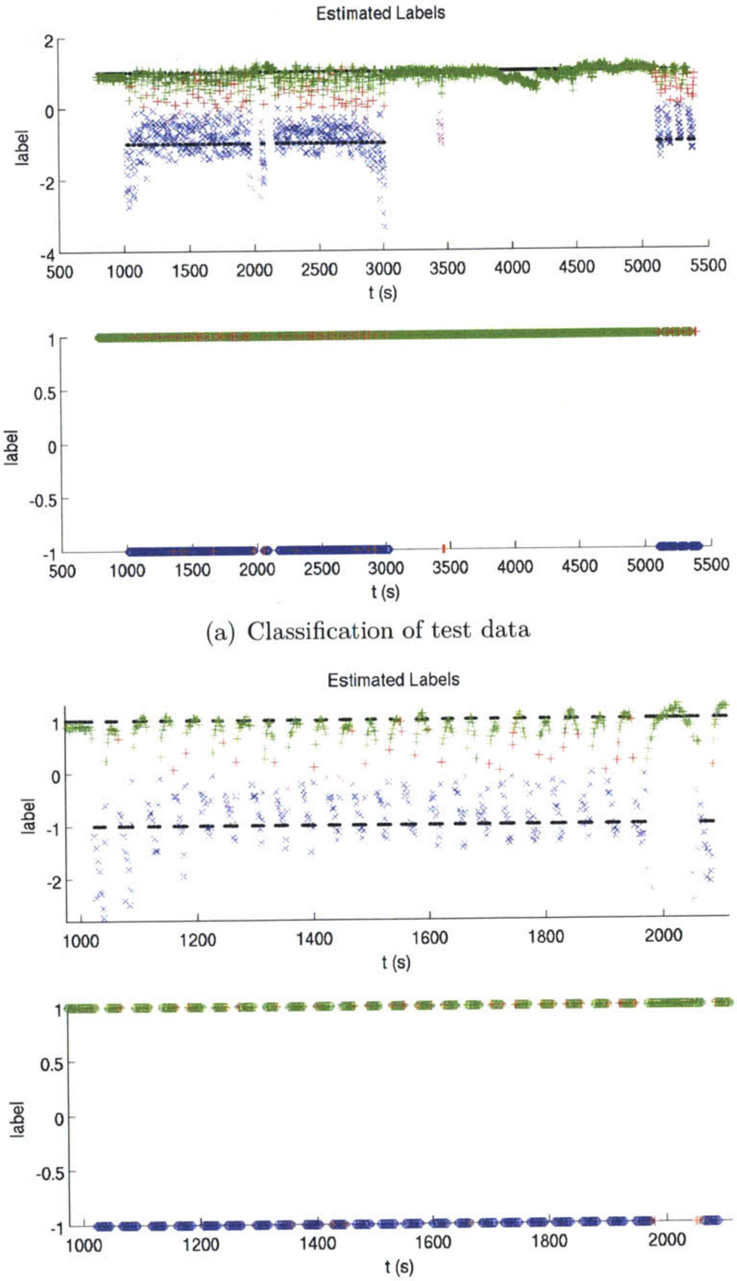
Of the chosen features in the classifier feature space, the vertical lidar feature and reported IMU velocity uncertainty feature were most significant and had the greatest correlation to label. The reported dilution of precision and satellite count were slightly less correlated, and the horizontal lidar features were the least correlated of the features, but not insignificant to the classifier performance. The importance of the reported velocity uncertainty was expected, as this statistic was chosen largely due to its growth in GPS poor conditions during preliminary experiments. The limitation of this statistic is the delay between encountering GPS poor conditions and the uncertainty growth, yet the feature remains useful for reinforcing other evidence. Figure 3-3(b) shows how uncertainty growth is slow when initially entering canopy conditions. The vertical lidar feature does not suffer the delay for canopy conditions.

Figure 3-8 shows how the classifier mode selection applies to the vehicle trajectory. As desired, the portions of the trajectory where the vehicle was under a bridge have a different mode than the trajectory portions where the vehicle had good GPS observability.

For comparison, a classifier was trained with only the number of satellites as its input. The resulting classification error averaged 14%, which is several times larger than the error from the classifier using the full feature set. While the satellite-only classifier was correct nearly 86% of the time, the classification errors had a distribution less practical than the full-feature classifier. The full-feature classifier exhibited errors clustered at transitions points, and the satellite-only classifier shows errors near transition points and during canopy conditions. Figure 3-10 shows the vehicle trajectory for a survey of the Harvard Bridge colored by mode for both the full classifier and the satellite only classifier. The ideal classifier would show red trajectories during GPS hazards such as the canopy conditions under the bridge. The full classifier shows stable red trajectories under the bridge while the satellite-only classifier incorrectly shows blue trajectories under the bridge.

Not only does the satellite-only classifier have greater error, the distribution of error during GPS poor conditions makes it less practical for our application. In Figure 3-10(a), the presence of the default mode for path segments under the bridge reveals temporary increases in reported number of satellites by the device. A possible explanation is the sensor is unable to reject multipath signals due to a reduced set of good signals for comparison and the multipath signals become too similar to good signals in close proximity to the structure. The temporary increases in reported satellites do not reflect the quality of signals. The classifier using only satellite counts chooses the wrong mode in those conditions.

Next we discuss the effects of the full-feature classifier relative to results without the classifier. Figure 3-9 shows how the vehicle trajectory compares when using a single mode and the dual mode in a particular situation where GPS/IMU unit provided poor velocity estimates. In this example, the reported uncertainties from the GPS/IMU device are overconfident, likely due to violating Gaussian error assumptions within the device's internal EKF-based filter. While comparing the vehicle trajectories shows relatively small differences, those small differences are magnified when projecting range data. Figure 3-9 highlights the effect on projected range data for a situation where the robot travels underneath a large bridge. The projected ranges are more consistent with the true bridge structure when localization is aided by the classifier. The classifier modes are critical to achieving map consistency in the proximity of GPS hazards.



(a) Classification of test data

(b) Classification of subset of test data

Figure 3-7: Classifier output for a dataset. Blue and green denote the two models, while red and magenta denote errors relative to the given label (shown in black). The second and fourth plots threshold the classifier output to one of the two labels for clarity with all errors marked red. The lower plots in Figure 3-7(b) magnify a portion of upper plots.

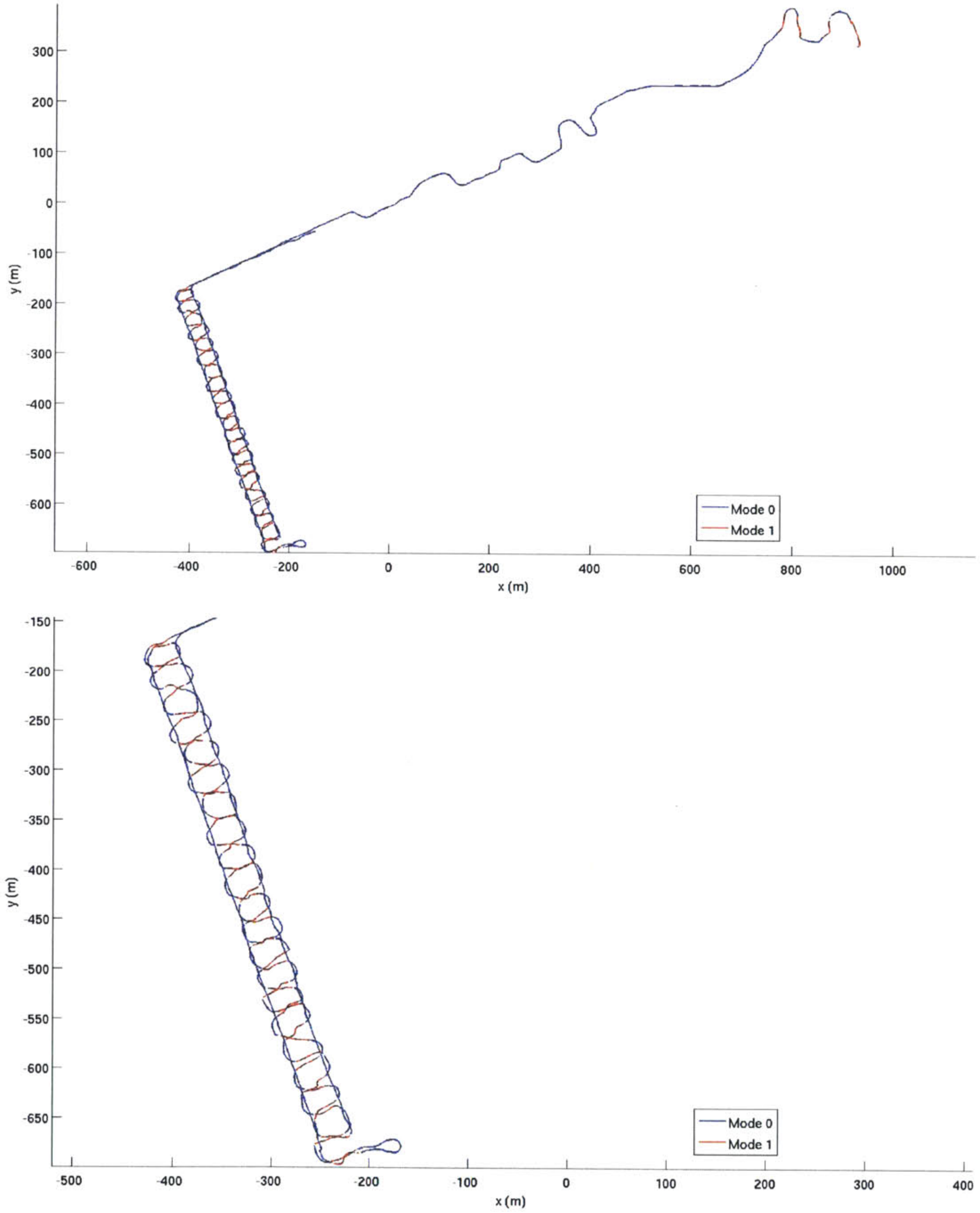
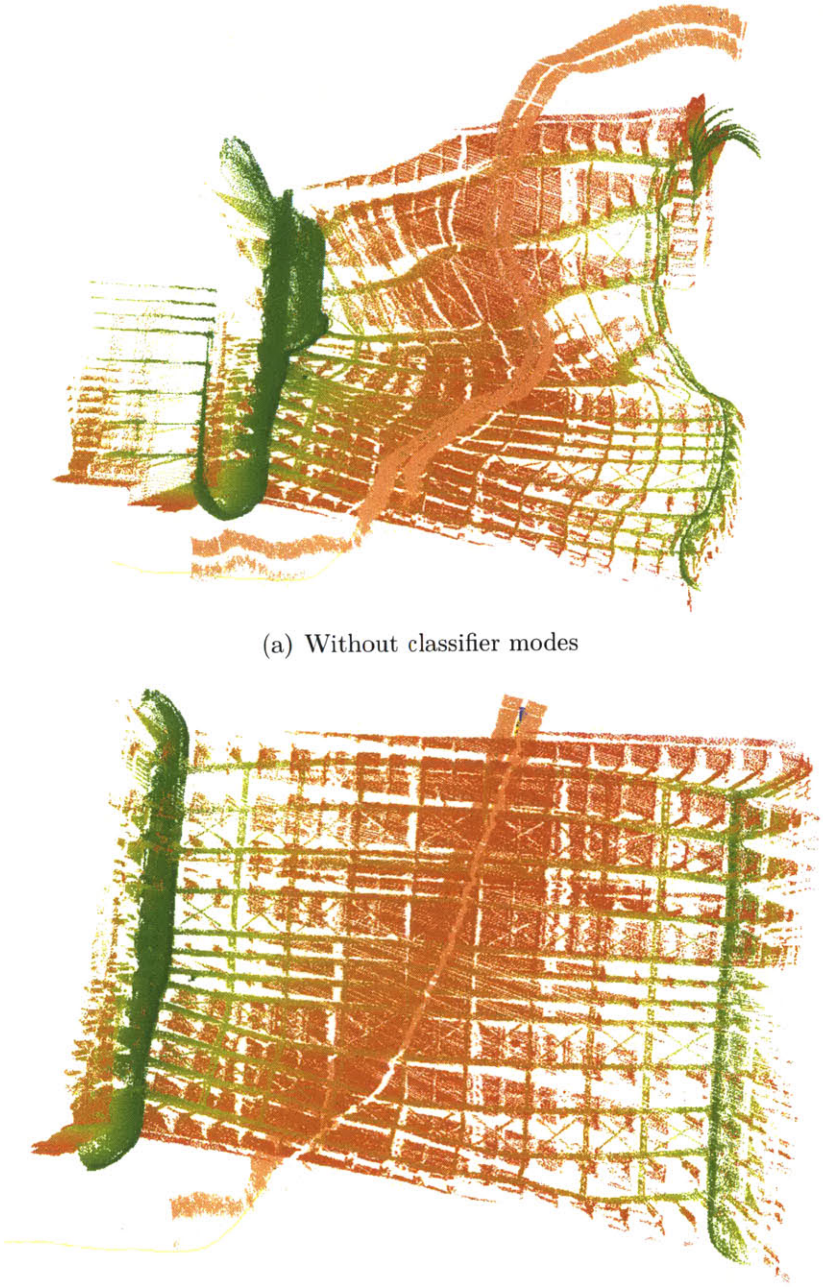


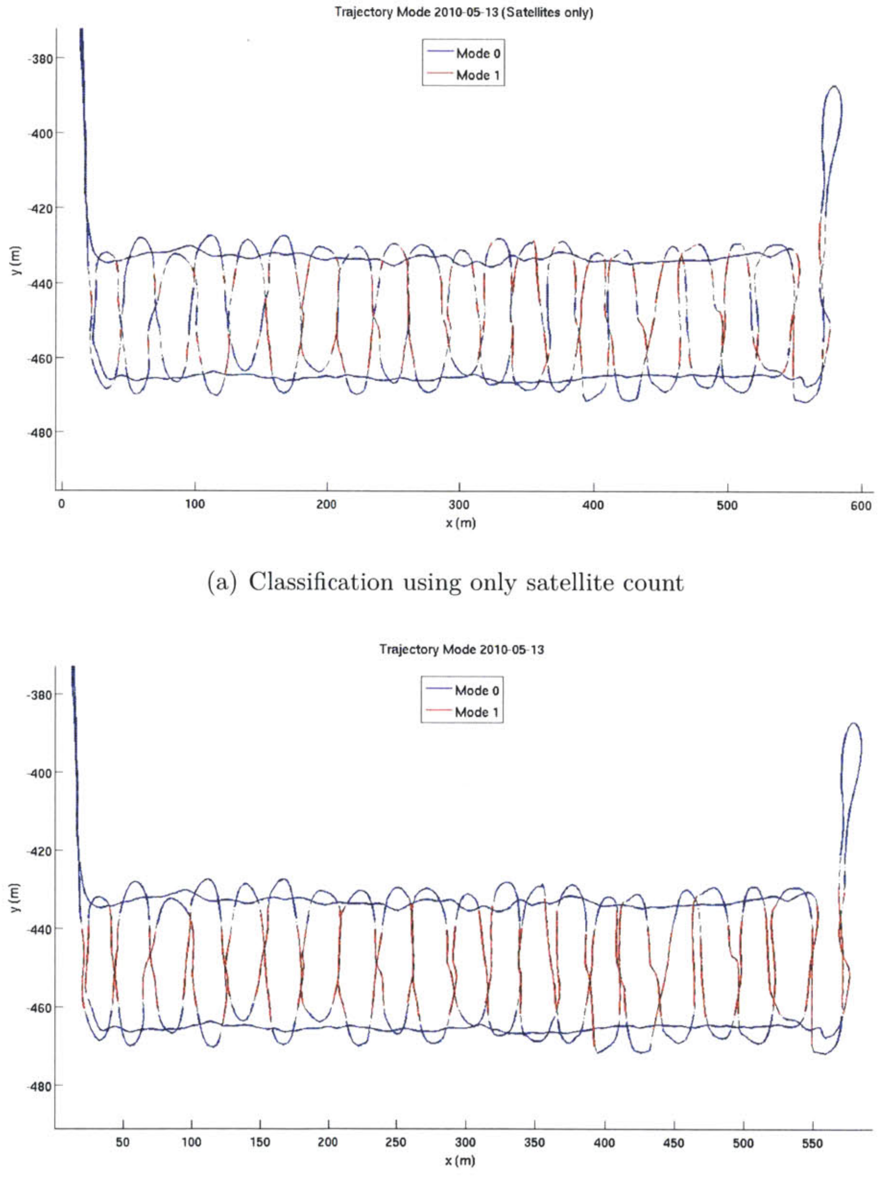
Figure 3-8: A trajectory is shown with color according to the classifier mode. A closer view is shown below, where the vehicle traversed under a bridge multiple times.



(a) Without classifier modes

(b) With classifier modes

Figure 3-9: Reprojections of range data using trajectories computed from a single vehicle model (above) and two models (below) using the classifier under a bridge. The classifier modes in this situation improve consistency as seen in the projected range data.



(a) Classification using only satellite count

(b) Classification with all features

Figure 3-10: The figures compare the result of training the classifier with an alternative feature space compared to the full feature space by coloring the trajectory by the mode given by the classifier in a region where the vehicle made many traversals under a bridge. In Figure 3-10(a), only reported number of satellites are used, while in Figure 3-10(b), the full feature space is used in training the classifier. In the top figure, the presence of the blue mode while under the bridge indicates the device reported number of satellites is insufficient for determining canopy conditions.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 4

A Hierarchical Approach to 3D Marine Mapping

The major new contribution of the thesis is an implemented algorithm for 3D marine mapping. Building on substantial recent progress in the field discussed in Chapter 2, we integrate a number of state-of-the-art techniques and approaches, including pose graphs, and divide-and-conquer multi-session SLAM with submaps. A key enabler for achieving 3D mapping results with our data has been the development of a GPS classifier that can be trained to classify degraded GPS values online.

4.1 System Overview

The overall system consists of three main stages shown in Figure 4-1. The first stage comprises the optimization of the estimate vehicle trajectory and incorporates the GPS mode classifier described in Chapter 3. The second stage consists of submap generation using an estimated vehicle trajectory. The final stage aligns submaps in the global frame to create the final map positions and the bathymetry map. The general flow of data is also shown in greater detail in Figure 4-2. In other SLAM systems, the global alignment may be coupled with the trajectory for optimization of both maps and trajectory [114]. Our system does not optimize both map and trajectory simultaneously. The constraints between submaps are not tied to a single pose; they are generated using sequences of poses, all of which are already tied to a global coordinate system (GPS). A second reason for separate trajectory and map optimization stages is that grid-based metric maps do not decouple localization uncertainty from measurement uncertainty. Feature-based maps could permit accurate localization, if one would capture the environmental complexity with a small alphabet of features and the vehicle observed a sufficient number to permit alignment to another map (loop closure). A positive consequence of the decoupling of trajectory and global map optimization is scalability, since the global map optimization does not iterate to reoptimize the trajectory and require map regeneration.

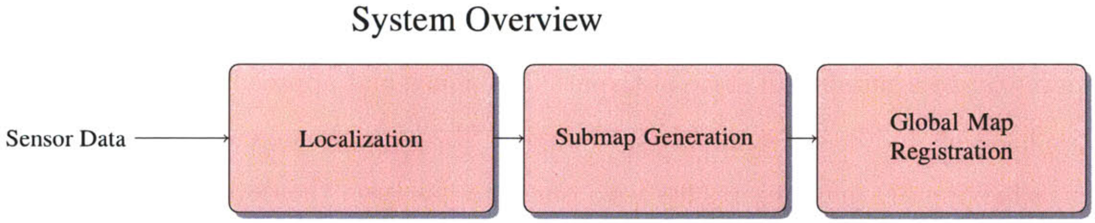


Figure 4-1: The basic components of the system. The process begins with estimating the vehicle trajectory, then submap segmentation and generation, followed by global registration and optimization of multi-session submaps.

The perceptual ability of the vehicle is limited with the sidescan vertical and horizontal lasers, see Figures B-10 and 4-3. The sidescan configuration permits observation of large areas at the expense of localization ability.

Observation is perpendicular to the principal direction of travel, and surface variation of most natural environments is high. Horizontal range observations suffer from sparse observations insufficient to constrain the vehicle in the observation plane, let

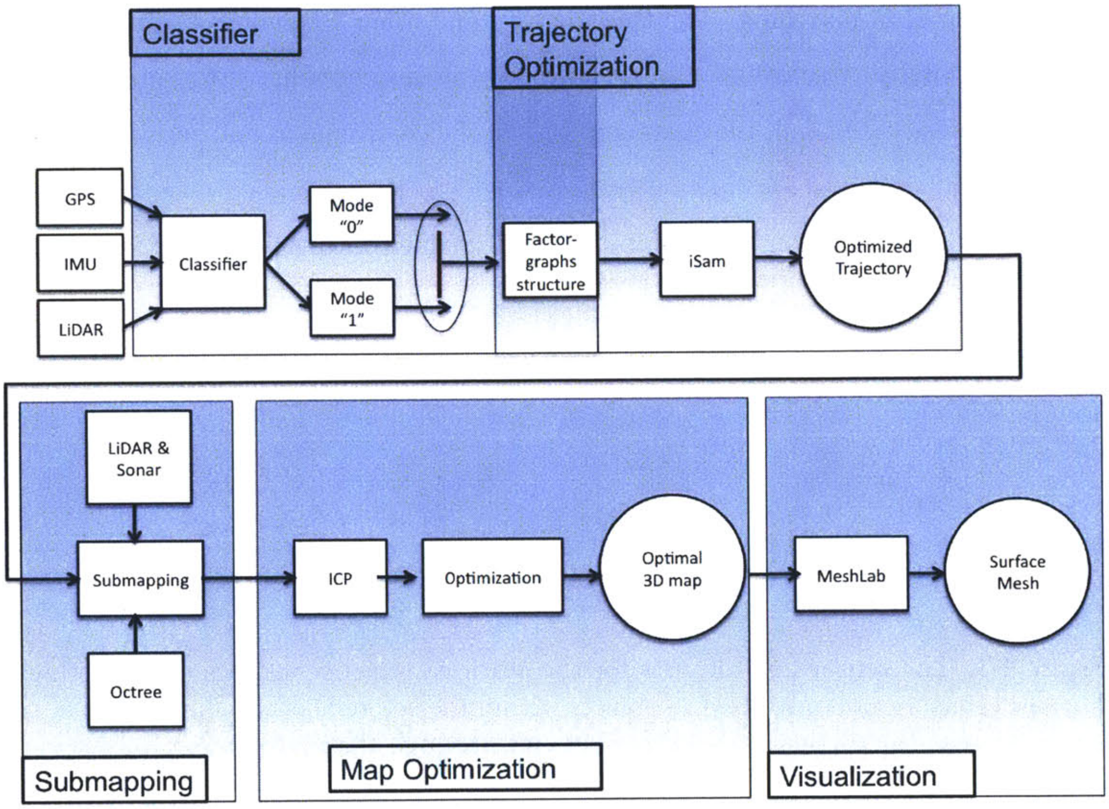


Figure 4-2: The major components in the mapping system show the flow of data through the framework.

alone in six degrees of freedom. The vehicle dynamics and travel upon a fluid, with waves making pitch and roll effects severe when near surfaces with a high incidence, such as a gentle slope on a dock or river bank or occlusions from horizontal edges of a structure close to the observation plane. Figure 4-4 shows a situation where the robot is near a river bank (see the inset camera image), and laser range data is projected onto a reconstruction of the environment. The horizontal laser beams encounter high incidence angles with the low slope of the river bank, which presents scan-matching difficulties when small changes in pitch or roll result in large differences in ranges even for sequential scans. Range data from the vertical laser sensor also presents a poor case for scan-matching. The trees, leaves, and other scene objects present little surface continuity. Occlusions and the variation in surface relief is too great for the sampling density. Overall, the characteristics of the environment and sensing abilities rarely permit accurate localization from a single or several sequential observations. Consequently localization and mapping are decoupled into separates stages in our system.

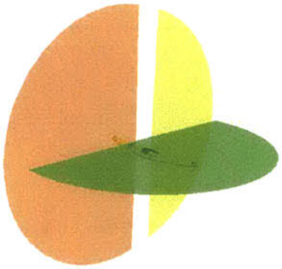


Figure 4-3: The sensor field of view for the static laser range sensors on the vehicle. The laser sensors can only observe objects within the colored disks. The motion of the vehicle sweeps the vertical sensors' field of view through the environment for mapping structures.

Smoothing and Mapping

Both the localization problem and the global alignment of the maps may be posed as a smoothing and mapping problem. Common to both the localization and global alignment stages in the system is a non-linear optimization framework built upon the iSAM library [79]. We briefly describe the fundamentals of the smoothing and mapping problem, and provide details of invocation later in the sections for localization and global alignment.

The basic probabilistic formulation for EKF-based SLAM solutions depends on a measurement model $h(\cdot)$ to explain noisy sensor measurements $\mathbf{z}$ and a predictive motion model $f(\cdot)$ for predicting robot motion from current state $\mathbf{x}_k$ and current

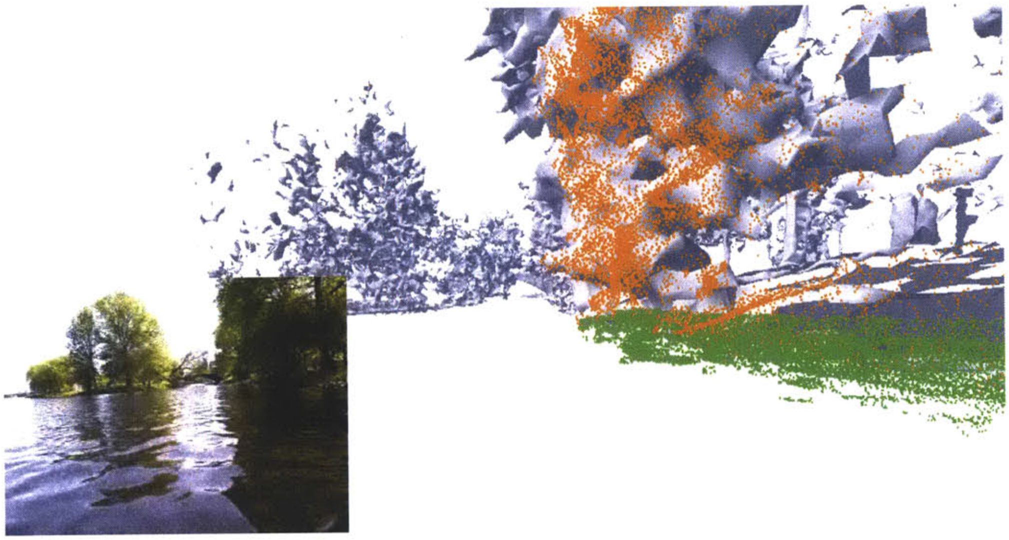


Figure 4-4: Natural environments such as one shown here present difficulty for simple feature extraction and sequential scan matching. The image projects laser ranges for a several seconds on a mesh reconstruction of the area, and a camera image is inset on the left.

control inputs $\mathbf{u}_k$. The basic assumption is that the error of the models $f(\cdot)$ and $h(\cdot)$ may be characterized by Gaussian random variables with covariances $\mathbf{Q}$ and $\mathbf{R}$, and the errors are independent. In mapping problems with explicit landmarks, such as points rather than robot poses in scan-matching, the state vector may be augmented with landmark states. For simplicity we omit specifying such states separately from the robot states. Similarly, the following discussion employs measurement models referring to a single robot state for simplicity, although models for measurements constraining multiple states are possible.

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{q}, \quad \mathbf{q} \sim \mathcal{N}(\mathbf{0}; \mathbf{Q}) \tag{4.1}$$

$$\mathbf{z} = h(\mathbf{x}) + \mathbf{r}, \quad \mathbf{r} \sim \mathcal{N}(\mathbf{0}; \mathbf{R}) \tag{4.2}$$

The Gaussian assumption of the motion and measurement models implies their probability distributions:

$$
\begin{aligned}
p(\mathbf{x}_{k+1}|\mathbf{x}_k, \mathbf{u}_k) &= (2\pi \det \mathbf{Q})^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\|\mathbf{x}_{k+1} - f(\mathbf{x}_k, \mathbf{u})\|^2_{\mathbf{Q}}\right) && (4.3)\\
p(\mathbf{z}|\mathbf{x}) &= (2\pi \det \mathbf{R})^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\|\mathbf{z} - h(\mathbf{x})\|^2_{\mathbf{R}}\right) && (4.4)\\
& && (4.5)
\end{aligned}
$$

Here we use the following notation for the inner products as Mahalanobis distances:

$$\|\mathbf{x}\|^2_{\mathbf{\Sigma}} \triangleq \mathbf{x}^T\Sigma^{-1}\mathbf{x} \qquad (4.6)$$

When considering the sequence of all robot states, observations, and control inputs for generating joint distributions, we define the aggregate sets as follows:

$$
\begin{aligned}
X &= \{\mathbf{x}_0 \ldots \mathbf{x}_K\} && (4.7)\\
Z &= \{\mathbf{z}_0 \ldots \mathbf{z}_M\} && (4.8)\\
U &= \{\mathbf{u}_0 \ldots \mathbf{u}_K\} && (4.9)
\end{aligned}
$$

To specify the association of an observation $\mathbf{z}_m$ to a particular robot pose $\mathbf{x}_k$, we use the subscript notation $\mathbf{x}_{mk}$. The joint distribution may be written as follows:

$$p(X, U, Z) = p(\mathbf{x}_0) \prod_{k=1}^{K} p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{u}_k) \prod_{m=1}^{M} p(\mathbf{z}_m|\mathbf{x}_{mk}) \qquad (4.10)$$

While the process model dictates the evolution of robot poses, the trajectory must be rooted into a valid space, which may be achieved by assigning a Gaussian prior to the initial robot pose. The initial pose estimate may incorporated into the set of measurements to simplify the discussion. When considering the full SLAM problem for smoothing, the objective is to obtain the maximum a posteriori (MAP) estimate for the unknown states X given the known inputs U, Z:

$$\begin{aligned} X^* &= \arg\max_X p(X, U, Z) \\ &= \arg\max_X p(X|U, Z) \end{aligned} \tag{4.11}$$

Substituting the distribution in Equation 4.10 into Equation 4.11 and using a log-likelihood transformation, then discarding the constant terms and common multipliers, the maximization may be transformed into the following minimization :

$$\begin{aligned} X^* &= \arg\min_X - \log P(X, U, Z) \\ &= \arg\min_X \sum_{k=1}^{K} \frac{1}{2} \|\mathbf{x}_k - f(\mathbf{x}_{k-1}, \mathbf{u}_k)\|^2_{\mathbf{Q}_k} + \frac{1}{2} \log(2\pi \det \mathbf{Q}_k) \\ &\quad + \sum_{m=1}^{M} \frac{1}{2} \|\mathbf{z}_m - h(\mathbf{x}_{mk})\|^2_{\mathbf{R}_k} + \frac{1}{2} \log(2\pi \det \mathbf{R}_m) \\ &= \arg\min_X \sum_{k=1}^{K} \|\mathbf{x}_k - f(\mathbf{x}_{k-1}, \mathbf{u}_k)\|^2_{\mathbf{Q}_k} + \sum_{m=1}^{M} \|\mathbf{z}_m - h(\mathbf{x}_{mk})\|^2_{\mathbf{R}_k} \end{aligned} \tag{4.12}$$

Using Equation 4.12, the MAP estimate may be posed as a non-linear optimization problem with the objective function J:

$$J = \sum_{k=1}^{K} \|\mathbf{x}_k - f(\mathbf{x}_{k-1}, \mathbf{u}_k)\|^2_{\mathbf{Q}_k} + \sum_{m=1}^{M} \|\mathbf{z}_m - h(\mathbf{x}_{mk})\|^2_{\mathbf{R}_k} \tag{4.13}$$

To reduce the terms in the objective function from Mahalanobis distances to Euclidean norms, the terms within the inner products may be premultiplied by the transposed Cholesky factors of their inverse covariance, also known as square root information matrices. The following example illustrates the transformation.

$$\begin{aligned}
\|\mathbf{v}\|_{\boldsymbol{\Sigma}}^2 &= \mathbf{v}^T\boldsymbol{\Sigma}^{-1}\mathbf{v} \\
&= \mathbf{v}^T\boldsymbol{\Sigma}^{-\frac{1}{2}}\left(\boldsymbol{\Sigma}^{-\frac{1}{2}}\right)^T\mathbf{v} \\
&= \left(\left(\boldsymbol{\Sigma}^{-\frac{1}{2}}\right)^T\mathbf{v}\right)^T\left(\left(\boldsymbol{\Sigma}^{-\frac{1}{2}}\right)^T\mathbf{v}\right) \\
&= \|\boldsymbol{\Sigma}^{-\frac{T}{2}}\mathbf{v}\|_2^2 \qquad (4.14)
\end{aligned}$$

Given the objective function in Equation 4.13 and using the transformation shown in Equation 4.14, the MAP estimate becomes a standard least-squares optimization. For a practical implementation of solutions to non-linear least-squares, the non-linear terms are linearized as first-order Taylor series expansions around the current estimate. Using the notation $\bar{x}$ as the current estimate of state x, the terms are linearized as follows:

$$\begin{aligned}
\mathbf{x}_k - f(\mathbf{x}_{k-1}, \mathbf{u}_k) &\approx \bar{\mathbf{x}}_k + \mathbf{I}\delta\mathbf{x}_k - f(\bar{\mathbf{x}}_{k-1}, \mathbf{u}_k) - \mathbf{F}_k\delta\mathbf{x}_{k-1} \qquad (4.15) \\
\mathbf{z}_m - h(\mathbf{x}_{mk}) &\approx \mathbf{z}_m - h(\bar{\mathbf{x}}_{mk}) - \mathbf{H}_k\delta\mathbf{x}_{mk} \qquad (4.16) \\
\mathbf{F}_k &= \left.\frac{\partial f}{\partial x}\right|_{\mathbf{x}=\bar{\mathbf{x}}_{k-1}} \qquad (4.17) \\
\mathbf{H}_m &= \left.\frac{\partial h}{\partial x}\right|_{\mathbf{x}=\bar{\mathbf{x}}_{mk}} \qquad (4.18)
\end{aligned}$$

With the terms linearized, the constants may be collected into innovation errors defined as:

$$\begin{aligned}
\mathbf{b}_k &= \bar{\mathbf{x}}_k - f(\bar{\mathbf{x}}_{k-1}, \mathbf{u}_k) \qquad (4.19) \\
\mathbf{c}_m &= \mathbf{z}_k - h(\bar{\mathbf{x}}_{mk}) \qquad (4.20)
\end{aligned}$$

Substituting the linearized terms into objective function, the objective function is in a standard least squares form.

$$\begin{aligned} J \approx & \sum_{k=1}^{K} \|\mathbf{Q}_k^{-\frac{T}{2}} \left(\mathbf{I}\delta\mathbf{x}_k - \mathbf{F}_k\delta\mathbf{x}_{k-1} - \mathbf{b}_k\right)\|^2 \\ & + \sum_{m=1}^{M} \|\mathbf{R}_m^{-\frac{T}{2}} \left(\mathbf{H}_m\delta\mathbf{x}_{mk} - \mathbf{c}_m\right)\|^2 \end{aligned} \tag{4.21}$$

The linearized formulation of the objective function may be posited as a single sparse linear system to solve for the incremental adjustments to each linearized state, as shown below in Equation 4.22. The construction of the solution in sparse matrix form is as follows and illustrated below. The entire set of states X is stacked into a single aggregate state $x = [\mathbf{x}_0 \ldots \mathbf{x}_K]^T$. Each term in the summations, which may be viewed as a constraint, in Equation 4.21 becomes a block row in the matrix form. Indices for estimated states in the summation become column block indices in the A matrix. Let the sparse rows be $\mathbf{G}_k$ for process model constraints and $\mathbf{W}_m$ for the measurement constraints. To clarify, let $\mathbf{G}_1$ be the term corresponding to the process model between states $\mathbf{x}_0$ and $\mathbf{x}_1$, and the corresponding row would be $[\mathbf{I} \;\; -\mathbf{F}_1 \; \mathbf{0} \ldots \mathbf{0}]$. Similarly for measurements affecting a state $\mathbf{x}_{mk}$, the column block corresponding to the states of $\mathbf{x}_{mk}$ contains the Jacobian $\mathbf{H}_m$ term, and all other column blocks are zero. Constructing the vector b in Equation 4.22 simply stacks the innovation terms from Equations 4.19 and 4.20. Then:

$$\delta x^* \;=\; \arg\min_{\delta x} \|A\delta x - b\|^2 \tag{4.22}$$

$$A = \begin{bmatrix} \mathbf{Q}_1^{-\frac{T}{2}}\mathbf{G}_1 \\ \mathbf{Q}_2^{-\frac{T}{2}}\mathbf{G}_2 \\ \dots \\ \mathbf{R}_j^{-\frac{T}{2}}\mathbf{W}_m \\ \dots \end{bmatrix}, \; b = \begin{bmatrix} \mathbf{Q}_1^{-\frac{T}{2}}\mathbf{b}_1 \\ \mathbf{Q}_2^{-\frac{T}{2}}\mathbf{b}_2 \\ \dots \\ \mathbf{R}_m^{-\frac{T}{2}}\mathbf{c}_m \\ \dots \end{bmatrix}$$

Solving the sparse linear system in Equation 4.22 may be done efficiently with several methods such as stochastic gradient descent or matrix factoring with column reordering; further details may be found in [43, 44, 88]. The implementation used in this thesis for solving sparse systems is based on the publicly available iSAM library [79], which uses column reordering to increase efficiency. iSAM offers *incremental* optimization to quickly incorporate new measurements and augmented states and performs complete column reordering periodically, amortizing its cost.

4.2 Localization

This section describes the localization methods generating the optimized vehicle trajectory later used for map generation. We begin by defining the states estimated and the models for the robot and sensors used within the smoothing framework described above in Section 4.1. The classification engine described in Chapter 3 is used to manage the factors generated in our localization optimization for better results. The main objective of the localization module is to fuse the sensor data into the best estimate of the vehicle trajectory. Figure 4-5 depicts the operational structure of the localization procedure.

The vehicle state is represented by nine terms: three positional states, three rotational states, and three positional velocity states in the global coordinate frame. The velocity states are necessary to permit fusion of velocity measurements from multiple

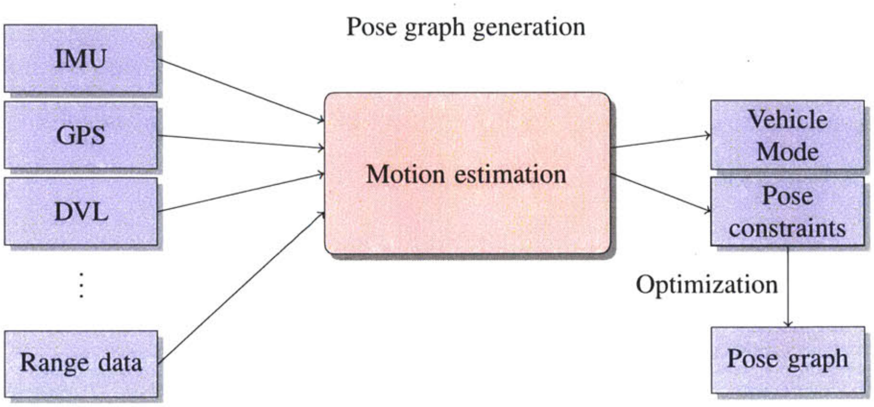


Figure 4-5: Overview of the pose estimation system. Pose estimation employs a mode classifier in the factor graph construction.

sensors, such as the IMU, GPS, and DVL. For simplicity, rotational velocities are not directly estimated since the update rate of the IMU sensor is high in comparison to the low-speed maneuvering of the vehicle.

$$\mathbf{x} = \begin{bmatrix} x & y & z & \phi & \theta & \psi & \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^T \tag{4.23}$$

The rotational states consist of the Euler angles for yaw, pitch, and roll. Rotation from vehicle to global coordinates R_v^g is defined in 4.24. The *sine* and *cosine* functions are abbreviated as c and s for brevity.

$$R_v^g = \overbrace{R_\phi}^{\text{yaw}} \cdot \overbrace{R_\theta}^{\text{pitch}} \cdot \overbrace{R_\psi}^{\text{roll}} \tag{4.24}$$

$$= \begin{bmatrix} c\phi & -s\phi & 0 \\ s\phi & c\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} c\theta & 0 & s\theta \\ 0 & 1 & 0 \\ -s\theta & 0 & c\theta \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\psi & -s\psi \\ 0 & s\psi & c\psi \end{bmatrix} \tag{4.25}$$

$$R_g^v = (R_v^g)^T \tag{4.26}$$

We assume a constant velocity model for rotational states, and constant acceleration model for positional states. As an example for a one-dimensional state, a constant velocity model is shown in Equation 4.28 and a constant acceleration model is shown in Equation 4.27.

$$x_t = x_{t-1} + \dot{x}_{t-1}\delta t + \frac{1}{2}a_t(\delta t)^2 \qquad (4.27)$$

$$x_t = x_{t-1} + v_t\delta t \qquad (4.28)$$

Let the process acceleration input be $u_t = \begin{bmatrix} \ddot{x} & \ddot{y} & \ddot{z} \end{bmatrix}^T$, where the acceleration input is in the vehicle frame. The basic vehicle motion model is shown in 4.29.

$$\mathbf{x}_t = \mathbf{F}_x\mathbf{x}_{t-1} + \mathbf{F}_u\mathbf{u}_t \qquad (4.29)$$

$$\mathbf{F}_x = \begin{bmatrix} \mathbf{I}_{3x3} & \mathbf{0}_{3x3} & \delta t\mathbf{I}_{3x3} \\ \mathbf{0}_{3x3} & \mathbf{I}_{3x3} & \mathbf{0}_{3x3} \\ \mathbf{0}_{3x3} & \mathbf{0}_{3x3} & \mathbf{I}_{3x3} \end{bmatrix} \qquad (4.30)$$

$$\mathbf{F}_u = \begin{bmatrix} \frac{\delta t^2}{2}R_v^g \\ \mathbf{0}_{3x3} \\ \delta t R_v^g \end{bmatrix} \qquad (4.31)$$

$$(4.32)$$

Equations 4.33-4.38 specify several of the measurement models used within the localization framework. The process model is shown in Equation 4.29. Initialization is performed with the factor shown in Equation 4.33 once the sensors have initialized and all states have been observed. Note the rotation matrices are functions of the vehicle state, and they require linearization to create the Jacobian matrices required for system of equations in Equation 4.21. State propagation utilizes a non-linear model.

$$\begin{aligned}
h_{Prior}(\mathbf{x}) &= \mathbf{I}_{9x9}\mathbf{x} && (4.33)\\
h_{GPS}(\mathbf{x}) &= \begin{bmatrix}\mathbf{I}_{3x3} & \mathbf{0}_{3x6}\end{bmatrix}\mathbf{x} && (4.34)\\
h_{DVL}(\mathbf{x}) &= \begin{bmatrix}\mathbf{0}_{3x6} & R^v_g\end{bmatrix}\mathbf{x} && (4.35)\\
h_{VEL}(\mathbf{x}) &= \begin{bmatrix}\mathbf{0}_{3x6} & \mathbf{I}_{3x3}\end{bmatrix}\mathbf{x} && (4.36)\\
h_{ROT}(\mathbf{x}) &= \begin{bmatrix}\mathbf{0}_{3x3} & \mathbf{I}_{3x3} & \mathbf{0}_{3x3}\end{bmatrix} && (4.37)\\
h_Z(\mathbf{x}) &= \begin{bmatrix}0 & 0 & 1 & \mathbf{0}_{1x6}\end{bmatrix} && (4.38)
\end{aligned}$$

While application of the measurement models is straight forward, elevation measurements requires special attention. In practice, GPS elevation estimates are much less accurate than the reported latitudes and longitudes. The poor accuracy of elevation is due to satellite geometry. Correction of elevation estimates from the GPS ellipsoid model to the actual terrain sea level is nearly constant for a small region of the earth surface, and the correction does not remedy the noise. Due to the high noise and periodic biases of GPS elevation and the desire to prevent accumulated dead-reckoning error, a pseudo-sensor provides elevation observations. In our application, one may assume that while the vehicle is operational, it remains afloat, and observation of operational status implies a noisy estimate of elevation. In the absence of mass or buoyancy changes, the noise is a consequence of wave motion. We incorporate a sensor model for vehicle elevation into the vehicle localization as shown in Equation 4.38. To prevent over-confidence, this pseudosensor provides observations at a low frequency with a conservative noise parameter.

4.3 Mapping

This section describes the methods used for generating maps. Here we describe a local map representation and registration methods for aligning all local maps. Although the underwater sonar maps are paired with the surface maps, the sonar processing is described in Section 4.4.

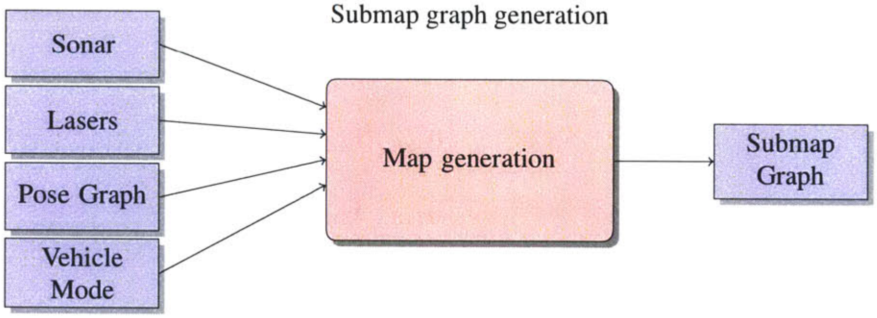


Figure 4-6: Overview of the map generation system. Map generation segments vehicle trajectories with cues from the mode classifier and creates paired surface and subsurface octree submap estimates.

4.3.1 Grid-based Metric Maps

Representation of the environment with occupancy grid maps have been a classical approach in the robotics community. The pioneers of occupancy grid maps were Elfes [48–50] and Moravec [105]. The basic principle of occupancy grids is to represent the environment as a field of binary random variables in a grid of cells. The random variable for each cell represents whether the cell is occupied or not. Occupancy grid cells may also be called pixels in 2D maps and voxels in 3D maps.

In occupancy based mapping, the environment is partitioned in cells m_k such that the map is the set $m = \{m_0 \ldots m_N\}$, and a function f that maps the cell index k to a physical location $v \in \mathbb{R}^d$, where d is typically 2 or 3. For example, consider a

10x10 grid of 100 cells of square resolution r and indexed in row major order. The corresponding mapping of index to location could be as follows:

$$\begin{bmatrix} x \\ y \end{bmatrix} = r \begin{bmatrix} \frac{1}{2} + k - \lfloor \frac{k}{10} \rfloor \\ \frac{1}{2} + \lfloor \frac{k}{10} \rfloor \end{bmatrix} \tag{4.39}$$

Each cell is associated with a value for the probability the cell is occupied. Assuming the cells are independent, the total map probability is the product of the individual cell probabilities. Given a history of robot states $x_{1:t}$ and observations $z_{1:t}$

$$p\left(m|z_{1:t}, x_{1:t}\right) = \prod_k p\left(m_k|x_{1:t}, z_{1:t}\right) \tag{4.40}$$

For practical reasons of numerical stability and ability to represent probabilities approaching one or zero, the probability is represented in log odds form, where the probability of cell m_k is $p_{k,t} = p\left(m_k|x_{1:t}, z_{1:t}\right)$, the log odds $l_{k,t}$ is computed as shown in Equations 4.41 and 4.42.

$$\begin{aligned} l_{k,t} &= \log\left(\frac{p_{k,t}}{1 - p_{k,t}}\right) && (4.41) \\ p_{k,t} &= 1 - \frac{1}{1 + \exp\left(l_{k,t}\right)} && (4.42) \end{aligned}$$

Ranges from the sensor are cast as rays into the local coordinate frame, and the cell in which a ray ends is updated as occupied. Cells in which a cast ray penetrates (but in where it does not end) are updated as free. A more detailed discussion is given in [153]. The inverse sensor model is analogous to ray tracing in computer graphics. If the sensor beam directly hits the cell, the probability of occupancy is increased, and if the beam penetrates the cell, the probability decreases. Unobserved cells are assigned a prior probability value.

$$\widehat{l}_{k,t}(z_t) = \begin{cases} l_{occ} & \text{if beam ends in cell } k \\ l_{free} & \text{if beam passes through cell } k \end{cases} \tag{4.43}$$

$$l_{k,t} = l_{k,t-1} + \widehat{l}_{k,t}(z_t) \tag{4.44}$$

Specific examples of recent research using 3D metric maps with laser sensors include Bosse et al. [26] and Ryde et al. [135].

In [25] Bosse et al. extended the traditional parameterization of voxels. Voxel parameters were the mean and covariance of the enclosed samples, and derived measures of planarity and cylindrical-likeness based upon the eigenvalues of the sample covariance. The utility of the parameterization was in aiding correspondence between metric maps. Similar use of eigenvalues as measures of curvature are defined in [123] by Pauly et al. and used used for normal estimation and outlier rejection in point clouds of indoor environments by Rusu et al. [133].

The motivations for using the metric map representation are as follows. Metric maps do not assume known structure as do feature based methods. Consistent use of resolutions and anchoring submap coordinates provides uniformity among the map representations to facilitate map matching from multiple sessions or reentrant traversals. Regular and consistent space partitioning within the maps provide a means to simplify or reduce maps to lower resolution to address computational and memory constraints.

For large scale environments a memory efficient map representation is necessary to perform most operations such as visualization or operations on multiple maps like co-registration. In problems where a dense representation is unnecessary, tree-based space partitioning provides substantial memory savings over a full dense grid representation. Environments in many robotic applications are naturally sparse. For comparison, estimating tissue density ([71, 137]) at high resolution within a human head would be a dense problem that would be better suited with a dense representation. Sparse voxel octrees (SVO) provide regular recursive spatial decomposition for three dimensional volumes [89, 136, 139, 160, 162]. Other popular spatial partitioning structures are kd-trees [18, 19] and more generally binary spatial partitioning [122].

Local maps are generated in an incremental manner from the optimized vehicle trajectory and the range sensor data. Within a session, partitioning of the maps occurs when a local map nears the extent of representation or at vehicle mode transitions. The first case is a consequence of the octree representation since the addressable grid cells map to physical locations bounded by a multiple of the map resolution. Vehicle model changes provide information that the state of localization quality has changed, which hint toward a need for alignment. The mode change segmentation policy is critical to achieving global consistency. Apparent misalignment of the raw data suggests local biases in GPS derived positioning for structures such as a bridge observed on different days or observed at different times on the same day after intermediate signal obstruction. By segmenting maps at mode changes, the effects of changing satellite constellation biases are deferred for correction by the global submap registration and alignment.

4.3.2 Metric Map Matching

Global optimization of the submaps requires constraints between the submaps, and here we describe the registration methods to generate constraints between submaps. Each metric map is converted from an occupancy grid to a binary voxel tree by thresholding the likelihood values of the cells. After thresholding, the voxels of a submap may be treated as point cloud, permitting alignment between maps with point-wise registration methods such as ICP.

Given a set of submaps $\mathcal{M}$ generated from the vehicle trajectory, we search for alignment constraints between submaps. Using the bounding volume of each submap, we search for candidates for registration. For two submaps m_i and m_j having bounding volumes V_i and V_j, the heuristic metric ρ in Equation 4.45 determines whether or not registration is attempted. We use a conservative (.10) threshold, meaning both candidates considered for alignment must span at least 10% of the volume of space of the smaller map. This gating statistic prunes the search space reducing the run time of search. It also ignores map pairs unlikely to co-register in the actual registration routine, thus reducing computational time and latency in loading the maps from disk. Preprocessing for registration also uses the intersection of bounding volumes as an approximate region of interest (ROI) and improves correctness and performance of the ICP implementation used in this work. Map pairs having both significant overlap and significant disjoint volumes do not converge well without clipping. The overall percentage of matching points may be small relative to total points in the maps. Large numbers of points with low likelihood of matching increase computation time and memory usage. An ICP implementation using randomized sampling might suffer without ROI clipping due to increased likelihood of sampling unmatchable points.

$$\rho = \frac{||V_i \cap V_j||}{\min\left(||V_i||, ||V_j||\right)} \tag{4.45}$$

The ICP implementation used in this work is based upon the method shown in [73]. ICP iteratively computes correspondences between the input point sets and the optimal transformation given those correspondences until convergence. The rigid body transformation consists of a three dimensional translation vector $\mathbf{t}$ and rotation matrix $\mathbf{R}$. The desired transformation minimizes the error in Equation 4.46. Let $\mathbf{x}_i$ and $\mathbf{y}_i$ represent the ith point in the two input maps after correspondences are chosen.

$$E(\mathbf{t},\mathbf{R}) = \sum_{i=1}^{N} \|\mathbf{x}_i - (\mathbf{t} + \mathbf{R}\mathbf{y}_i)\|^2 \tag{4.46}$$

The transformation minimizing Equation 4.46 is computed as follows. First, the matched input point sets are de-meaned; the centroid is subtracted from points in the set. Letting $\tilde{\mathbf{x}}_i = \mathbf{x}_i - \bar{\mathbf{x}}$, $\tilde{\mathbf{y}}_i = \mathbf{y}_i - \bar{\mathbf{y}}$, the error function may be written as follows.

$$\begin{aligned} E(\mathbf{t},\mathbf{R}) &= \sum_{i=1}^{N} \|\tilde{\mathbf{x}}_i - \mathbf{R}\tilde{\mathbf{y}}_i - (\mathbf{R}\bar{\mathbf{y}} + \mathbf{t} - \bar{\mathbf{x}})\|^2 \\ &= \sum_{i=1}^{N} \|\tilde{\mathbf{x}}_i - \mathbf{R}\tilde{\mathbf{y}}_i\|^2 + \|\mathbf{R}\bar{\mathbf{y}} + \mathbf{t} - \bar{\mathbf{x}}\|^2 \\ &\quad -2(\tilde{\mathbf{x}}_i - \mathbf{R}\tilde{\mathbf{y}}_i) \cdot (\mathbf{R}\bar{\mathbf{y}} + \mathbf{t} - \bar{\mathbf{x}}) \end{aligned} \tag{4.47}$$

The final term of Equation 4.47 is zero, since both $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{y}}$ have mean zero. The second term, which is constant over the summation, is minimized at zero, where the optimal translation is shown in Equation 4.48. So the optimal translation is a function of the optimal rotation.

$$\mathbf{t} = \bar{\mathbf{x}} - \mathbf{R}\bar{\mathbf{y}} \tag{4.48}$$

The optimal rotational component minimizes the first term in Equation 4.47. The optimal rotation is calculated using the formula given in [73], where the quaternion representing $\mathbf{R}$ is the eigenvector with the largest eigenvalue of the following matrix of covariance terms. Subscripts of elements in the matrix refer to the scalar components

(x, y, z) of the vectors $\mathbf{x}$ and $\mathbf{y}$.

$$\begin{bmatrix} s_{xx}+s_{yy}+s_{zz} & s_{yz}-s_{zy} & s_{zx}-s_{xz} & s_{xy}-s_{yx} \\ s_{yz}-s_{zy} & s_{xx}-s_{yy}-s_{zz} & s_{xy}+s_{yx} & s_{zz}+s_{xz} \\ s_{zx}-s_{xz} & s_{xy}+s_{yx} & -s_{xx}+s_{yy}-s_{zz} & s_{yz}-s_{zy} \\ s_{xy}-s_{yx} & s_{zz}+s_{xz} & s_{yz}-s_{zy} & -s_{xx}-s_{yy}+s_{zz} \end{bmatrix}$$

An alternative calculation of the rotation matrix using singular value decomposition of the covariance matrix between the point sets is shown in [5].

4.3.3 Global Map Alignment

This section describes the procedure used to globally align submaps. The global alignment consists of two main steps. First, a graph is constructed with submaps as nodes and pair-wise registrations as edges. Second, the graph is posed as an optimization problem to obtain the final positions of submaps.

Constructing the graph of submaps requires a search for the edges. Since map registration is computationally intensive with maps consisting of hundreds of thousands to millions of voxels, an outer gating threshold determines whether a registration attempt should be made. Although registration will fail quickly for maps too distant from each other, the latency of loading the maps into physical memory may be avoided altogether using the gating threshold, which depends upon pre-computed information about each map, such as the bounding volume. Algorithm 1 shows how the set of all possible map pairs are reduced to a smaller set of candidates having better chances of successfully registering. The procedure in Algorithm 2 performs the registration, and successful registrations produce edges in the graph.

The edges generated from the pair-wise registrations are constraints between pairs of maps. For example, consider maps m_a and m_b have an edge, and let the rigid body transformation be T_{ab}. The transformation between the map origins are related as $T_{ab} = x_b \ominus x_a$ The notation $x_b \ominus x_a$, which alternatively may be written as $\ominus x_a \oplus x_b$, follows the convention seen for spatial relationships in [142]. Figure 4-8 illustrates an example. The transform vectors incorporate the displacement between nodes and the rotations required to convert one coordinate frame to another.

Algorithm 1: Generation of hypotheses for map registration

```
GenerateEdgeHypotheses($\mathcal{M}$)
$Q \leftarrow \emptyset$
for $m_a \in \mathcal{M}$ do
    for $m_b \in \mathcal{M} - \{m_a\}$ do
        $rho \leftarrow$ matchPotential $(m_a, m_b)$
        if $o \geq \bar{\rho}$ then
            $Q \leftarrow Q \cup \langle \rho, a, b \rangle$
        end
    end
end
```

Algorithm 2: Construction of the graph by verifying possible pair-wise constraints

```
GraphBuild($\mathcal{M}$)
$\langle V, E \rangle \leftarrow \langle \mathcal{M}, \emptyset \rangle$
$H \leftarrow$ GenerateEdgeHypotheses $(\mathcal{M})$
for $h \in H$ do
    $T_{ab} \leftarrow$ register $(h.a, h.b)$
    if $T_{ab} \neq \emptyset$ then
        $E \leftarrow E \cup \langle a, b, T_{ab} \rangle$
    end
end
```

Global optimization of the graph of submaps and their pair-wise constraints is performed as a non-linear least squares problem in similar fashion to the trajectory optimization. In graphical form, the nodes are base coordinate frames of the submaps, and constraints are the rigid body transformations resulting from successful registrations. A unary constraint is added for each map as the prior for that map's pose. The prior constraints are a direct consequence of the map generation using GPS based localization, a global coordinate frame. This contrasts with methods seen in frameworks such as Atlas [27], where the graph is topological, and global alignment requires picking a root node to assign a prior (typically the map where the robot starts). In systems where measurements lead to only relative constraints, such as odometry, scan matching, or visual loop closures, a single prior is usually enforced to root the system into a single Euclidean coordinate system. With GPS sensors, priors exist on each node in the submap graph. From a graphical perspective, each map becomes a node in the factor graph and has a unary factor for a prior. The rigid transforms from successful registrations become binary factors. Figure 4-9 illustrates a small two dimensional example. Figure 4-9(a) shows bounding boxes of submaps. Overlapping regions in this example imply a successful registration. Figure 4-9(b) illustrates the resulting factor graph.

The system of equations for global map optimization is constructed in identical fashion to the smoothing and mapping SLAM formulation used in the vehicle trajectory estimation. The mathematical formulation is as follows. Let constraint c_j be $z_j = T_{a,b}$ be the rigid body transform between maps m_a and m_b, and $\bar{x}_k$ be the prior for map m_k. The measurement function h_j for constraint z_j becomes $h_j(x) = x_b \ominus x_a$, and $J_0 = \nabla_{x_a} h_j(x)$ and $J_1 = \nabla_{x_b} h_j(x)$. The error function in Equation 4.49 consists of terms for the prior of each map and terms for the pair-wise registration constraints. Linearization of the system leads to the matrix of Jacobians in Equation 4.50 and the residual vector in Equation 4.51. Although not shown, each row is pre-multiplied by the inverse Cholesky factor of the constraint covariance matrix, or the square root information matrix. The resulting system is batch optimized to obtain the final map locations.

$$E = \sum_{i=0}^{N-1} \|\bar{x}_i - x_i\|^2_{\Sigma_i} + \sum_{j=0}^{C-1} \|z_j - h_j(x)\|^2_{R_j} \tag{4.49}$$

$$A = \begin{bmatrix} I_{6x6} & \dots & \dots \\ 0_{6x6} & I_{6x6} & \dots \\ J_0 & J_1 & \dots \\ \dots & \dots & \dots \end{bmatrix} \tag{4.50}$$

$$b = \begin{bmatrix} \bar{x}_0 - x_0 \\ \bar{x}_1 - x_1 \\ T_{0,1} - x_1 \ominus x_0 \\ \dots \end{bmatrix} \tag{4.51}$$

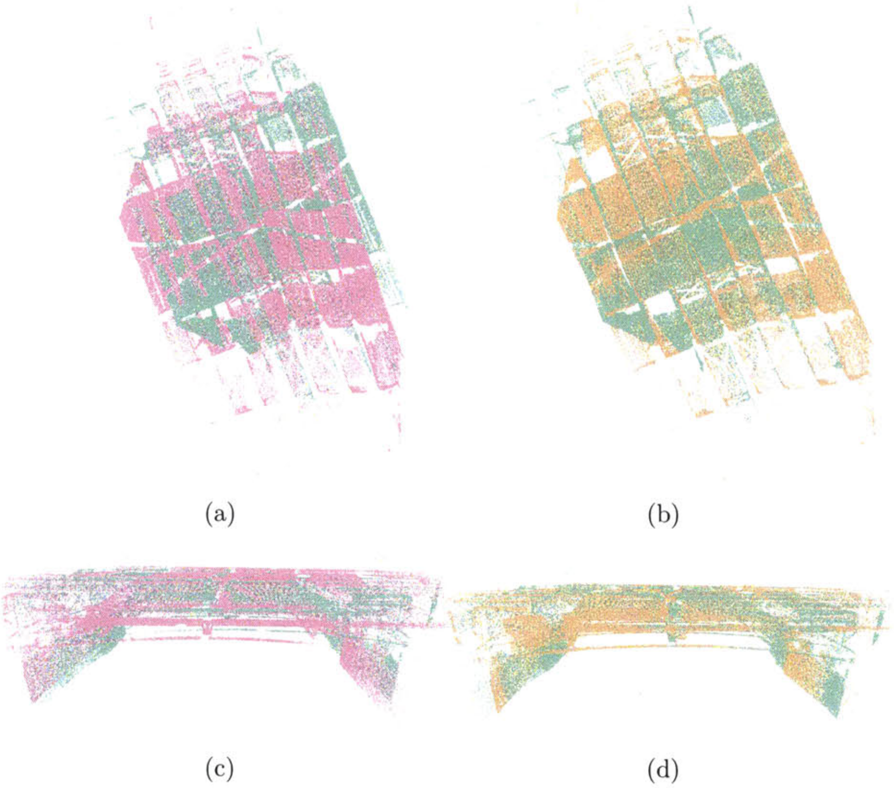


Figure 4-7: Two submaps aligned with an ICP registration with top and side views. Prior to registration the maps appear as shown in 4-7(a) and 4-7(c) in green and magenta, and after alignment they appear as shown in 4-7(b) and 4-7(d). The aligned magenta submap is now shown in brown.

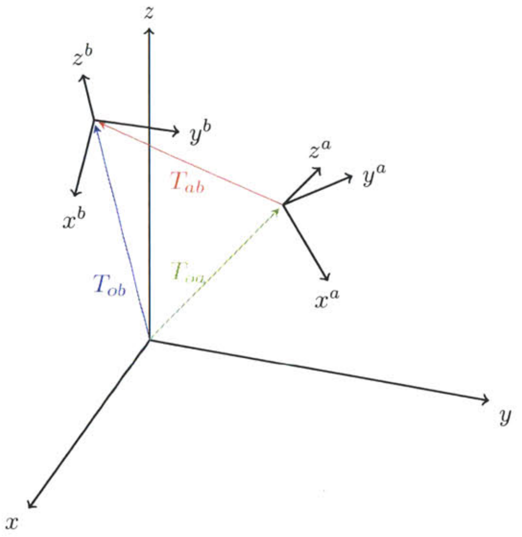
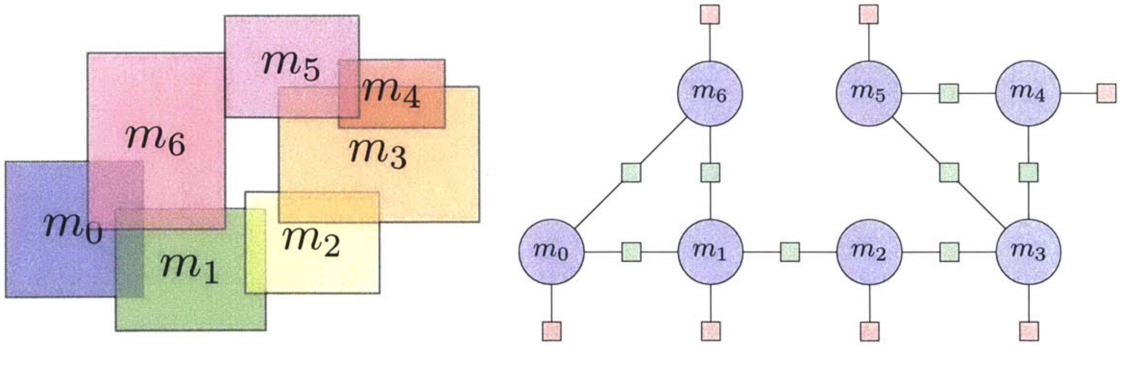


Figure 4-8: Example of two nodes with distinct poses in a global frame and the transformation between them. The colored vectors represent the transformations between the origin and nodes, and axes are labeled to illustrate the rotational information in the transformations. In the global alignment procedure, submap origins are nodes and the successful alignment transformations provide the edges between nodes.



(a) Map areas

(b) Factor graph

Figure 4-9: An example of a small factor graph for the types of factors in the global alignment is shown on the right. At left bounding volumes are shown for the maps and overlap implies a successful registration, or edge, between the maps. Each node (map) has a unary factor as a prior due to GPS aided localization. Successful registrations between maps result in the binary factors between nodes.

4.4 Sonar Mapping

Underwater maps created from sonar sensing share the same initial steps as the surface maps, however they differ in image processing and the construction of a final bathymetry map. Due to the noise characteristics of the sonar, low sensor coverage, and low variation observed in the underwater terrain, the sonar maps are not used for registration. The sonar submaps inherit final positions from their sister surface maps after global alignment.

4.4.1 Sonar Image Processing

Each ping of the imaging sonar provides an image covering the 45 degree field of view within the sensor's operating range of one to ten meters. Pixel values represent intensities of the acoustic sonar returns. See Figure 4-10 for an example image. Processing of this image begins with an implementation of the Canny Edge Detector [29]. A one-dimensional range scan is created taking the range pixel with maximum intensity value preceded with a positive gradient for each bearing angle. Ranges less than 1.0 meters or having intensity values below a threshold are ignored.

4.4.2 Bathymetric Mapping

Given the optimal alignment of local maps from the method given in Section 4.3.3, a global bathymetry map is generated. The global bathymetry map is generated as a digital elevation map (DEM). Although the surface maps are general three dimensional metric maps, the global bathymetry map provides a single elevation value for a given (x, y) location. Due to the difference in noise characteristics of the sonar sensor in comparison to the surface lidar sensors and the absence of experimental data observing objects with topology requiring multiple occupancy in a grid location,

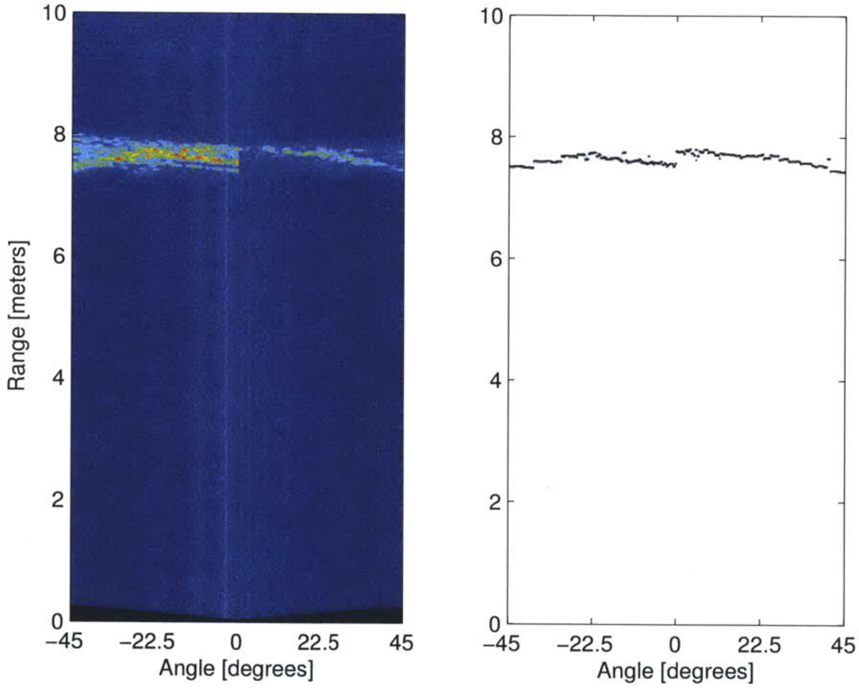


Figure 4-10: Example sonar ping image and corresponding range data. Image on left is color mapped for visibility, and image on right overlays extracted range data over the raw gray scale intensity image. Note the sonar is constructed of two physical staves, which causes the discontinuity.

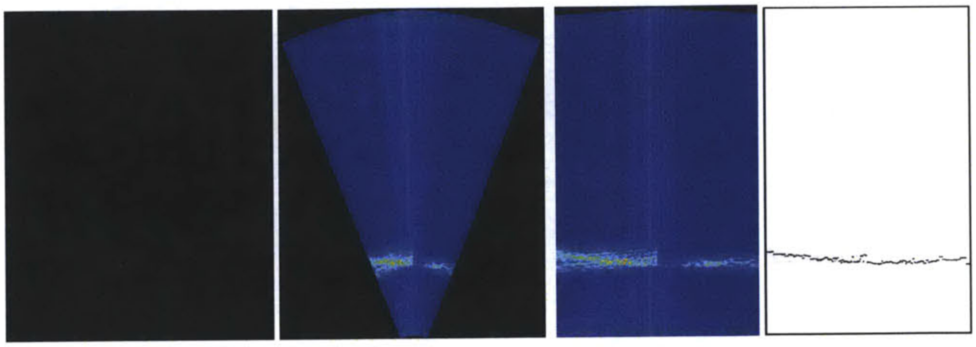


Figure 4-11: An example ping image with different visualizations. At the far left, the raw intensity image is shown mapped to Cartesian coordinates, and the dynamic range of the input is difficult to see. The image second from left is the same image with false coloring for better visualization. Third from left is a false colored image of the ping in the more natural polar coordinates of the sensor. On the far right, is the polar intensity image, which is the form used in processing.

a digital elevation map representation was chosen.

Like the surface maps, the subsurface maps are initially represented as a sparse voxel octree. The details for voxel updates are vary slightly for sonar data to incorporate the intensity values. A function maps intensity values to occupancy probabilities : $f : I \mapsto p$. The function domain is specific to the sensor, which provides intensity values in the range of 0 to 65535. Occupancy probabilities must be above 0.5. The function range is compressed slightly to 0.60 and 0.90 to limit dominance of outliers or non-contribution of numerous weak observations. More critically, the domain is clamped within two thresholds: the lower threshold (the noise floor) discards observations of weaker intensity and the upper threshold (saturation intensity) limits the input domain to maximum value. Once intensities are clamped, a linear mapping of the log of intensity to the output occupancy probability determines the occupancy update probability. Other than the use of intensity values to influence the voxel updates, the subsurface map generation is identical to surface map generation.

Algorithm 3: Sonar map voxel updates

```
input  : A sonar range intensity: $i$
output: A log odds value for updating a voxel: $l$
if $i < noiseFloor$ then
    return $0$
end
if $i > saturationValue$ then
    $i \leftarrow saturationValue$
end
$i \leftarrow \log i$
$s \leftarrow \frac{p_{hi} - p_{lo}}{\log(saturationValue) - \log(noiseFloor)}$
$p \leftarrow p_{lo} + s\,(i - \log(noiseFloor))$
$l \leftarrow \log \frac{p}{1-p}$
```

Creation of the bathymetry map as an elevation map requires reduction of the vertical slices of the tree to a single z value. Regions observed more than once often have multiple occupied values in the voxel tree due to the noise of sensor, aspect dependence of the underwater acoustics with respect to observed riverbed, acoustic effects not modeled in the range extractor, fish, and lack of well defined surface boundary in the mud. In many cases, there are scattered occupied cells within what should be the water column that must be handled as outliers to achieve a smooth and consistent map. Outliers tend to create unwanted peaks in the elevation map. To address this problem, a neighbor consistency check is performed to remove the obvious outliers from a given location by removing cells too far from any cells in nearby locations. This procedure, shown in Algorithm 4, is performed in two iterations over the local voxel trees prior to the final elevation map creation. For the final bathymetry map, vertical columns of voxels in a given location are reduced to a single mean value.

Mesh generation from the final bathymetry map uses triangulation of the neighboring cells.

Algorithm 4: Bathymetry map filtering: consistency check

```
input  : A data tuple $d_0 = (z, p, flag)$ at node $n_0 = (x, y, list < d >)$ and
         threshold $distThresh$
output: Valid bit $d_0.flag$ when neighbors exist and validate $d_0$
$k_0 \leftarrow$ xy2key($n_0.x, n_0.y$)
$K \leftarrow$ neighborKeys($x, y, distThresh$)
if $K = \emptyset$ then
    return 1
end
invalidate data unless valid neighbor exists
$d_0.flag \leftarrow 0$
for $k \in K$ do
    $n \leftarrow$ key2node($k$)
    if $n \neq \emptyset \wedge n.data \neq \emptyset$ then
        for $d \in n.data$ do
            if $d.flag$ then
                $d_0.flag = 1$
            end
        end
    end
end
return $d_0.flag$
```

4.5 Mesh Generation

This section describes the generation of mesh reconstructions for the surface maps. Surface reconstruction from the submaps is performed for visualization. Unlike the sonar bathymetry mesh generation, mesh reconstruction of the surface maps employs the Ball-Pivoting algorithm (BPA) [20]. We use an open-source implementation of BPA provided in the Visual Computing Library [35] as part of the application Meshlab [34]. The ball pivoting algorithm exhibits linear complexity with the number of input vertices, does not require all data to be resident in memory, is relatively fast compared to other mesh reconstruction methods, and reasonably tolerant of noisy samples. Various other surface reconstruction methods were attempted, and the ball pivoting algorithm produced the best qualitative results with consistency among the alternatives. One alternative method was a Poisson surface reconstruction [81]. The Poisson method produced fast closed surface reconstructions. Generally the reconstructions did not preserve the true topology, and the method would be more suitable for closed surfaces without many holes.

To provide more complete sampling of the surfaces to yield more consistent faces, submaps having a high enough overlap and successful registration are merged into single maps prior to generating surfaces. The rule for merging involved the overlap metric used in Equation 4.45, thresholded at 0.60, and more significantly, the registration successfully matched 75% of points in *both* maps. Additionally, the merged map could not exceed the volume representable in SVO structure.

The reconstructed mesh surfaces may be used in alternative registration methods using information such as the normals for matching.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 5

Experimental Results

In this chapter we describe the experiments conducted. We begin with a description of the operational setup, including a description of the environment and operating procedure. Next we present the results using the methods described in Chapter 4. First, we discuss the localization results. Next we present the mapping results with a discussion of the submaps generated using our method and validation of selected surface submaps with a prior model of a bridge. We provide bathymetry results and additionally fuse bathymetry data collected with an AUV for greater coverage of the operating region. We conclude the chapter with a discussion of the results, including limitations to our methods.

5.1 Experimental Setup

The data presented in this thesis was collected in the Charles River between Boston and Cambridge Massachusetts over a two week period from late April to early May of 2010. The operational area was approximately 1.6km by 1.1km, and a satellite image of the region is shown in Figure 5-1. The location was chosen for the variety of structures which could be mapped from the vehicle as well as the proximity. Of the structures in the region, there are three bridges, two sailing pavilions, and a yacht club. At least one of the structures, the Harvard Bridge, has publicly available blueprints which are used in generating a model for quantitative validation in Section 5.4. The Cambridge side of the river in the operating area has a stone retaining wall, but the Boston side has a natural river bank.

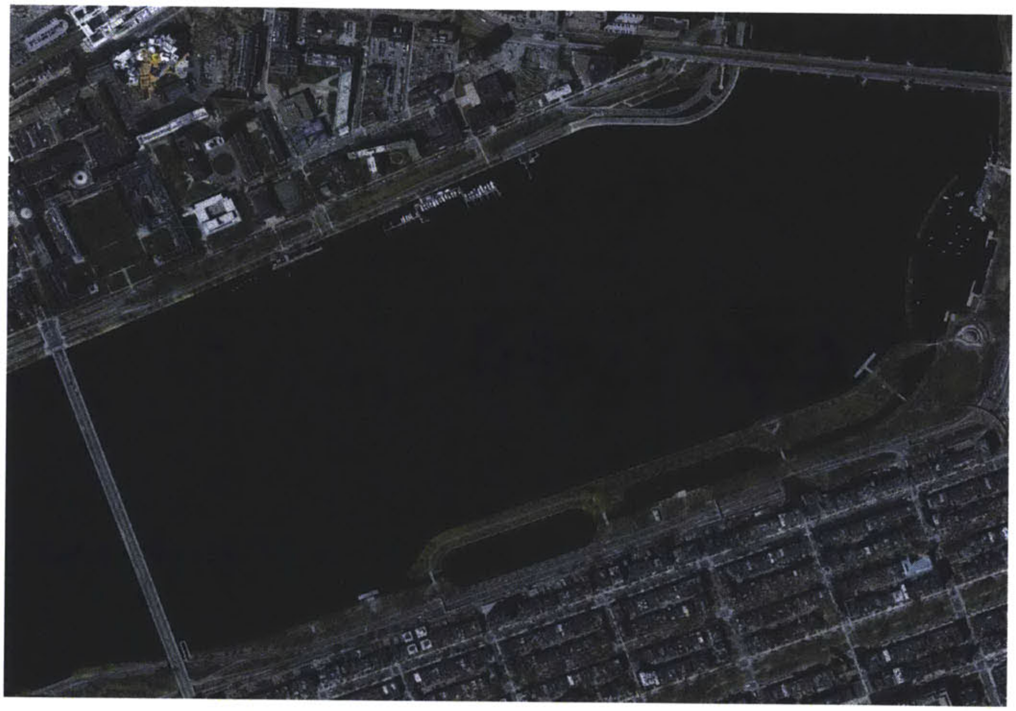


Figure 5-1: A satellite view of the operating area for experiments. The Harvard Bridge is on the western edge of the operating area and the Longfellow Bridge is on the northeastern edge of the operating area.

During the course of experiments, nine separate missions of appreciable duration were collected. Of those nine, the longest six were processed and results shown in this section. The duration of these missions ranged from 30 to 90 minutes, and the vehicle traveled nearly 7 kilometers during the longest mission. Those six missions had identical sensor configurations.

Weather conditions during experimentation were generally clear with varying levels of wind. Rain was not a critical factor in the missions processed. While rain is not a hazard to the platform, the ranging sensors as well as the camera lack a mechanism to maintain clear lenses, and the actuated laser scanners were not fully protected. The wind conditions varied significantly, and wind is a significant factor in vehicle control and the amplitude of surface waves. Waves resulting from windy conditions are the primary cause of pitching and rolling motions of the vehicle. Water current was not a significant factor during the experiments, and water level did not change appreciably during the experiments.

Operating procedures for surface craft in complex marine environments have little contemporary publication, with the notable exception in [106, 107] documenting operations in an environment very similar to this work. We employed a hoist to deploy and recover the vehicle from the water, as shown in Figure 5-2. Although a possibility, the vehicle was not deployed from a ramp to maintain the integrity sonar mounting configuration. Without a sonar mounted under the hull, one may easily deploy the vehicle from a ramp by placing the vehicle thruster over the ramp edge and sliding the vehicle into the water. Each experiment had a human operator remotely controlling the vehicle with a radio control (RC) receiver. A small chase boat pursued the vehicle to maintain RC range, navigate the vehicle relative to structures, and address any hazards. The chase boat avoided navigating into the sensing field of the robot to prevent corruption of the data. For example, one mission is manually segmented into two data sets due to the propeller hitting a shallow rock resulting in a disengaged servo gear which required manual intervention to reset the gear. A similar experience occurred when the propeller struck a submerged diagonal support strut for a bridge. Loss of radio control also presented an operational hazard. Only a few of the control losses required manual intervention, usually due to an actual vehicle hazard, such as the thruster hitting a submerged object and disengaging the gears in the servo mechanism. Future work would change the settings on the vehicle support board's servo switch, which by default has a seven second safety delay for reactivation after a signal loss. Removal of the safety delay would allow faster reacquisition of control, but could result in collisions if the signal is lost in a very cluttered environments.

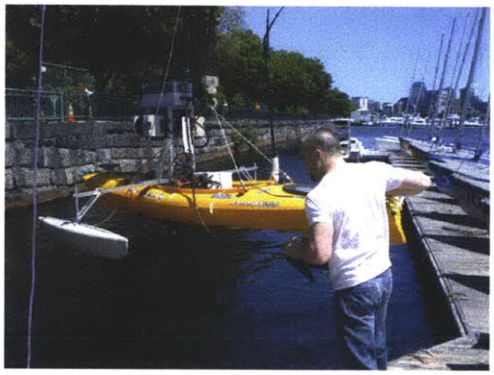


Figure 5-2: The small craft permits rapid vehicle launch and recovery with the aid of an electric hoist. Usually one operator performed the procedure, although we recommend two operators.

Two of the missions terminated prematurely due to under-voltage conditions, which causes sensors to fail. One of those resulted in incomplete coverage of the Longfellow Bridge, as sensors began to fail while under that bridge. In that particular case, the DVL failed first, then the IMU. After the actuated laser assembly failed, the DVL briefly returned to operation before failing again.

Vehicle operators employed a general coastal navigation strategy during the experiments, where coastal is with respect surface structures and not limited to the river boundaries. The purpose of the strategy is to navigate the vehicle such that surface structures remain within the sensor field of view. A consequence of the coastal strategy is forfeit of bathymetric data for more central portions of the river. One may note portions of the Charles river are deeper than the ranging limits of the sonar sensor, even if the sensor mounting had been straight down. The sparse bathymetry coverage is shown in Section 5.5 as well as a more complete bathymetry map generated by fusing results from several of our prior experiments with an AUV.

5.2 Localization

In Figures 5-3 and 5-4 we show the optimized trajectory for a mission on May 13, 2010 in comparison to trajectories computed by other methods. Similarly, Figures 5-5 and 5-6 show trajectories optimized for a mission on May 11, 2010. The second figures in each pair show a closer view of portions of the trajectories. In each figure, the trajectory shown in blue is the GPS position with minimal outlier rejection, and the trajectory shown in red is based upon integration of the IMU orientation and the GPS-based velocity. The trajectory shown in green is dead-reckoning using IMU orientation and DVL velocities with a *hold* strategy for when DVL briefly loses bottom lock. The dark purple trajectory represents the optimized trajectory. As expected with dead-reckoning, error accumulates over time, as seen with DVL and GPS-based velocity paths. The GPS position path usually stays near the optimized trajectory but is locally non-smooth. The non-smooth behavior is most apparent near structures, as seen in Figure 5-4(a) and 5-6(a). Perhaps most apparent in the figures illustrating elevation over time is the higher uncertainty in elevation estimates. After fitting a bias correction, the DVL dead-reckoning shows minor error accumulation. GPS position fixes (blue in the figures) do not accumulate error over time but lack local smooth smoothness seen in dead-reckoning trajectories. Elevation estimates are particularly noisy and may be seen in Figures 5-3(b),5-4(b)5-5(b), and 5-5(b). Traversals underneath the bridge have a visible effect on the error. For example, Figure 5-4 shows a portion of a trajectory where the vehicle first enters a canopy condition around time 1020 seconds and continues to weave underneath the bridge.

The relatively stable IMU dead-reckoning as well as the GPS position fixes show noticeable oscillations as the visible satellite constellation changes.

On the scale of the operating area, localization error is difficult to detect from the trajectory alone. The error becomes much more apparent when visualizing the ranging sensor data, as the ranges naturally leverage the localization error into the global frame. For example in Figure 5-7, the first image shows surface laser data projected using the trajectory derived from the IMU and GPS-velocities. Looking back at Figure 5-3(a), that trajectory demonstrates a low drift rate in comparison to the DVL trajectory in the (x, y) plane. In Figure 5-7, the localization error leads to a significantly distorted point-cloud for a bridge. The lower figure shows the final mapping results for the same region. In both of the figures, the points are colored by height. While the lower figure correctly shows stable coloring for the central portions of the bridge and a slight decrease in elevation at the end of the bridge, the upper figure shows unstable color variations due to inaccurate elevation estimates.

From the time-series data shown in Figures 5-5 and 5-3 as well as the color-shifting in Figure 5-7, elevation estimates placing high confidence in GPS derived elevations would lead to poor results. The sensor noise for GPS elevation measurements is highly non-Gaussian.

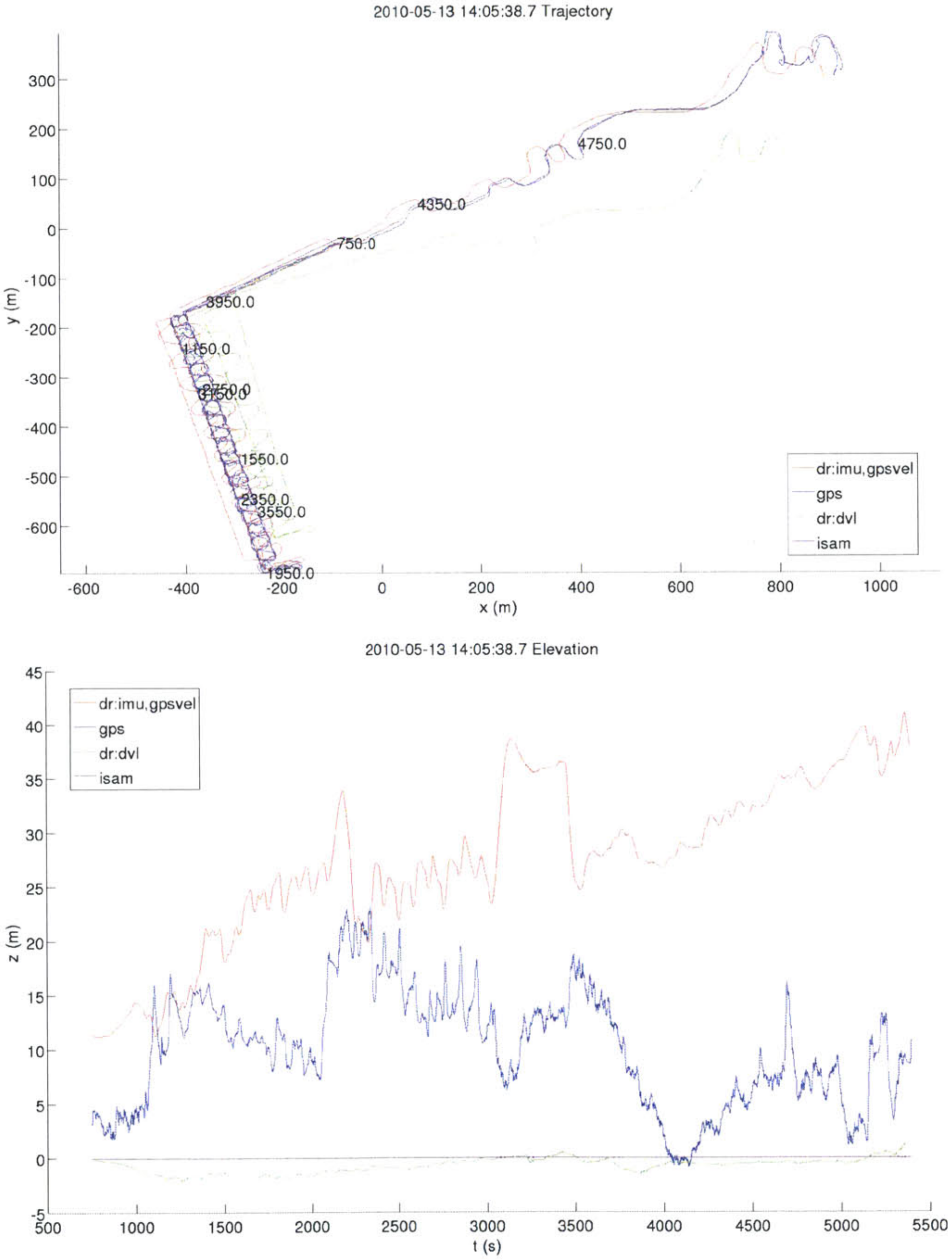


Figure 5-3: The optimized trajectory (dark purple) for a mission is shown compared to trajectories computed by alternative methods in the (x, y) plane. Elevation is shown in the lower figure. Methods based on dead-reckoning (DVL in green and IMU in red) are prone to accumulate error over time. GPS positioning (blue) maintains moderate global accuracy with local discontinuities. Our method (dark purple) combines the benefits of local and global consistency.

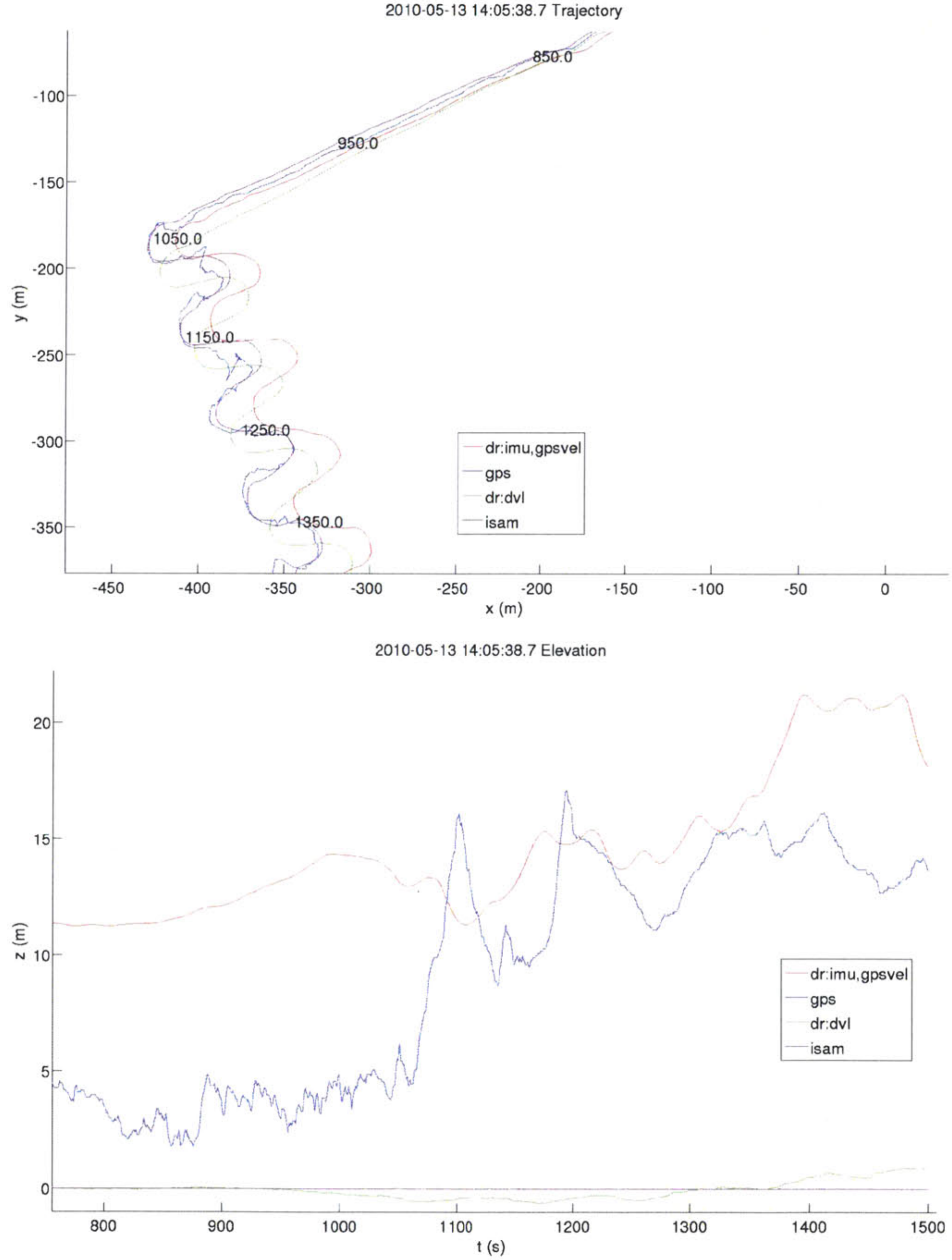


Figure 5-4: The optimized trajectory (dark purple) for a mission is shown compared to trajectories computed by alternative methods. This figure illustrates a zoomed in view of Figure 5-3 for clarity.

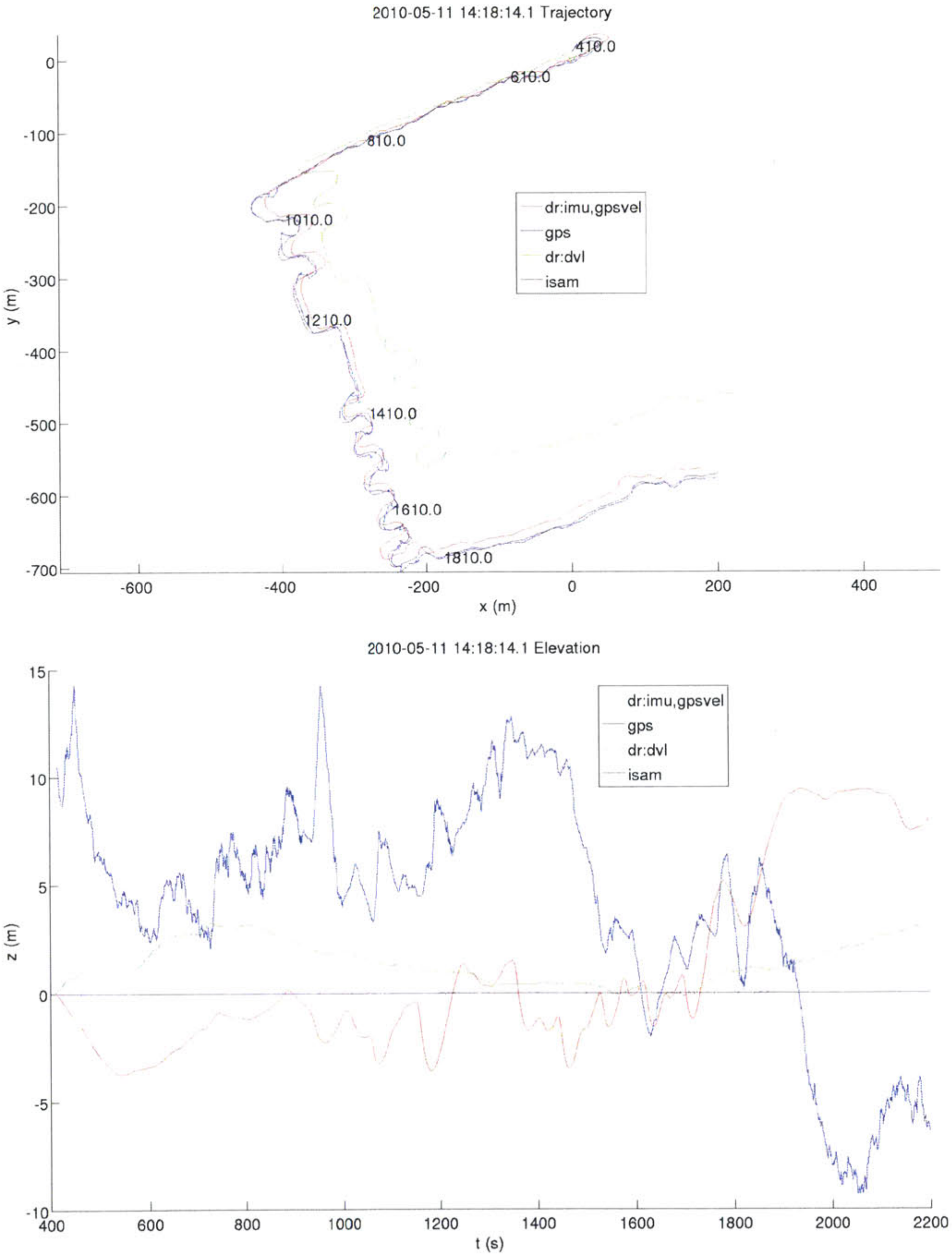


Figure 5-5: The optimized trajectory (dark purple) from a May 11 data set is shown compared to trajectories computed by alternative methods. Elevation is shown in the lower figure.

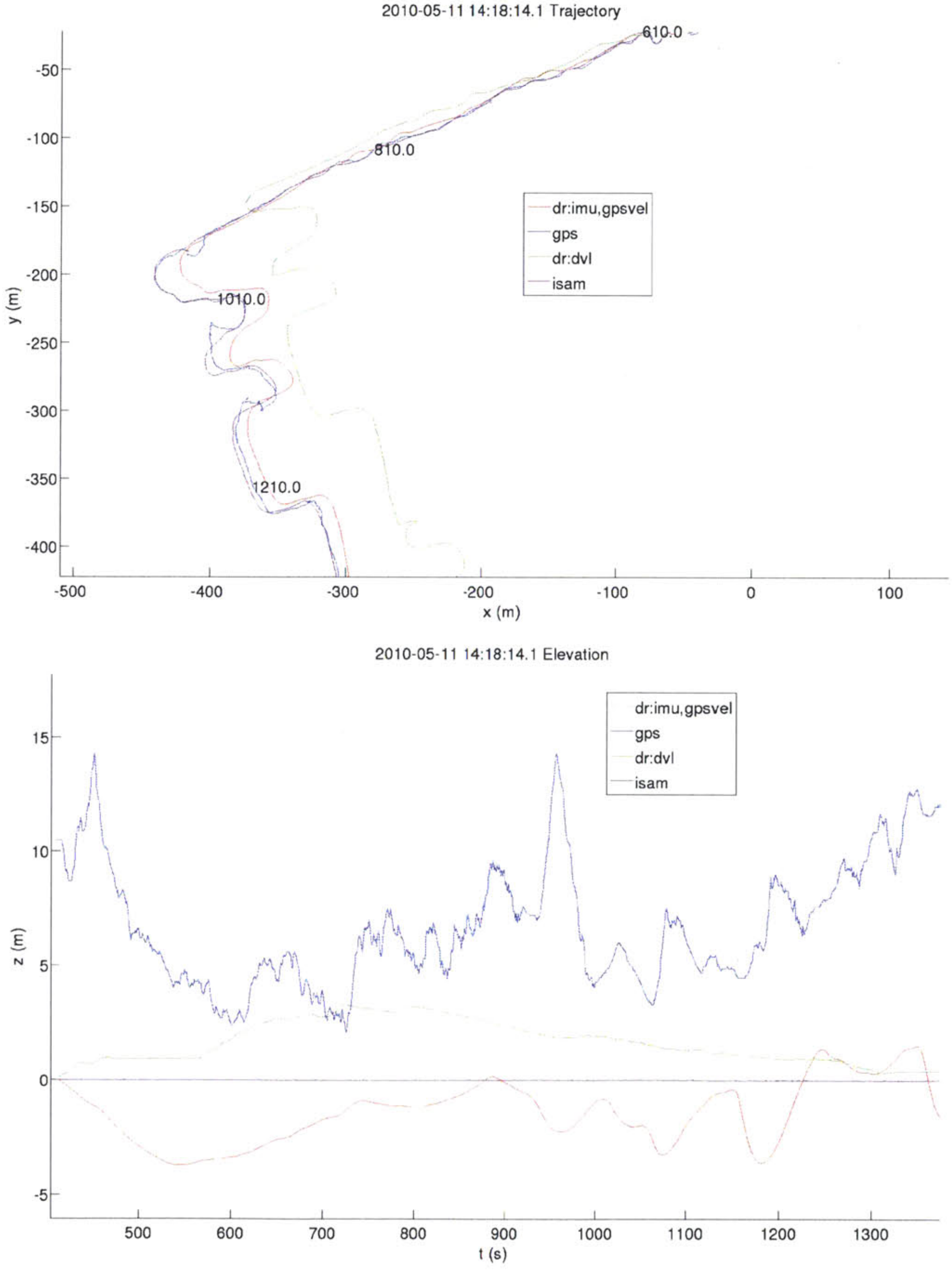
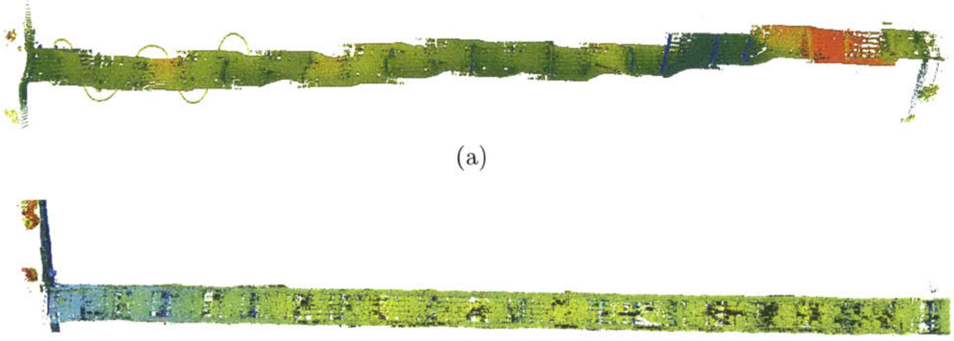


Figure 5-6: The optimized trajectory (dark purple) from a May 11 data set is shown compared to trajectories computed by alternative methods. This figure illustrates a zoomed in view of Figure 5-5 for clarity.



(a)

(b)

Figure 5-7: Localization errors become more apparent in the projected sensor data. In the top figure, range data is projected using the IMU and GPS-velocity based trajectory. In the lower figure, a portion of the final map for the same area is shown. Both images are false-colored by height, and the color shifts in the top image reveal elevation drift.

5.3 Submaps

The experimental data led to 98 individual maps. Of the 4753 possible pair-wise registrations, only 1155 had overlapping regions and 341 registrations were successful. Figure 5-8 shows the approximate regions of the submaps as well as edges from successful registrations. Figure 5-9 shows the resulting adjacency matrix. The submaps having the most edges were large portions of a bridge when the vehicle traveled parallel to the structure.

With the final positions of the submaps obtained in the graph alignment optimization, the global map is reconstructed from the submaps. The bathymetry map is reconstructed using the method described in Section 4.4.2. The resulting maps over the operating area are shown in Figure 5-10. Local consistency is difficult to see at the global scale, and we select several interesting regions to view in greater detail.

Figure 5-11 shows mapping results near the Harvard Bridge at two different scales. Figure 5-11(b) shows the global map reconstruction for more than half (approx. 500m)

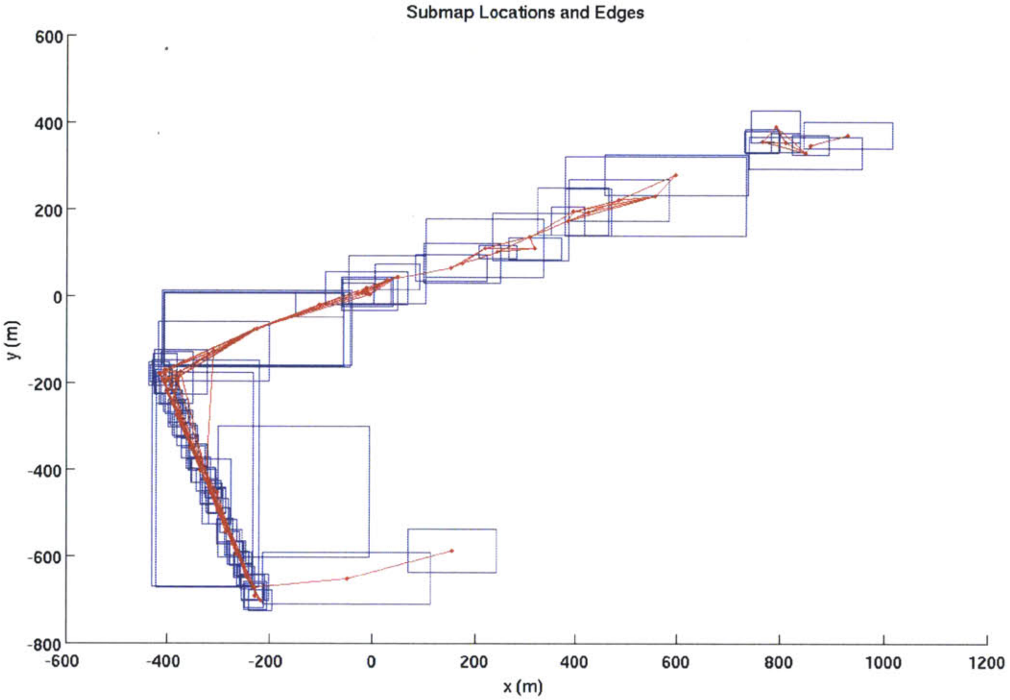


Figure 5-8: Bounding areas of submaps are shown in blue. Red lines represent successful registrations between maps.

of the bridge and the adjacent retaining wall. The Harvard Bridge area encompasses more than half of the submaps due to the repeated session surveying the area and the large number of traversals under the bridge leading to submap segmentation. The large number of maps presented a large amount of sensor coverage as well as a challenge for global consistency. Figure 5-11(c) provides a closer view underneath the bridge to illustrate the level of detail not readily seen in the large scale views. Another view of only surface maps in this region is shown in Figure 5.3. This view also renders the prior model of the stone bridge piers for a reference, and details of the prior model are discussed in Section 5.4.

Another notable structure of interest in our results is the Longfellow Bridge, approximately 1500m downstream from the Harvard Bridge. Unlike the Harvard Bridge,

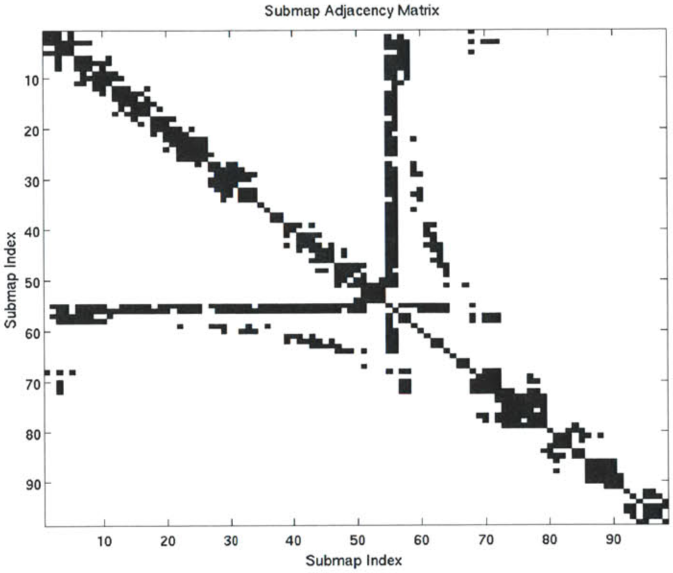


Figure 5-9: Adjacency matrix for the graph of submaps. Dark regions indicate an edge. Submaps with the highest connectivity have many off-diagonal adjacencies and correspond to trajec

our surveys did not provide near complete coverage of the structure, yet sufficient coverage was achieved to produce the results shown in Figure 5-13. Both surface and bathymetry maps are shown with independent false coloring by elevation. A camera image of the bridge is shown above the mapping results for clarify the structure for the reader.

Figure 5-14 shows mapping results near the MIT Sailing Pavilion. The pavilion area is unlike other regions in the operating environment with regard to dynamics. This figure shows surface mapping results from a single day (two submaps) along with the global bathymetry map. Later in Section 5.6 we discuss the effects of dynamics in this region upon the maps along with views of multiple maps for this region.

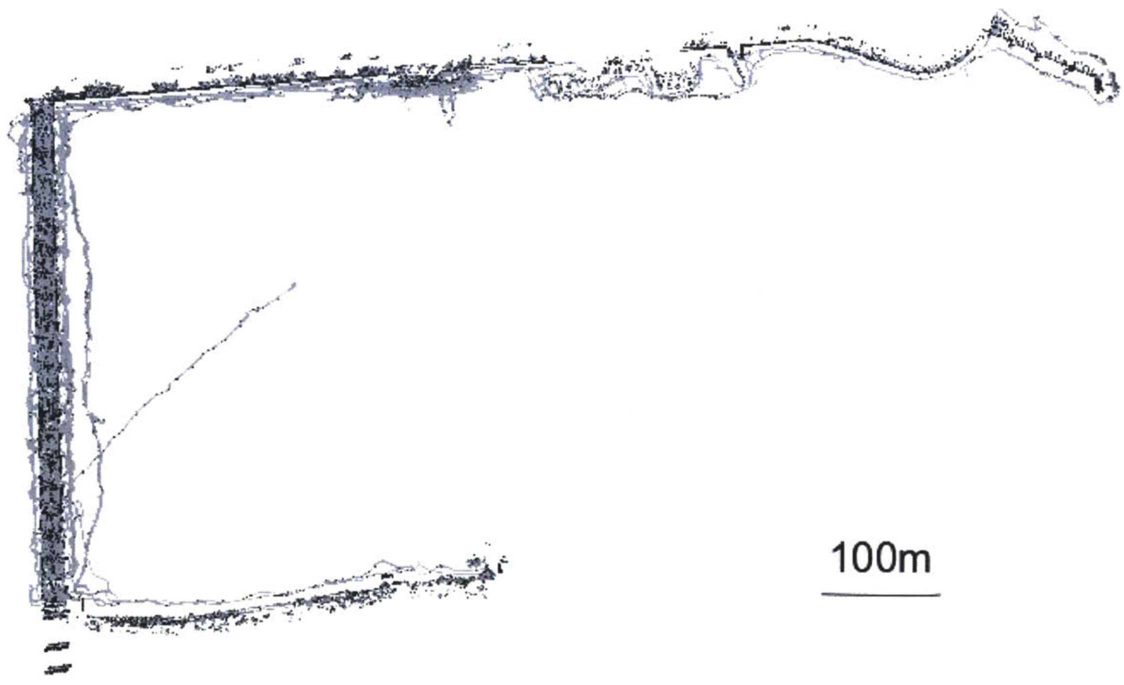


Figure 5-10: The results of global map alignment produce a consistent reconstruction of the environment.

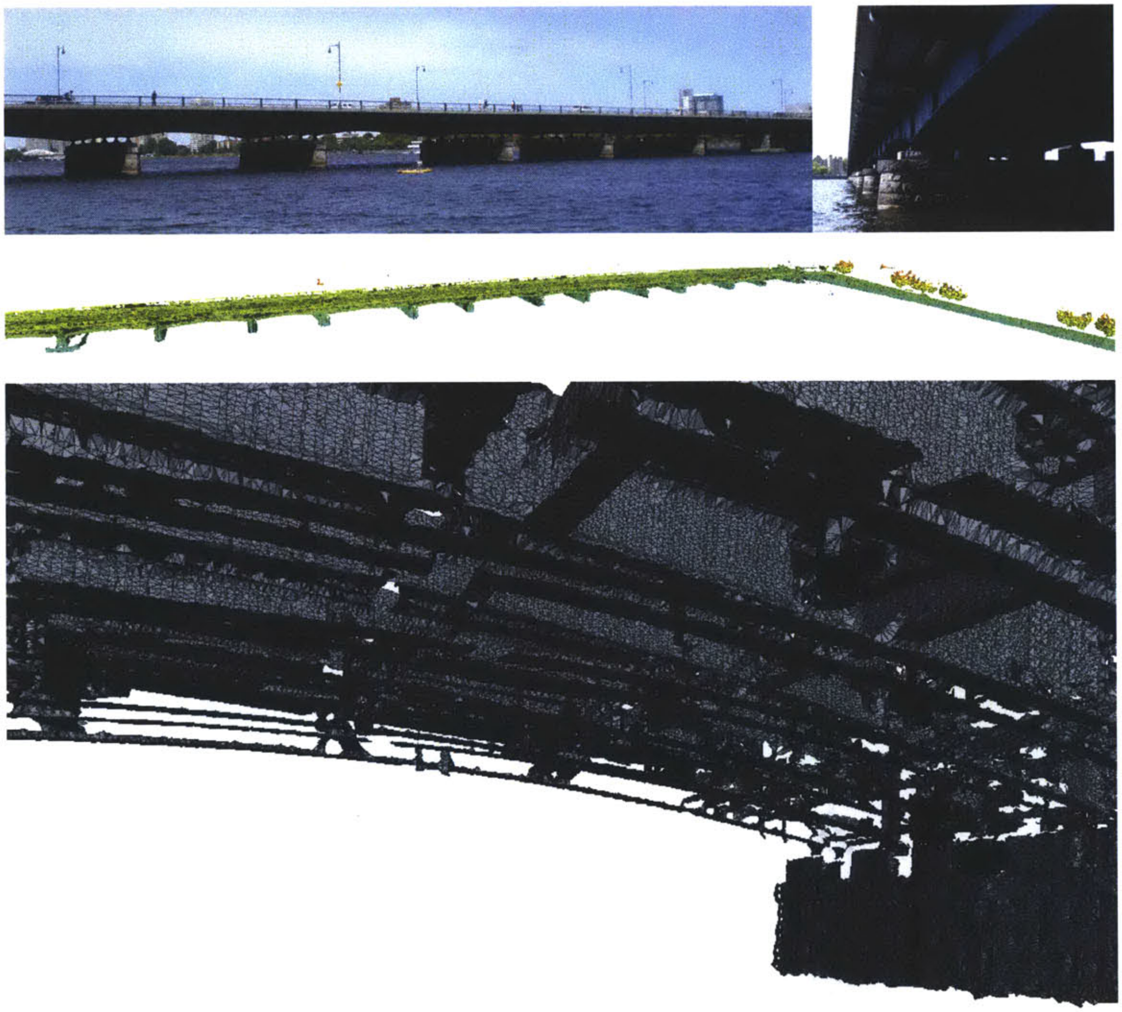


Figure 5-11: Example views of the estimated surface maps to demonstrate consistency on multiple scales. The images at top are shown for qualitative comparison. The middle figure shows maps estimated near the Harvard Bridge and the adjacent retaining wall with false coloring by elevation. The lower figure provides a close view of the mesh surface underneath the bridge between two piers.

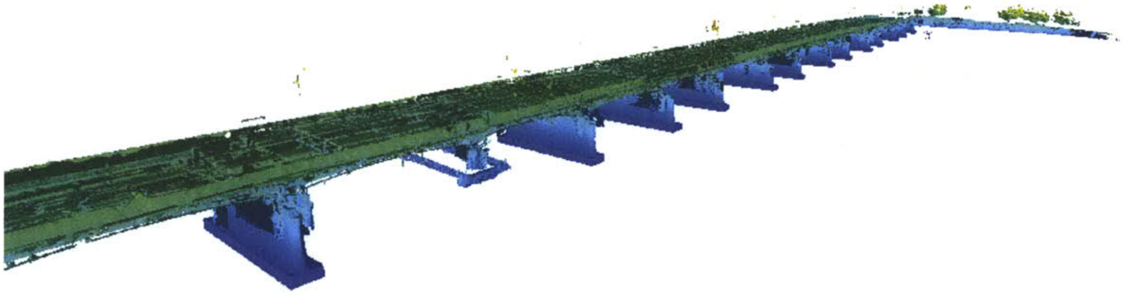


Figure 5-12: A view of the surface maps is shown along the Harvard Bridge. The maps are false colored according to height. The prior model of stone piers are shown as a reference.

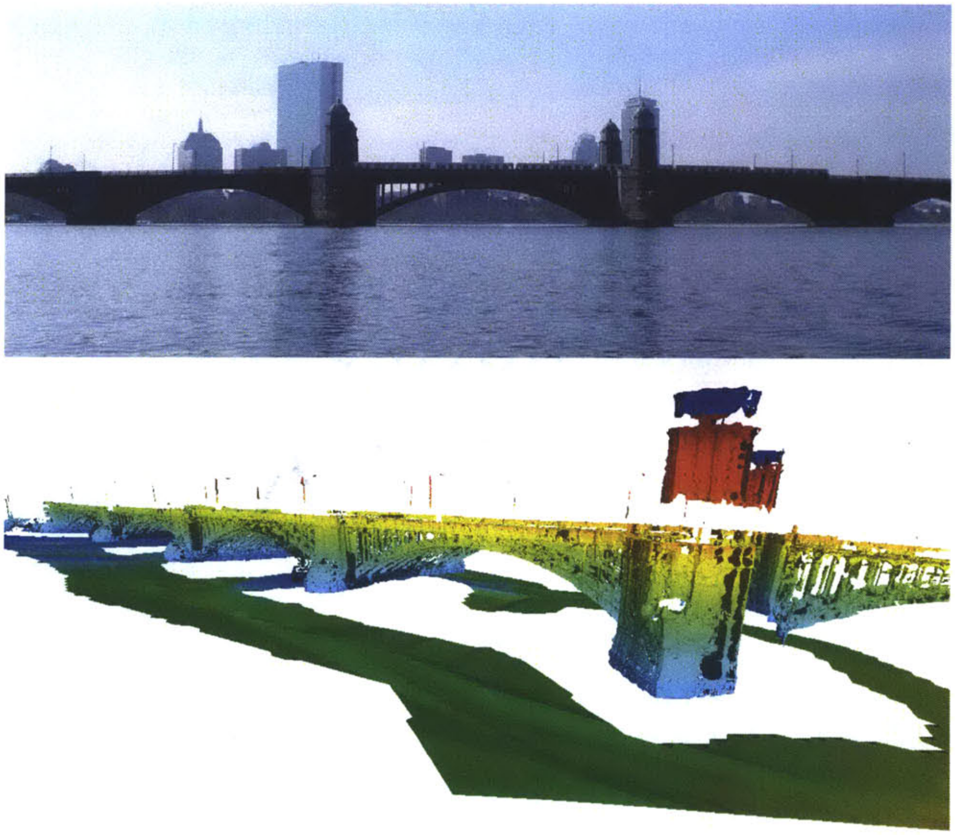


Figure 5-13: Mesh reconstruction and bathymetry results are shown along the Longfellow Bridge. The stone pillars were partially occluded by their larger bases, preventing observation of the complete stone structure from the water level.

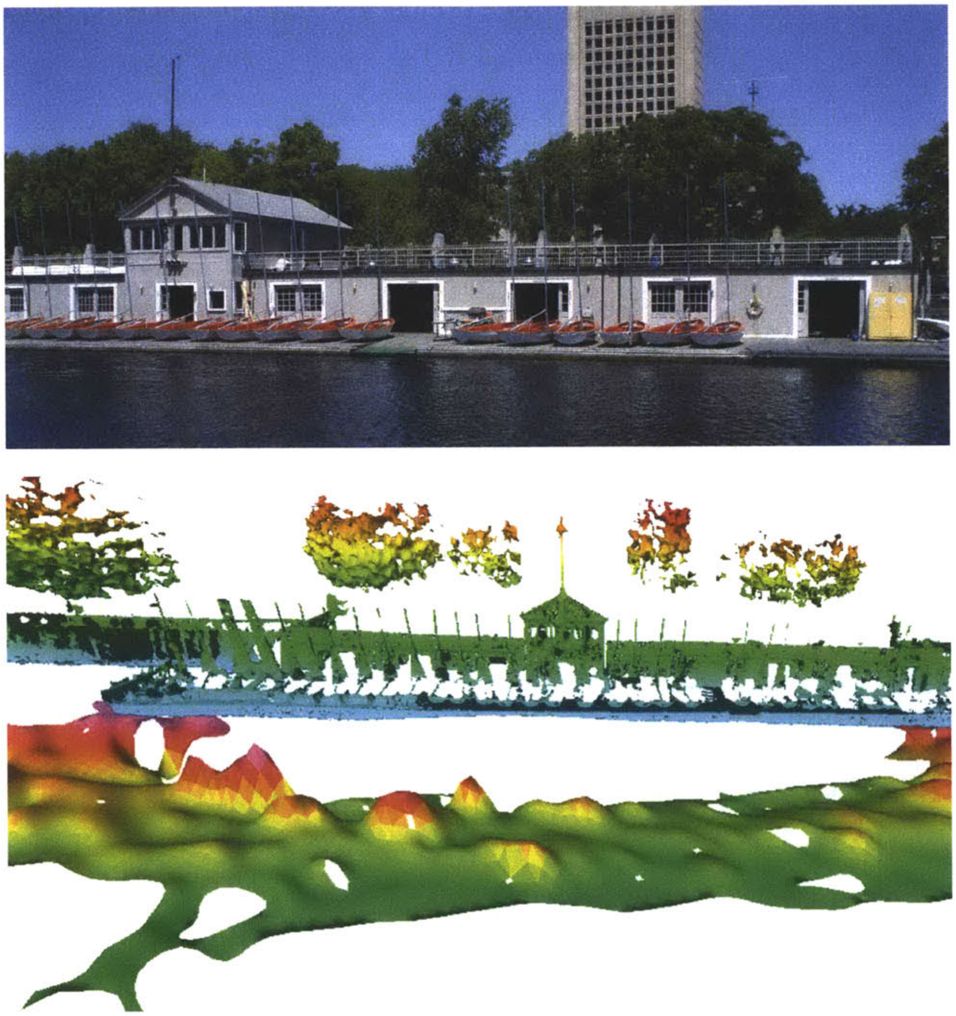


Figure 5-14: An image of the MIT Sailing Pavilion is shown above, and several surface maps and global bathymetry are shown below. The sharp peak in bathymetry at left is one of the supporting pylons of the pavilion.

5.4 Bridge Model Validation

Although ground truth data is unavailable to assess map error for the operating area, a reference model of the Harvard Bridge was generated from the blueprints archived by the Library of Congress [116]. A three dimensional model of the bridge's stone piers was created and georeferenced as shown in Figure 5-15. Further details concerning the model creation are provided in Appendix A. Using dense Monte Carlo surface sampling, a point-based map of a million samples was generated as the reference map as shown in Figure 5-16. Submaps containing portions of the bridge were matched against the model using the registration method detailed in Section 4.3.2.

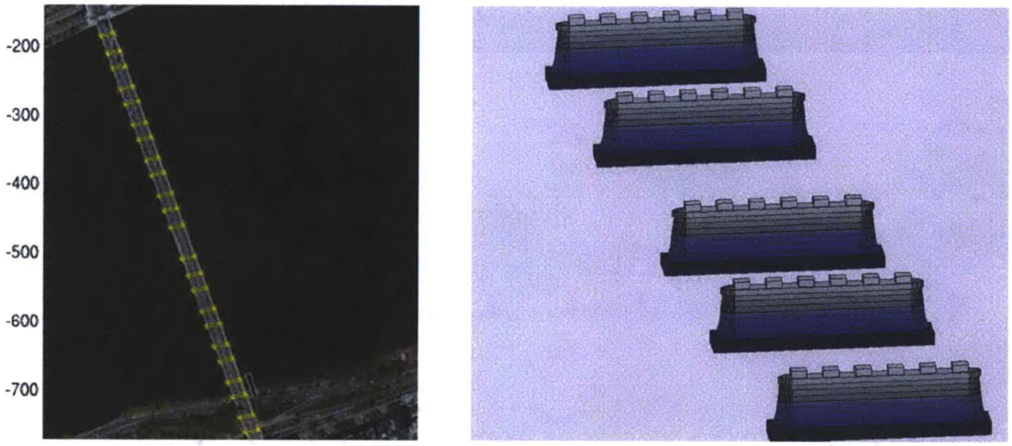


Figure 5-15: Model of Harvard Bridge from Library of Congress [116] — used for validating map quality.

Our first method of validation with the prior model uses relative spatial relationships between features of both the model and maps. Comparison of relative distance within the map provide a measure of local consistency, and sequential pier distance is a quantity independent of the global reference frame. Error arising from the georeferenced position of the model does not affect the sequential pier distance. Relative distances between sequential piers is calculated for the aggregate of submaps and compared to those provided in the model. Figure 5-17 shows map data and samples of the model data for a small elevation interval. The coordinate frame is rotated to show the bridge horizontally. The map data appears generally consistent with the model data, and we wish to quantify the consistency. Using the map data shown in Figure 5-17, we calculate the distance between sequential piers and compute the error relative to what the model expects. The distribution of error is shown in Figure 5-18, and the mean error is 0.11 meters.

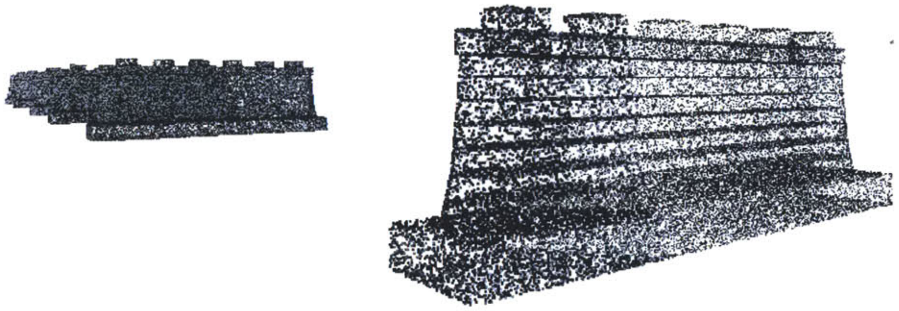


Figure 5-16: Dense sampling of the prior model is used for registering maps permits the use of the same registration method for submap alignment in validation.

We also wish to investigate the absolute accuracy of the final maps by registering the maps to the georeferenced model. The procedure to register submaps to the model uses the same registration method from Chapter 4 for matching submaps with minor modifications to the algorithm. The major difference is in cropping the inputs to a coarse Region Of Interest (ROI) to prevent incorrect solutions and reduce runtimes. The ROI was calculated using the intersection of dilated bounding volumes for the submap and model, and then clipped in the vertical direction. The vertical clipping removes much of the bridge superstructure seen in the submaps but not found in the model. The parameter value was set to ignore map points at elevations exceeding the model's highest point by more than 0.40 meters.

Registration to the model was generally successful and low in error. Figure 5-19

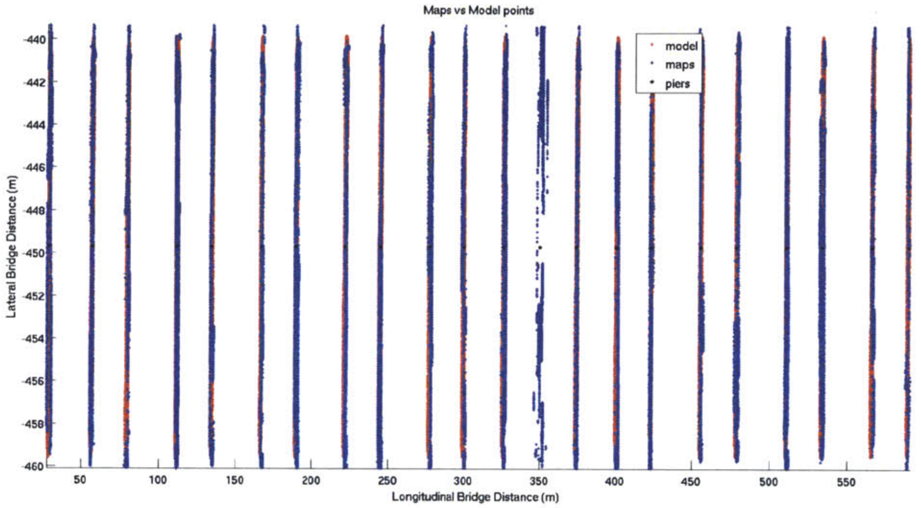


Figure 5-17: Map voxels (blue) are shown with samples of the bridge pier models for a small elevation range. The coordinate frame is rotated to place the longitudinal axis of the bridge on the x axis.

shows a histogram of the mean squared error (MSE), and the average MSE was 0.074 square meters. Over half of the registrations had MSE less than 5 square centimeters since the median MSE was only 0.04 square meters. Over 75% had MSE less than 10 square centimeters.

Registration failed with submaps in two regions. One region is adjacent to a retaining wall which is not in the model. The second region is near an unmodeled pier which the blueprints did not contain due to the bridge originally having a drawbridge in this region.

Figure 5-20(b) shows histograms of the number of points matched from each submap to the model as well as the ratio of the match count to the total points available for matching. The points available for matching are less than the size of the submap due to cropping. The majority of maps were able to match nearly all possible points.

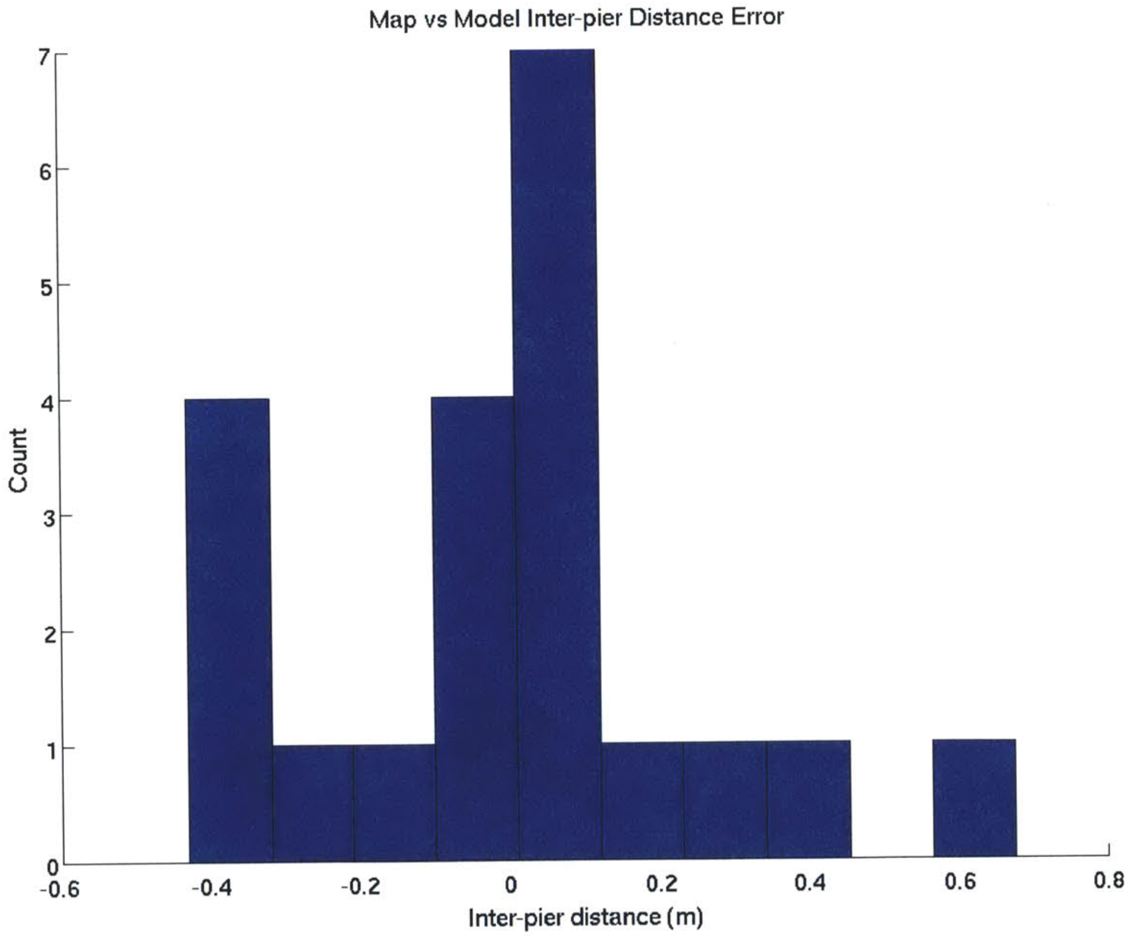


Figure 5-18: The distance between sequential piers from the map data is compared to the model distances in this error histogram. A highly peaked histogram centered at zero error indicates high quality. The mean and standard deviation of the error were 0.11 and 0.056 meters, respectively.

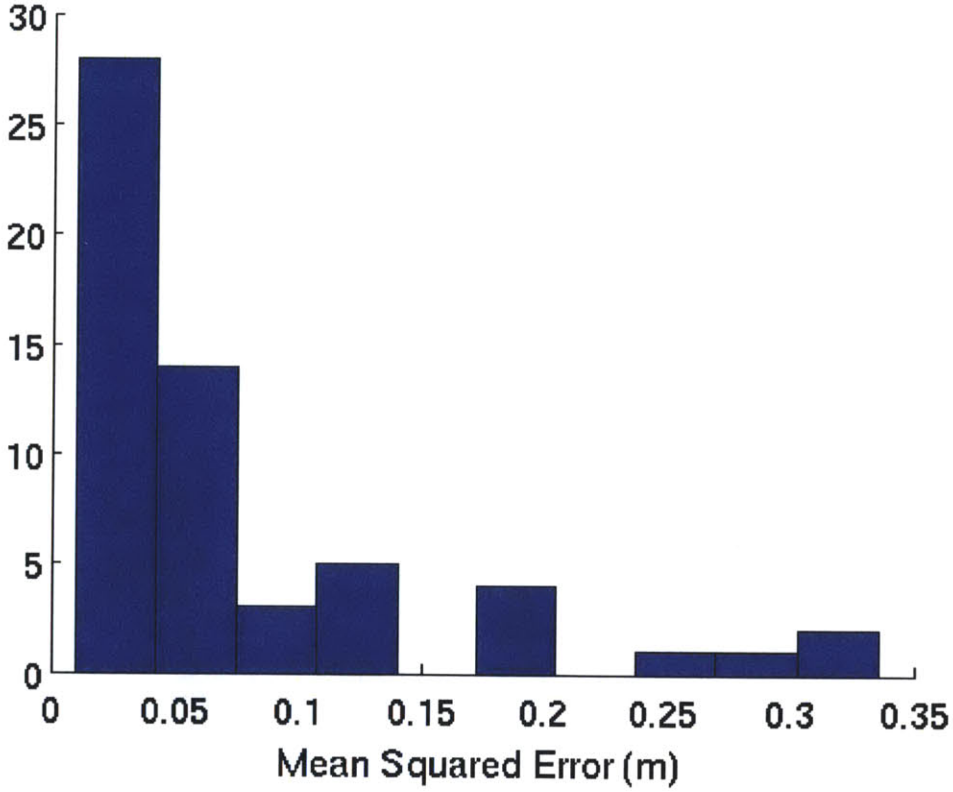
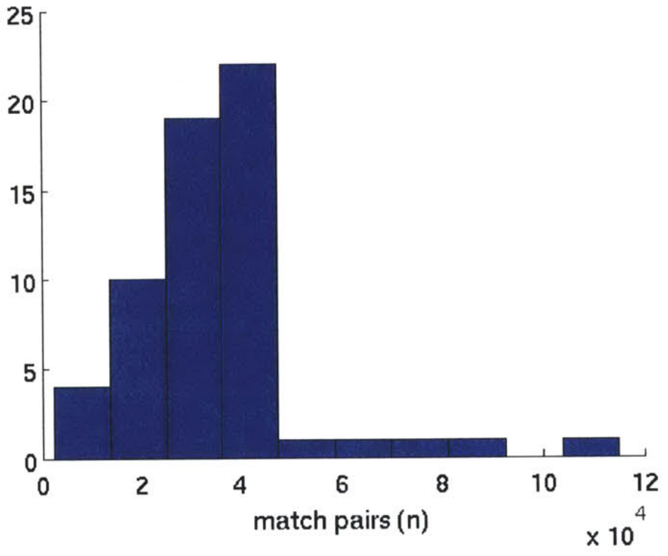
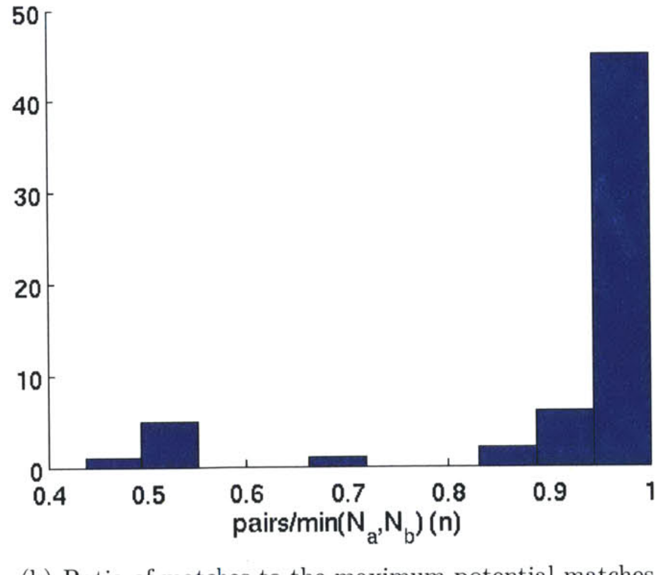


Figure 5-19: Submaps are validated against the prior model of the Harvard Bridge with an iterative closest point (ICP) registration. The distribution of mean squared error (MSE) is shown in the histogram. More than 75% of the maps registered with less than 10 cm^2 mean squared error.



(a) Number of matched points



(b) Ratio of matches to the maximum potential matches

Figure 5-20: A histogram of the number matched points in each submap to the bridge model is shown in the upper image. Below, a histogram of the ratio of the matching pairs to the maximum potential matches (size of the smaller map) is shown. Many of the submaps matched all pairs

Successful registrations to the model resulted in transformations of small effect. Absolute rotational components were all less than 5 degrees, and translation components less than 2.5 meters. Nearly all transformations had a negative z component with mean value of 0.23 meters. The common z displacement is likely due to error in referencing the model. Accuracy was greater for georeferencing the model in the (x, y) plane due to the availability of high resolution aerial ortho-imagery.

In Figure 5-21 several histograms show the distribution of components of the transforms. Significant non-zero mean values, or a *skew* in the histogram, could indicate georeferencing error in the prior model or a sensor bias due to the georeferencing imagery using a different sensor than the platform in this research.

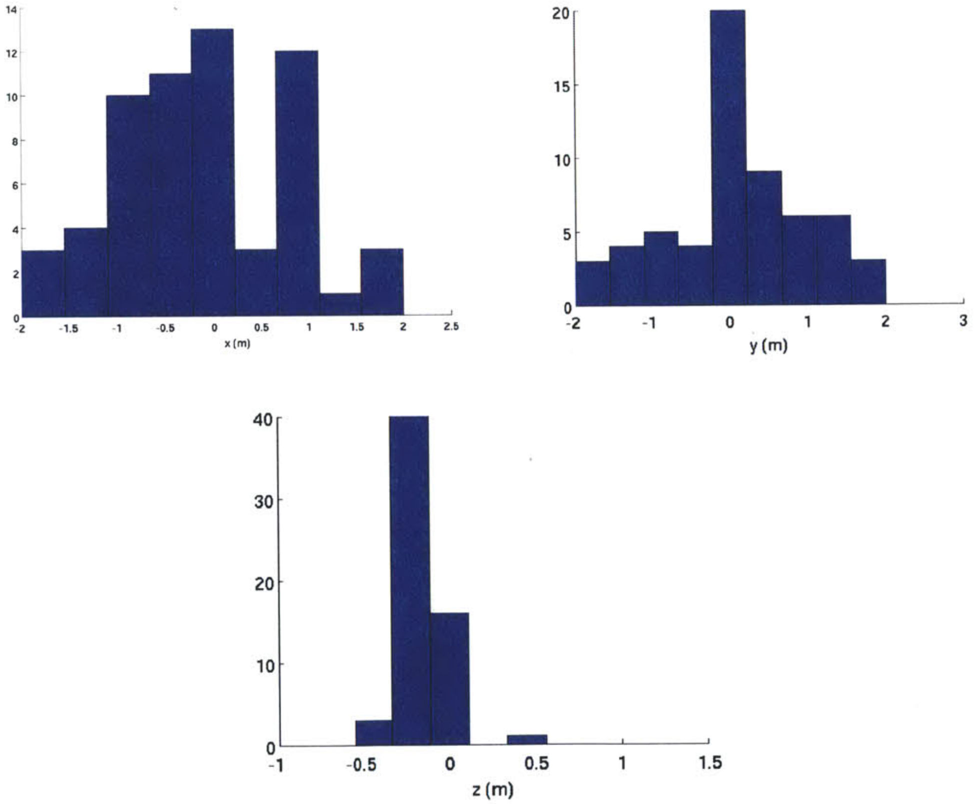


Figure 5-21: Registration of the submaps to the prior model had transformations with (x, y, z) components distributed as shown.

5.5 Bathymetry

The bathymetry recorded during experiments represents sparse coverage compared to the total river basin area within the operating area. The primary reasons for this outcome are the limited range and field of view of the sonar sensor and lack of extended missions into the central portion of the river. Figures 5-22 and 5-23 show the bathymetry map.

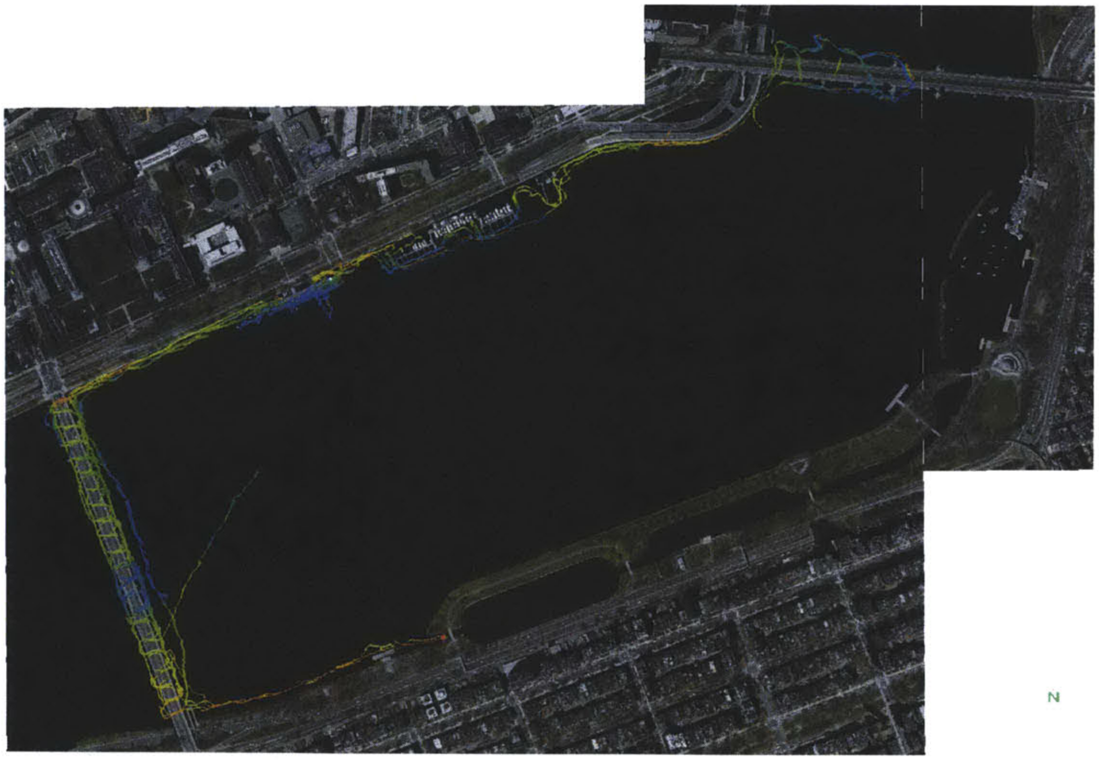


Figure 5-22: Bathymetry map created from fusing surface craft bathymetry submaps is overlaid on aerial imagery. Map is false colored with cooler colors (blue) deeper and warm colors (red) shallower. The limited sensor field of view and scale of the environment present significant challenges to achieving sensor coverage.

We augment the bathymetry map from the surface craft with bathymetric data gathered in nine earlier experiments using a REMUS AUV [3, 74]. Figure 5-24 shows

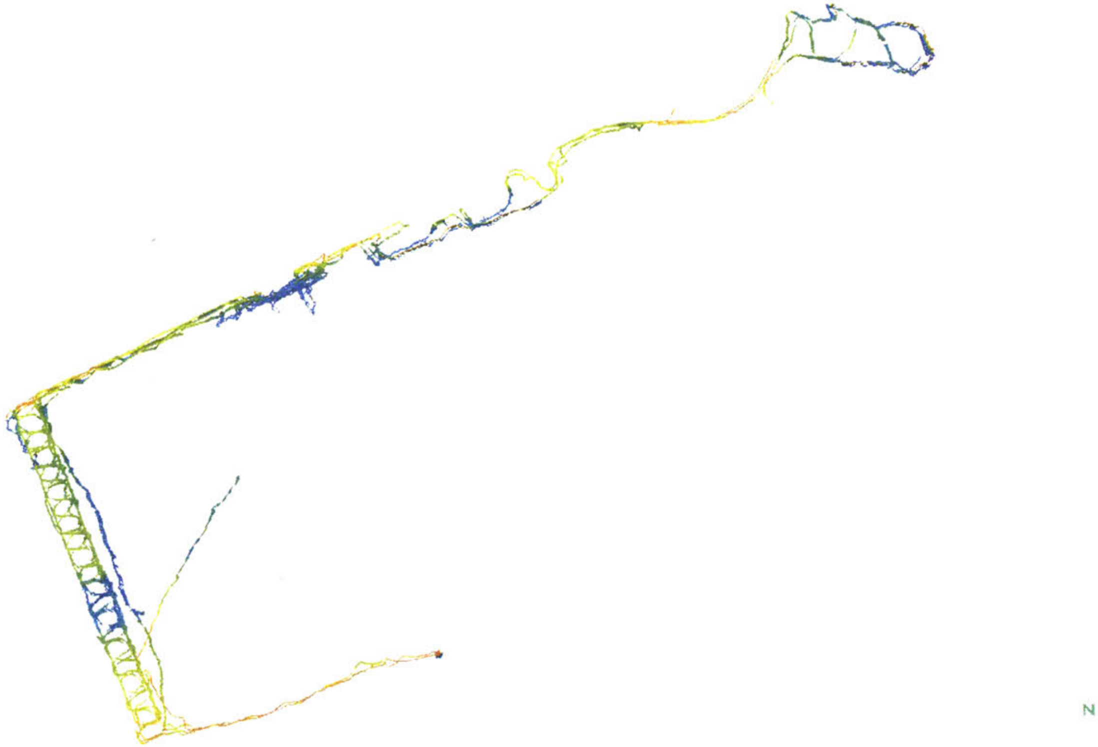


Figure 5-23: Bathymetry map created from fusing surface craft bathymetry submaps. Map is false colored with cooler colors deeper and warm colors (red) shallower.

the vehicle as it was deployed. Due to the risk of vehicle damage or loss by navigating near structures such as a bridge, most experiments were confined to the central portion of the river. Long Base-Line (LBL) acoustic ranging augmented the localization accuracy of the vehicle while submerged, but is not reliable near structures. In the final three missions, an expensive and highly accurate inertial navigation system ensured accurate navigation. Despite the risk of traversing near the bridge, one mission attempted to navigate through the center of one of the larger spans of the Harvard Bridge using georeferenced locations of the bridge piers. While the vehicle made a successful traversal to the west side of the bridge, the vehicle augered into the mud upon the return attempt. The vehicle encountered a slope too steep for the vehicle's altitude control to avoid with the increased length of the vehicle due to the INS module. The vehicle was eventually freed, and with navigation corrupted, collided with a stone retaining wall. Operators triangulated the approximate wreck using an acoustic modem for recovery[1]. Overall, the AUV proved very suitable for open water conditions, and the surface craft provides observational capabilities for safe navigation in the localities hazardous to AUV operations.

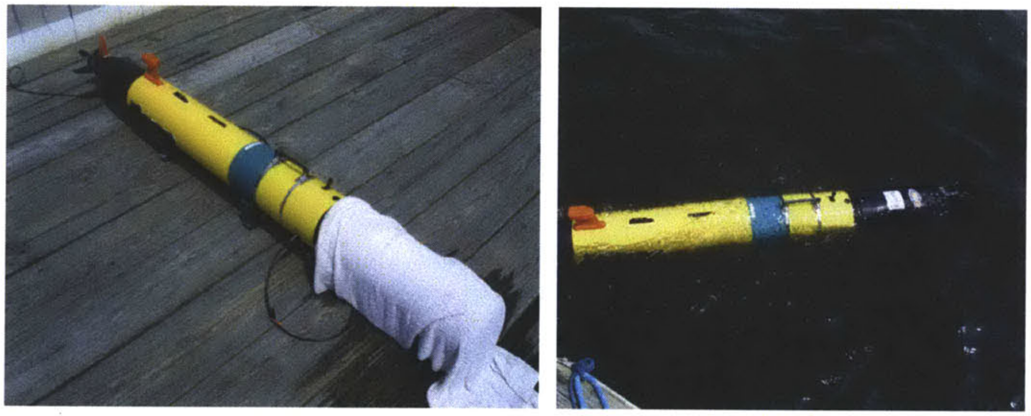


Figure 5-24: The REMUS AUV shown provided bathymetry data for the central region of the Charles River. The green segment of the vehicle body is a high accuracy inertial navigation system INS, and the black segment at the front is a forward-looking sonar.

Results fusing the surface craft bathymetry with AUV bathymetry are shown in Figure 5-25. An artifact of the bathymetry fusion is a resolution loss due to the resolution of the AUV bathymetry. Readers may notice the dilation of regions traversed by the surface vehicle only.

Absolute accuracy of the bathymetric maps is difficult to determine. With no ground truth available, there is no reference for a comparison. Assuming the depth estimates to be accurate, the accuracy in the *positioning* of depth estimates is linked to the accuracy of the surface maps.

[1] The author recovered the vehicle by swimming under a dock where the vehicle surfaced and was trapped by pylons.

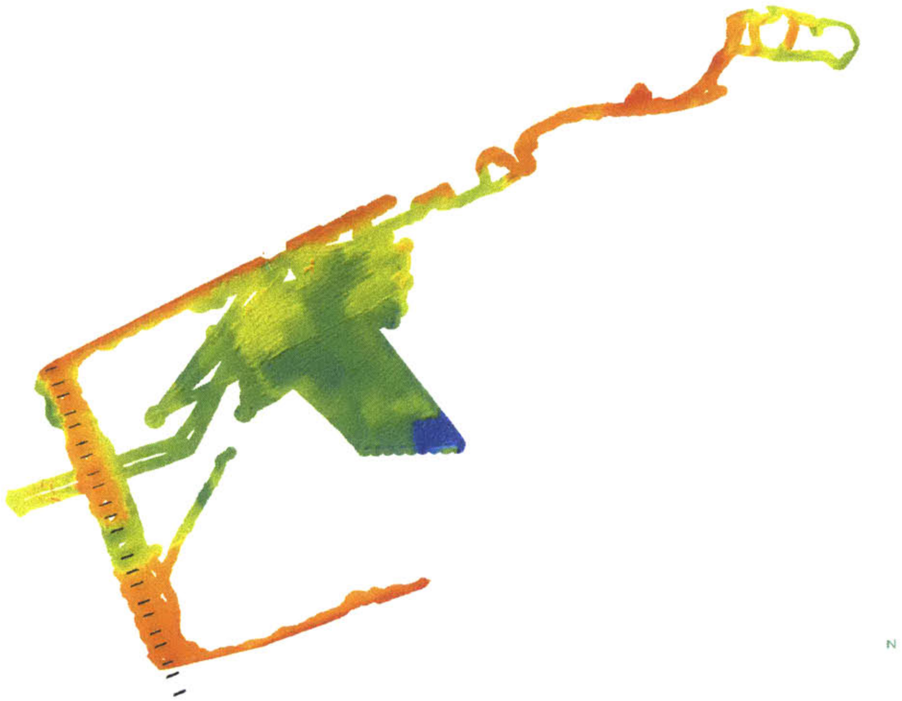


Figure 5-25: Bathymetry map fusing surface craft and submersible vehicle data. Map is false colored with cooler colors deeper and warm colors (red) shallower. Note the color mapping is different than the kayak only bathymetry to provide better contrast for the greater range of depth.

5.6 Discussion

Applying the methods described in previous chapters to actual data in a marine environment. Upon close inspection, the final maps contain subtle errors. For example, a region near a bridge showed some aliasing as shown in Figure 5-26. In this example, the misalignment is noticeable with several of the street lamps and for the concrete supports under the street lamp shown.

A major assumption of our method was semi-static environment, and dynamic environments violating this assumption were evident in the final data. Objects with high-dynamics, such as people on sidewalks or passing trucks, were not present long

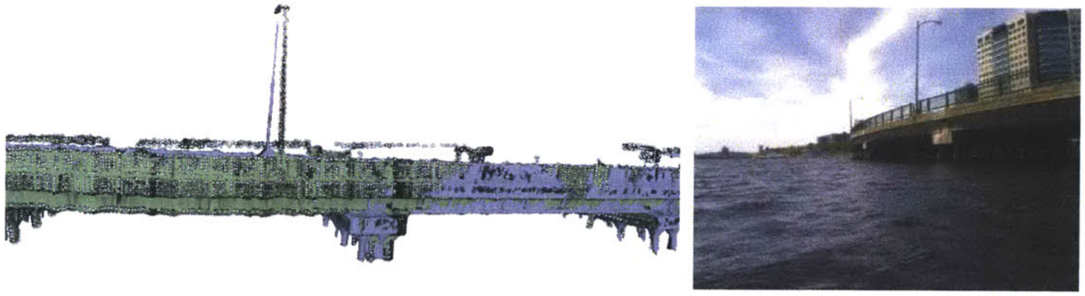
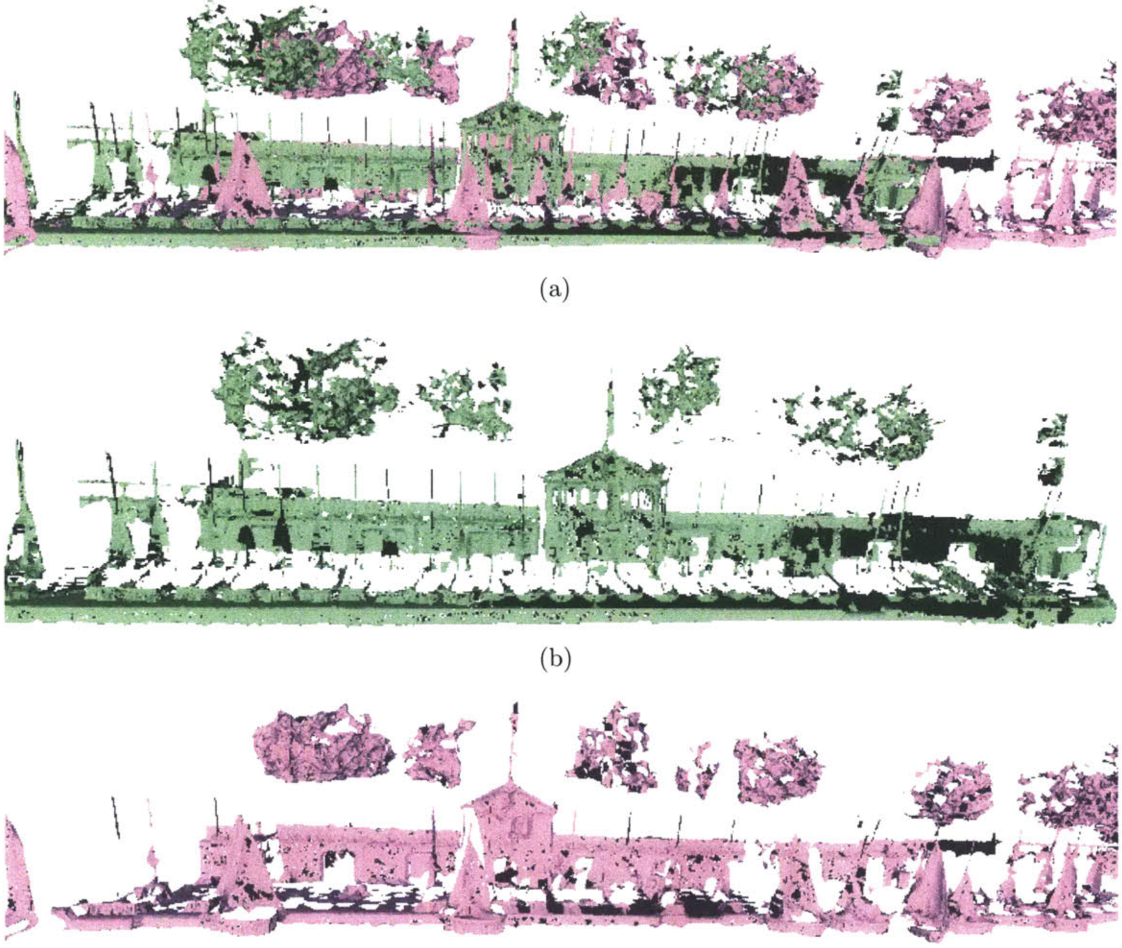


Figure 5-26: A mapping defect is apparent from aliasing of a street lamp and the concrete supports below. A camera image shows a similar portion of the structure for comparison.

enough to exceed the evidence threshold. Objects with lower dynamics led to some extra noise. Objects that moved or changed between days lead to very clear discrepancies in the maps. For example, the maps shown in Figure 5-27 show an area where sailboats have moved between the observation times of the maps. For clarity only two of the submaps are shown in the figure, and other maps near this location showed similar evidence of dynamic conditions. In our application, the maps in the pavilion area had enough static correspondence to achieve adequate pair-wise map registration, although the dynamic objects present visualization artifacts with less appeal. Another (low) dynamic situation is shown in Figure 5-28 where five moored sailboats drifted near their anchors between the days of the missions responsible for those maps. In this case the static retaining wall and static appearance of the trees show agreement between the two maps, as seen in the color "blending" of the top figure, and the central region shows eight boats. Considering the maps individually in the bottom two figures, the correct number of boats is five, and all but the second boat from the left moved significantly between the map observations.

With concern to the bathymetry maps, one observation is the limited coverage of the environment. Autonomy for surveys would be a promising direction to improve the coverage. An alternative approach would be changing the hardware, such as replacing the sensor with one having a greater field of view or changing the mounting configuration. Note the vehicle design already accommodates two sonars. When considering the mounting orientation of a single sensor, there is a tradeoff in perception. When mounting in the broom fashion used in this thesis, the vehicle may observe bathymetry, approach of river banks or obstacles, and some structures, but the structures would need to be very close. Achieving close inspection remotely is difficult, and autonomy is a promising solution. An alternative side-scan configuration similar to the vertical laser sensors would provide less observability of bathymetry, but provide easier observability of structures below the water as the vehicle travels past. We fused bathymetry data from earlier AUV experiments to provide a more complete bathymetric map. As operations with AUVs are usually in open water and shallow marine environments with obstacles are risky, but our platform is able to map the hazardous regions, AUV and ASC cooperative mapping can be complementary.

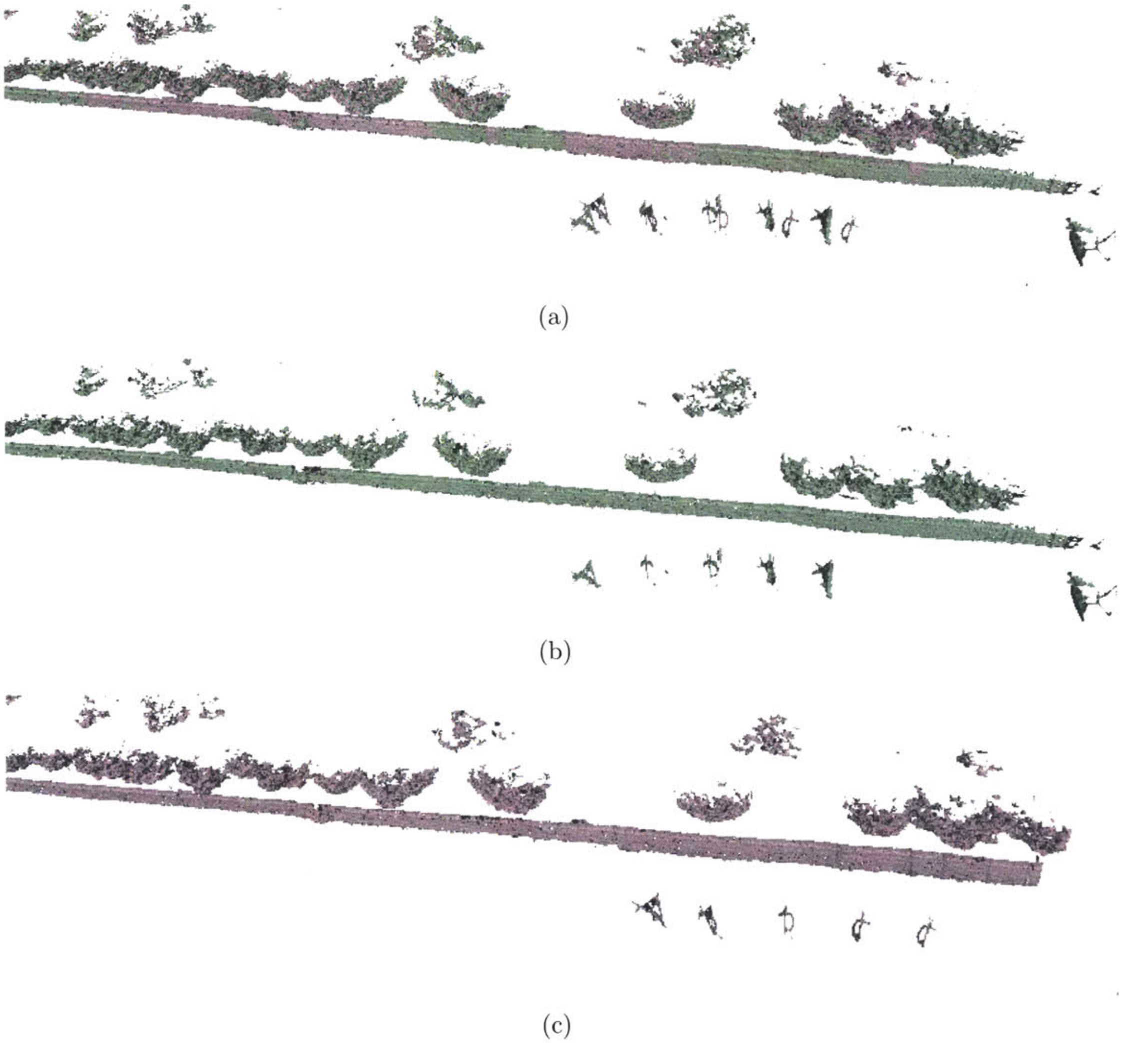


(a)

(b)

(c)

Figure 5-27: Mapping discrepancies arise from dynamic conditions. Two maps, shown in different colors, reveal different configurations of sailboats near a sailing pavilion. Both maps are shown superimposed in Figure 5-27(a), and then individually in Figures 5-27(b) and 5-27(c).



(a)

(b)

(c)

Figure 5-28: Mapping discrepancies arise from dynamic conditions. Two maps, shown in different colors, reveal different configurations of five moored sailboats near a retaining wall. Both maps are shown superimposed in Figure 5-28(a), and then individually in Figures 5-28(b) and 5-28(c).

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 6

Conclusion

6.1 Summary of Contributions

In this thesis, we have proposed a new approach for mapping shallow water marine environments, combining data from both above and below the water surface in a robust probabilistic state estimation framework. Whereas most recent AUV mapping research has been limited to open waters, far from man-made surface structures, in our work we have focussed on complex shallow water environments, such as rivers and harbors, where man-made structures block GPS signals and pose hazards to navigation. Robust autonomous mapping operations in these areas would be tremendously useful for a variety of applications, including inspection of marine structures and post-disaster search and rescue missions.

Our work makes systems, algorithmic, and experimental contributions in perceptual robotics for the marine environment. From a systems perspective, we have created a sensor-rich Autonomous Surface Vehicle platform and used it to acquire unique data sets for 3D marine mapping algorithm development.

Algorithmically, we have developed new techniques for addressing a challenging

3D SLAM problem. A key challenge to mapping is the extraction of effective constraints from 3D lidar data despite GPS loss and reacquisition. This was achieved by developing a GPS trust engine that uses a semi-supervised learning classifier to ascertain the validity of GPS information for different segments of the vehicle trajectory. This eliminates the troublesome effects of multipath on the vehicle trajectory estimate, and provides cues for submap decomposition.

Localization from lidar point clouds is performed using octrees combined with Iterative Closest Point (ICP) matching, which provides constraints between submaps both within and across different mapping sessions. Submap positions are optimized via least squares optimization of the graph of constraints, to achieve global alignment. The global vehicle trajectory is used for subsea sonar bathymetric map generation and for mesh reconstruction from lidar data for 3D visualization of above water structures.

Our experimental results focus on several structures spanning or along the Charles River between Boston and Cambridge, MA. The Harvard and Longfellow Bridges, three sailing pavilions and a yacht club provide structures of interest, having both extensive superstructure and subsurface foundations. To quantitatively assess the mapping error, we have compared our SLAM generated maps against a georeferenced model of the Harvard Bridge using blueprints from the Library of Congress. Our results demonstrate the potential of this new approach to achieve robust and efficient model capture for complex shallow-water marine environments.

6.2 Future Research

A number of important research areas stand out for future research investigation in this area. We first discussion possible research directions for improving perception, localization, and mapping. Subsequently, we discuss the potential for integrating real-time trajectory planning with the hierarchical 3D mapping system to achieve real-time obstacle avoidance and path planning.

6.2.1 Localization and Mapping

A future extension to the trajectory and map estimation would be to determining good features to use to improve localization. Even using features unstable for multiple trajectories, the localization quality might be improved. Additional sensors could also provide observability of landmarks that are repeatable for short durations and are sufficient to constrain the vehicle pose.

Research in visual SLAM and reconstruction has made significant progress in recent years. Vision in large marine environments faces many challenges. Lighting changes from shadows of structures or time of day may be minor challenges. Visual odometry using descriptors such as SIFT may provide erroneous features corrupting relative motion estimates. Complications may arise due to surface chop, especially high-contrast whitecaps reflections of sunlight. A minor but not trivial issue is maintaining a clear lens, since wind and splash can deposit water on the sensor, leading to distortions, and a few examples are shown in Figure 6-1. Due to the large scale of the environment and large expanses of open water, stereo vision would encounter difficulties in depth estimation due to distances to objects relative to the distance between lenses, but may be suitable when sensors are relatively close to structures.

In this thesis, a classifier was trained using semi-supervised learning methods to recognize poor GPS conditions in order to avoid certain GPS hazards. The provision of additional labeled training data could reduce classification risk for future samples without modification to the classifier model. The search for an improved feature space for the classifier remains an open problem. Generally speaking, the incorporation of semi-supervised learning methods to improve robustness and prevent data association corruption with persistent robotic SLAM remains an promising avenue for future research. Common corruptions and divergence in SLAM applications may be attributed to inadequacy of an underlying model and/or assumption violations, and learning techniques can provide robustness.

Another direction for improvement is extension of models to capture more of the true complexity of the problem. For example, biases due to a different view of the GPS satellite constellation or atmospheric effects could be explicitly modeled to provide improved localization and greater consistency of localization of missions taken on different days or different times of a day. Such extended modeling may not be possible with all GPS sensors, and existence of alternative means to robust estimation is lacking. An example of recent work incorporating extended models for GPS in localization has been demonstrated in [30]. Future research may also need to address signal interference hazards, as widespread construction of broadband network towers have recently been postulated as a hazard to GPS reliability in the United States.

While the implementation presented here incorporates submapping for scalability, localization at the mission level could be further improved with an out-of-core hierarchical method. One intention of the submap approach presented here was to divide the massive quantity of sensor data, but one unanticipated outcome was the slow buildup of pose states in localization. Despite using state-of-the art incremental optimization, in-core memory usage steadily climbed and incremental solutions almost approached sub real-time performance for the longest missions at the highest sampling frequency. Use of a lower sampling frequency (50Hz rather than 100Hz) maintained better than real-time performance without discernible differences in accuracy. To manage longer sessions, consideration must be taken toward techniques using out of core methods for the sparse matrix optimizations. Unification of techniques such as Nested Dissection for SAM [112] or Tectonic SAM [114] with the incremental property of iSAM [79] may be particularly promising by combining the advantages of being incremental and scalable to problems unable to fit in core memory.

Incremental improvements in perception and estimation may be made using the data already collected. For example, the actuated laser scanners provided observability in the direction of vehicle motion and could be useful for perception and navigation. The primary difficulty in incorporating data from the actuated scanners is the lack of an absolute orientation of the rotating shaft. An approach to this problem could use self-observation of the vehicle as the basis for optical encoder features. In future research, it would be interesting to apply Bosse and Zlot's sweep matching technique for 3D lidar data registration to data acquired with our platform's rotating Hokuyu scanners.

6.2.2 Autonomy

A principle motivation of this thesis was to develop a system with real-time mapping for integration with online path planning algorithms to achieve multiple objectives. While the planning implementation remains in the scope of future work, we discuss the objectives and potential solution for integrated mapping and path control for capturing marine environments. Autonomy could provide improvement to map quality and prevent some disadvantages of using a human operator. The need to maintain visual control for navigation and radio control from the human operator led to some corruption of the data by the chase boat. Additionally, remote navigation in the presence of occluding obstacles can be precarious, as experienced in our work and also documented by Murphy et al. in [106].

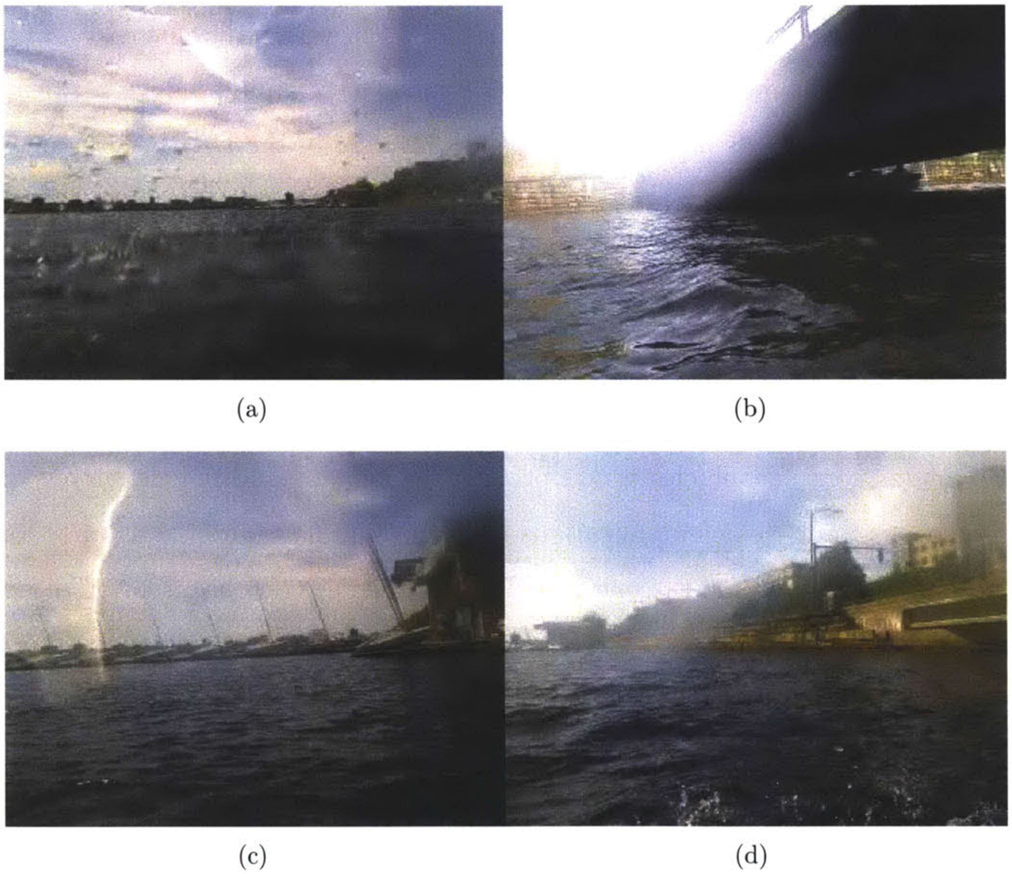
For autonomous mapping in marine environments, we propose the following set of objectives. The vehicle should obey pre-defined operating constraints to bound the operating area as well as the planning problem domain. Planning should provide safe trajectories by avoiding obstacles and maintain a safe water depth, where altitude observability is provided by the sonar sensor. Ideally, the planning algorithm will achieve sensor coverage of the operating area within a confidence bound. Along with coverage and obstacle avoidance, a planning solution should incorporate replanning to adapt plans as the robot gathers new information about the unknown environment, by rewarding exploration of new areas or reduction of uncertainty. Similar use of information gain, or reduction in map entropy, has been demonstrated in Roy et al. and Sim et al. [132, 141].

The objectives stated above do not define a static goal as a *place*, and the goal in the sense of objectives like coverage is a condition for which candidate trajectories can be quantified and ranked for improvement. This type of heuristic is not strict enough for popular goal oriented algorithms such as A* [84] or D* [146]. For a practical application, assumptions and sacrifices of optimality must be made. Two of the more obvious characteristics of the problem preventing a globally optimal planner are incomplete information and limited computational resources. Without perfect prior information about the environment, no algorithm can be guaranteed to find the optimal path. For real-time performance in a dynamic environment, computational resources must be limited since the robot cannot pre-compute the path without a prior map and cannot exhaustively search its search space for an infinite horizon.

There are several notable approaches to autonomy worth discussion for application to marine mapping in complex environments. One approach to autonomy which has demonstrated performance on both marine surface and underwater vehicles in the literature [15, 17] is the `MOOSIvP` framework [14]. The `MOOSIvP` helm module performs multi-objective optimization in a behavior based architecture. In the context of online marine mapping, the previously mentioned objectives would translate to `MOOSIvP` behaviors. For example, separate behaviors would encode the objectives of obstacle avoidance and sensor coverage, and the helm would determine the best action given the behaviors output. This framework has been used for sensor coverage in a marine environment in Shafer's work [138] for a target search problem.

Another promising planning solution is the use of Rapidly Exploring Random Trees (RRTs) [65, 90, 91, 156]. Although early research with RRTs focused on known and/or static environments, more recently RRTs have shown impressive performance in dynamic environments, such as autonomous driving in the DARPA Urban Challenge [93]. The RRT algorithm uses stochastic sampling and pruning to generate trajectory plans. The pruning of infeasible nodes and scoring candidate nodes for expansion within the RRT algorithm may accommodate the objectives for our application.

A further extension to autonomy would be multi-vehicle cooperative operations. Multiple homogeneous vehicles would provide redundancy and reduce survey times. Heterogeneous multi-vehicle applications could include navigation assistance for vehicles unable to safely navigate near structures causing GPS hazards, or the other vehicles lack the sensors to observe obstacles. For underwater vehicles, such a platform presented in this thesis could provide navigation information to the craft [41]. Another example application is to provide a station for Micro Aerial Vehicles (MAVs) to extend range due to limited battery life or as a communication link.



(a)

(b)

(c)

(d)

Figure 6-1: Camera lens distortions may cause difficulties for vision perception. The images show distortions from water on the lenses largely due to splash and wind spray. Lighting effects may be seen in the bright reflection in the lower left of Figure 6-1(c) and the high contrast conditions of Figure 6-1(b).

Appendix A

Harvard Bridge Model

In an effort to generate ground truth data for validating mapping accuracy, a three dimensional model of the Harvard Bridge was created. Here we describe how the bridge model was created. Publicly available documents from the Library of Congress [116] provided the data necessary to create a partial model of the Harvard Bridge[1]. Modelling was limited to the stone piers of the bridge. Incorporating the iron superstructure remains future work. After generating the structural model, the model was georeferenced into the global coordinate frame.

The general plan for the bridge is shown in Figure A-1, and example specifications for most of the piers are shown in Figure A-2. The general plan provides the necessary relative distances between piers and their quantity. The detailed pier specifications were used to generate the layered models of the piers. Our model includes every pier course, or layer of stone, and the rounded ends.

The blueprints are not entirely accurate due to changes made after the original construction. Focusing on the stone pier substructure, the most significant change was the removal of the draw bridge. In Figure A-3, the original plan is compared to

[1] Figures of blueprints and specifications reproduced in this section are courtesy of the Library of Congress

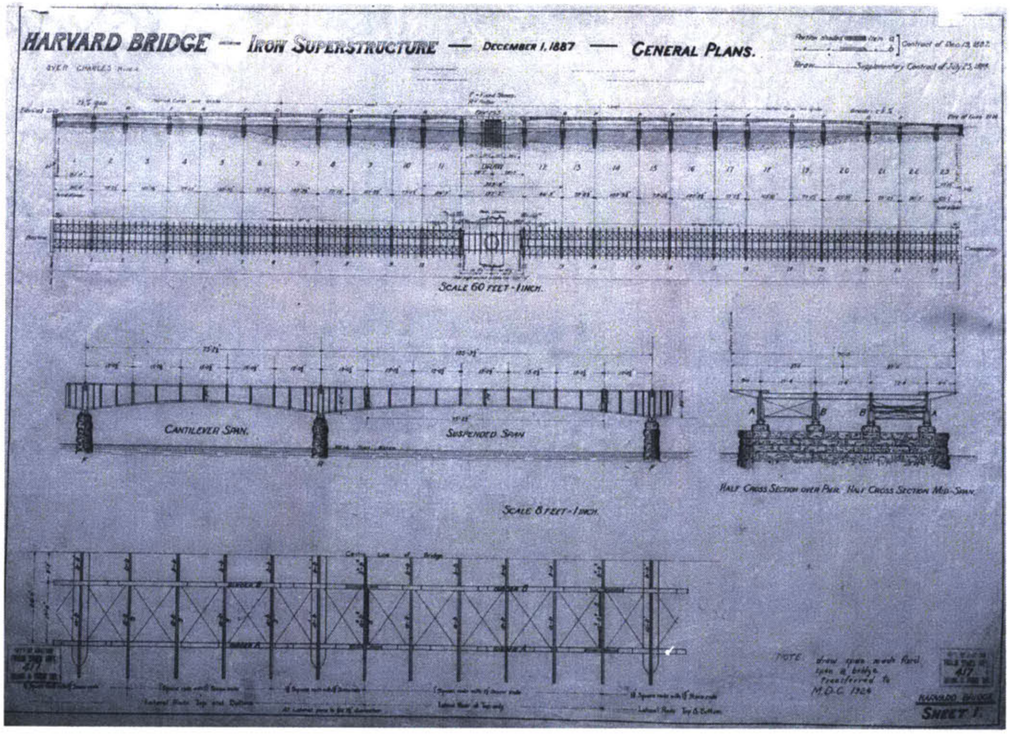

HARVARD BRIDGE — IRON SUPERSTRUCTURE — DECEMBER 1, 1887 — GENERAL PLANS.
SCALE 60 FEET - 1 INCH.
CANTILEVER SPAN.
SUSPENDED SPAN
SCALE 8 FEET - 1 INCH.
HALF CROSS SECTION OVER PIER
HALF CROSS SECTION MID-SPAN
M.D.C. 1924
HARVARD BRIDGE
SHEET 1.

Figure A-1: An original blueprint providing dimensions of the Harvard Bridge structure.

the modern bridge plan. The report documenting the removal of the draw bridge did not include specifications for the pier replacing the drawbridge seat. Consequently, the three-dimensional model generated for this thesis did not include the pier in the location of the former drawbridge. Figure A-4 provides a photograph of the pier not included in the model. Another discrepancy between the blueprint and the prior model is the number of stone seats for the metal girders supporting the bridge platform. While blueprints show four seats as on the right side of Figure A-1, the actual bridge has six stone seats per pier, as seen in Figure 5-11(a) and in the model in Figure A-5.

Georeferencing the model is necessary to compare mapping results with the model in a common coordinate system. The elevation, or $z = 0$, reference was assumed to be the average from the blueprint specifications for mean high water and mean low water relative to the piers, assuming the resulting mean represents the modern water level since the river is no longer tidal. To reference the model in the (x, y) plane, we use georegistered ortho-rectified aerial imagery provided by MassGIS [117]. The aerial imagery provides 15cm resolution. We manually located bridge abutment locations within the images, and used the pixel locations to reference the bridge. Given the bridge abutment locations, the locations of piers are determined using the relative displacements given in the blueprints. The aerial imagery provided pixel locations within the Massachusetts State Plane Coordinates System, which uses a Lambert Conformal Conic Projection and requires conversion to the WGS-84 geographic coordinate system used in this work.

Figure A-5 shows the model of the bridge. The blueprints are sufficient for constructing individual piers and their relative locations, which are shown on the right of Figure A-5. For validation, the model required geo-referencing to place in a global coordinate system. The geo-referenced locations of the bridge piers are shown on the left of Figure A-5.

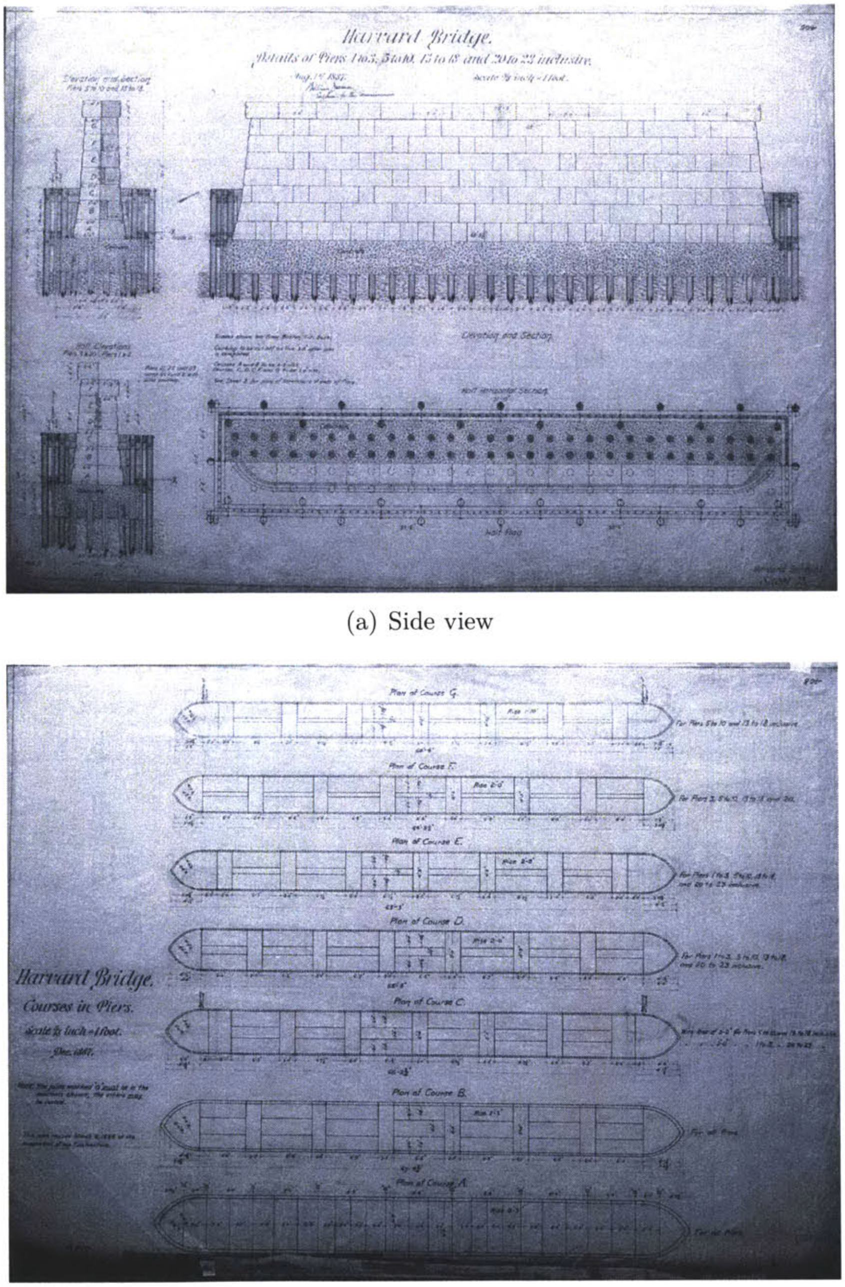
For validation of maps near the Harvard Bridge in Section 5.4, the maps are registered to a sampled version of the model. The model is densely sampled using a million points. To ensure near uniform sampling of the model surfaces, sampling is biased by the area of model faces. The procedure for generating samples on the model is shown in Algorithm 5. Algorithm 5 first chooses a face to sample, and then generates a sample on the selected face. Uniform sampling of a triangular face is shown in Algorithm 6. Algorithm 6 generates a random vertex on a triangle using random barycentric coordinates.

Algorithm 5: Generate random samples of triangular mesh model

```
input  : Integer $N$, the number of desired samples
input  : $list < face > F$, the model as a list of triangular faces
output: $list < vertex3 > P$, sample points with normals
// construct a list of faces and cumulative area
$A \leftarrow 0$
$L \leftarrow list < double, face >$
$L.append(pair(0, null))$
for $f \in F$ do
    $A \leftarrow A + area(f)$
    $L.append(pair(A, f))$
end
// generate samples
for $k \in [1 \dots N]$ do
    // sample faces with area bias
    $d \leftarrow A * rand(0,1)$
    $(y, f) \leftarrow L.lower\_bound(d)$
    // add sample from face
    $P.append(sampleFace(f))$
end
```

Algorithm 6: Generate a random vertex on triangular face f

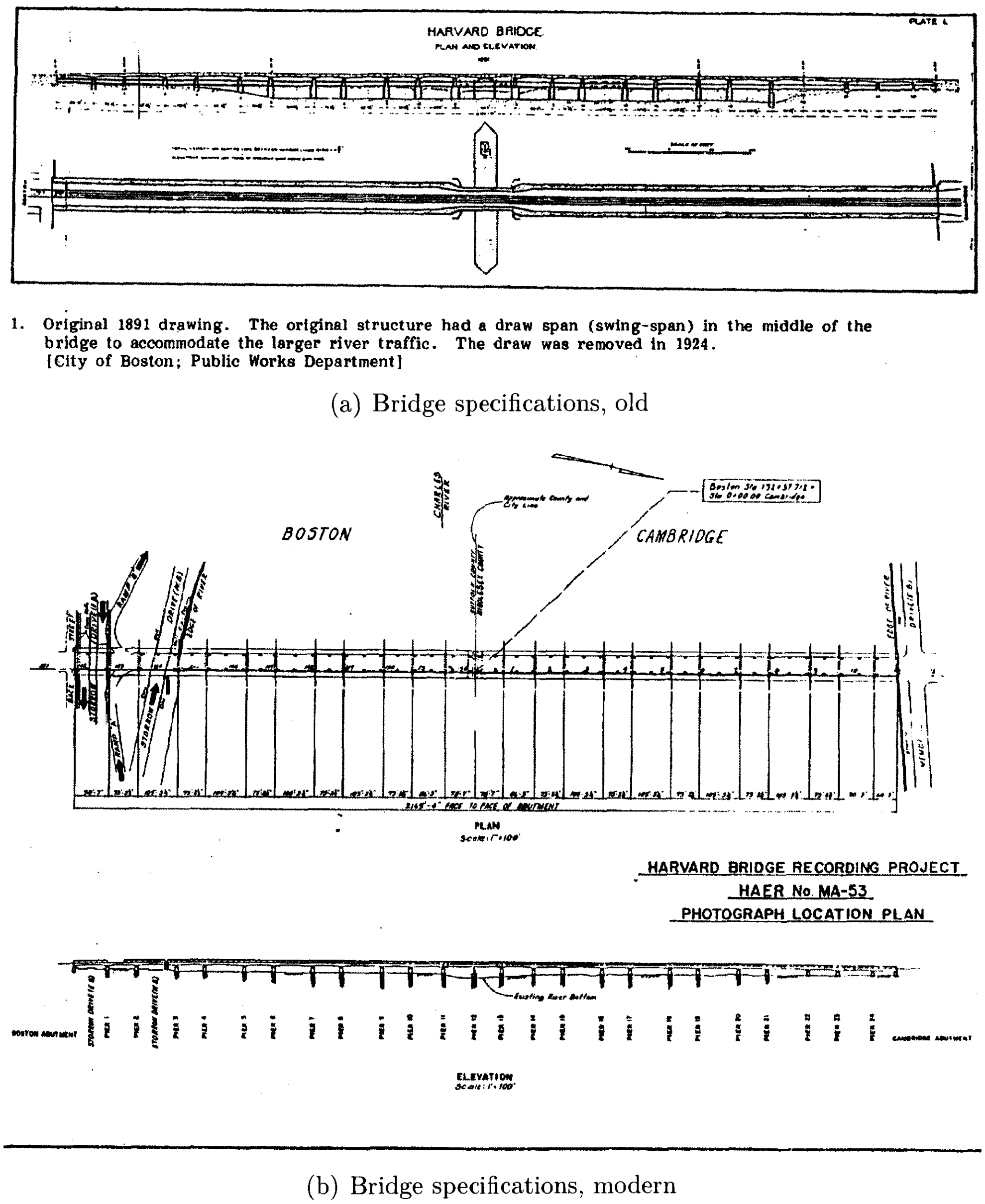
```
input  : Face $f$ to be sampled
output: Vertex $v$ consisting of a point and normal on $f$
// generate a random barycentric coordinate
$a \leftarrow rand(0,1)$
$b \leftarrow rand(0,1)$
// fold if necessary
if $a + b > 1$ then
    $a \leftarrow 1 - a$
    $b \leftarrow 1 - b$
end
$c \leftarrow 1 - a - b$
$v.p \leftarrow a * f.p[0] + b * f.p[1] + c * f.p[2]$
$v.n \leftarrow a * f.n[0] + b * f.n[1] + c * f.n[2]$
```



(a) Side view

(b) Top view

Figure A-2: The blueprints shown provide the dimensions for portions of the piers in the 3D model. Note the piers in these blueprints are not all homogeneous. These blueprints detail the majority of the piers, and similar blueprints (not shown) provide specifications for the piers with dimensions differing significantly from the majority.



1. Original 1891 drawing. The original structure had a draw span (swing-span) in the middle of the bridge to accommodate the larger river traffic. The draw was removed in 1924. [City of Boston; Public Works Department]

(a) Bridge specifications, old

(b) Bridge specifications, modern

Figure A-3: Comparison of the Harvard Bridge plans before and after removal of the draw bridge. Note the increase and renumbering of piers in the more modern plans.

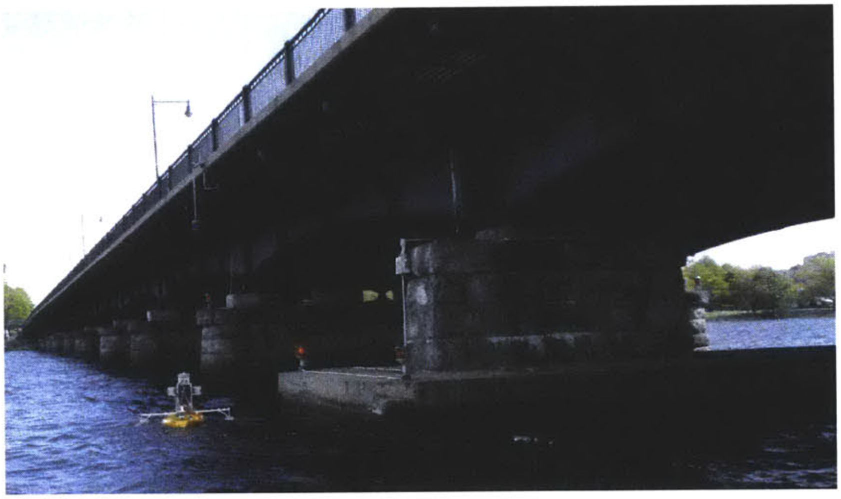


Figure A-4: Our model does not include the pier shown in the foreground near the robot. Original blueprints show a drawbridge, which was removed in 1924. Blueprints of the replacement pier are unavailable.

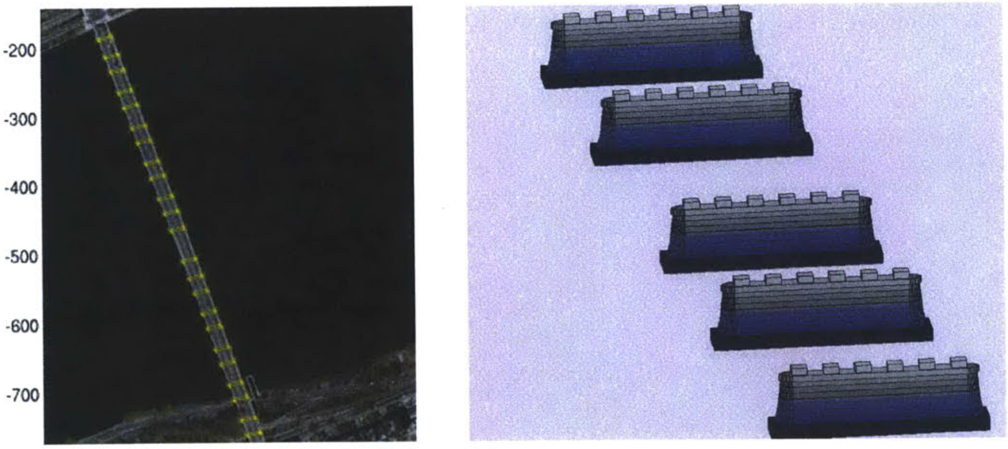


Figure A-5: Model of Harvard Bridge from Library of Congress [116] used for validating map quality. The image at left shows an aerial view of the geo-referenced pier locations. The image at right shows the 3D model of the piers in perspective view, and a semi-transparent blue plane indicates the water level.

THIS PAGE INTENTIONALLY LEFT BLANK

Appendix B

Platform Construction

This chapter provides details concerning the design and construction of the platform used in this thesis. Designing a robot capable of withstanding the conditions of the target environment, the technological limitations of low-cost embedded design, and addressing the perception challenges of marine environments are the key concerns in this thesis. The principal platform in this research is an Autonomous Surface Craft (ASC) of custom design and construction. The chief design objectives were:

- to create an ASC using primarily low-cost commercially available components;
- to provide a flexible interface to accommodate a variety of sensors;
- to provide high on-board processing capabilities;
- to provide both wired and wireless networking; and
- to maintain environmental protection as close to an IP67 rating (submergence up to 1 m depth) as possible.

Platform Overview

The robotic mapping platform is shown in Figure B-1. Table B.1 summarizes the various sensors used on the vehicle during the design process. We provide details of the sensors later in this chapter.

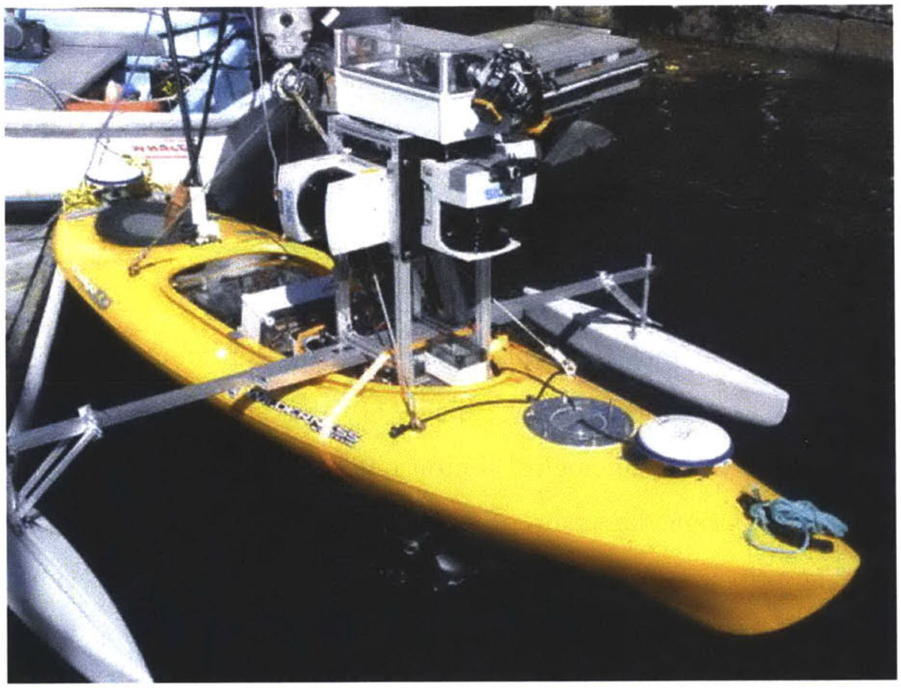


Figure B-1: The kayak and sensor suite is visible while being hoisted into the water.

Construction of the vehicle began with the hull of a stripped SCOUT [40] vehicle as a base. Items left intact from the former scout vehicle were the cooling system, the thruster and servo devices, a radio antenna mount, and parallel 80/20$^{\mathrm{TM}}$aluminum rails within the main compartment.

Systems requiring implementation included: main computational unit, vehicle control circuitry, radio control, and all modifications necessary for sensors. The main computer enclosure incorporated the vehicle control circuitry for simplicity of interconnects and space efficiency.

Sensor	Type	Quantity	Description
Crossbow NAV440 IMU	Localization	1	Position, orientation, and motion
Inertial Plus IMU	Localization	1	Position, orientation, and motion
OS5000 Compass	Localization	1	3-axis Magnetic orientation
Garmin GPS 18	Localization	1	Position
RDI DVL	Localization	1	Vehicle velocity relative to river bed
Blueview MB2200 Sonar	Ranging	1	Subsurface imaging sonar
SICK LMS291	Ranging	3	Surface ranging
Hokuyo UTM30-LX	Ranging	2	Surface ranging (actuated by rotating shaft)
Logitech Quickcam Pro 9000	Vision	1	Single lens webcam
Garmin Radar	Ranging	1	Long range surface

Table B.1: This table lists the sensors incorporated on the vehicle during the design process.

With additional construction and attachment of sensors, the vehicle mass and center of gravity elevated significantly. For additional buoyancy and roll stabilization, a pair of pontoons were attached, see Figure B-2. Although the pontoons increase drag, empirically we found the lower draft of the hull from increased buoyancy to compensate and allow the vehicle to achieve greater maximum velocity than without the pontoons. The pontoons orientation were originally enforced by friction of a screw upon the support rod. However, the pontoons were susceptible to lateral forces causing rotations away from the longitudinal axis of the hull, which resulted in increased drag and decreased vehicle speed. To prevent rotation of the pontoons, a triangle brace was added from the horizontal beam to each pontoon's vertical support shaft with a bolt penetrating the vertical shaft.

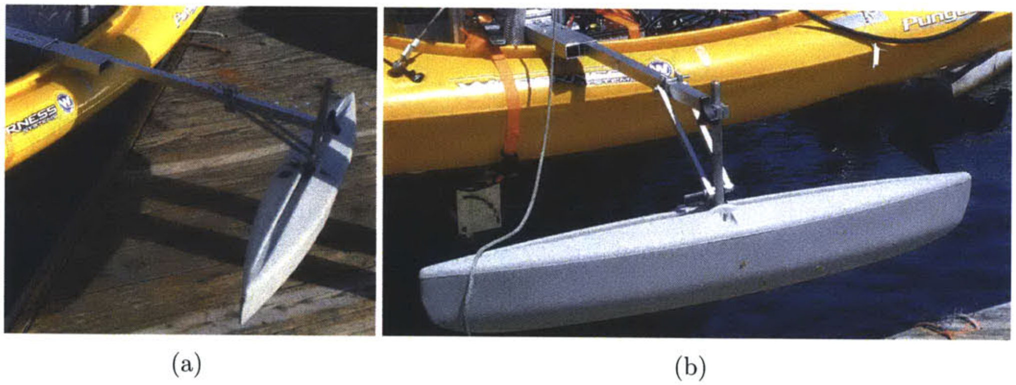

(a) (b)

Figure B-2: A pair of pontoons provide extra buoyancy and roll stabilization.

Enclosures for electronics were NEMA 4x or IP66/67 rated polycarbonate boxes by Fibox®. For the standardized USB and RJ45 (ethernet) interconnects between sensors, support modules, and the computer enclosure, the Bulgin Buccaneer®series of IP68 rated connectors and panel mounts for the enclosures were used. Power supply connections to the 12V main battery were custom connectors, using watertight screw type connector on the enclosure side, and Minn Kota®Quick Connect marine plugs for the power source (battery or fuse block). For the SICK LMS291 laser scanners, custom connectors from each sensor to the LMS support module were created using Bulgin Buccaneer®400 Series connectors.

The principles of modularity and leverage of established standards influenced the design for wiring interconnects for both power and data within the vehicle. All peripherals (sensors) use identical power connectors and ports, and the power source is an unregulated 12V Absorption Glass Mat (AGM) marine battery. All peripheral devices requiring regulated power or different voltage must provide the regulators or transformers at the sensor or in an intermediate support box. Data interconnects adhere to a similar principle. The vehicle supports two types of data transmission: Universal Serial Bus (USB) and RJ45/Ethernet. While many sensors have a serial interface, typically RS232 or RS422, the availability of fast USB to serial converter chips permits the conversion of serial to USB at or near the sensor. A substantial advantage of this convention is the use of homogeneous panel ports and wiring from sensors to the computer box, and reduces the need to create custom cables and ports tied to specific sensors. Very low power devices, such as a compass, simple GPS receiver, or webcam, may be powered passively with USB and not require a dedicated power connection. In addition, USB is a widely adopted protocol, and reduction in cables may be easily achieved with hubs. Such is the case with the sensor support box for the SICK laser scanners, as the box needs only one cable to the computer box for up to three laser scanners as well as one additional generic USB device through the extra input port on the box.

The wiring for power distribution is limited to three connections to the main battery. One connection is reserved for connecting a battery charger for convenience. The second battery connection goes to the main computer box, powering the computers, actuation, and cooling system. Due to the high current demands of the actuation, this connector is distinct from the third connection to the peripheral devices. The third connection for peripherals connects to a common fuse block.

Computation

A single box contains the system's computation and infrastructure logic, as shown in Figure B-3. Within the IP67 rated polycarbonate enclosure resides two computers, wireless amplifiers, an ethernet switch, kill-switch logic, a logic board to control the cooling, thruster and servo motors, and radio-controller.

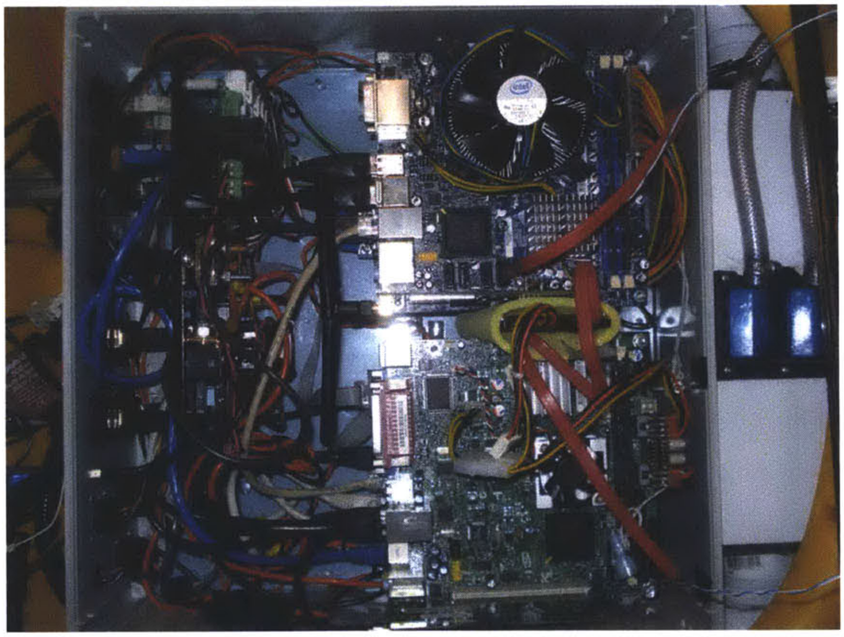
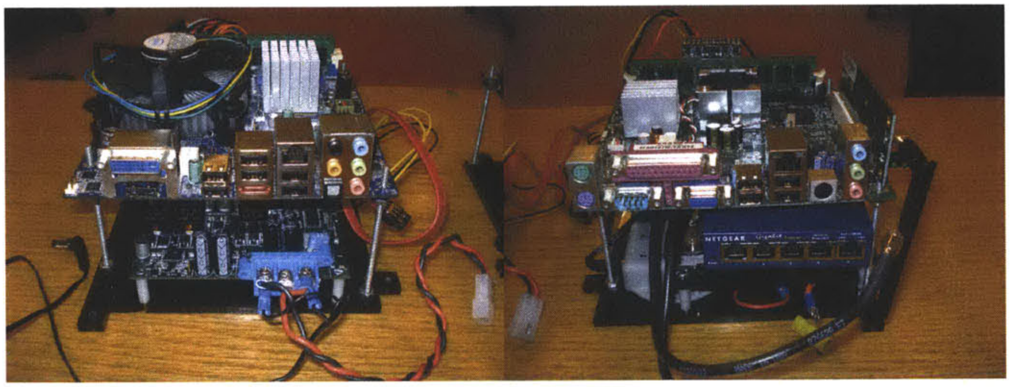


Figure B-3: The vehicle computer enclosure contains two computers, the motherboards on the right, and auxiliary electronics to control cooling, thruster and servo, radio control, motor controllers, and safety switches. Hidden beneath the motherboards are Wifi amplifiers, computer power electronics, and an ethernet hub.

The two computer modules rest upon custom construction and may be removed or inserted independently as modules. Figure B-4 shows the two computer modules removed from the central enclosure. The motherboards are mini-ITX form factor. The mini-ITX form factor motherboards have small footprints of 17cm x 17cm, good



(a) CPU1

(b) CPU2

Figure B-4: The two computer modules in the main computer enclosure. Figure B-4(a) shows the Quad-Core motherboard and power supply board on the lower layer. Figure B-4(b) show the low-power Intel Atom motherboard. The lower layer contains the ethernet switch and a wireless amplifier.

commercial availability, and supports more recent chip technology than other small form factor motherboards such as nano-ITX, pico-ITX, or PC/104. The systems chosen are an Intel Atom dual core 1.6GHz D945CLF2, and an Intel Quad Core Q9550S 2.83GHz CPU placed on an Intel DG45FC motherboard. The Q9550S model is a lower power version of the Q9550, reducing the thermal design power (TDP) from 95 to 65 Watts. Either of the two computers in the enclosure may control the vehicle, although they differ in processing power, power consumption, and heat generation. When under direct sunlight for several hours, especially when removing the vehicle from the water, the cooling system may not dissipate heat for high performance processors. The low power Intel Atom offers a low power alternative when intensive processing is not required.

The other major module within the main enclosure is the vehicle support module shown in Figure B-5. The vehicle support module is mounted vertically within the main enclosure due to space constraints. This module contains several components related to cooling, actuation, and safety. Refer to the labeled regions in Figure B-

5 for the following description. Box *A* contains a relay for the kill-switch. When the external kill switch is activated, this relay prevents Radio Control (R/C) and computer control inputs from affecting the thruster and servo (rudder) output. Box *B* contains a 50 Amp automotive circuit breaker to safely limit current draw to the thruster and servo. Box *C* is a custom logic board with several functions. The board contains voltage regulators for 5, 9, and 12 volts for convenience, an intermediate node for power inputs to the cooling system, and the circuitry for actuation. The actuation circuitry includes a Subtech SES-3 servo switch to provide radio control of the servo and connects the R/C and computer motor controller to the thruster controller and servo. Box *D* contains the thruster controller, a Viktor 884 motor controller, which permits Pulse Width Modulation (PWM) signals from both R/C and the computer motor controller to control thruster speed. Box *E* contains the computer motor controller board, a Pontek sv203. The computer motor controller converts commands from the software driver into PWM signals for the thruster and servo. Also visible in Figure B-5 at the top of Box *C* is a magnetic switch for a computer module. The other computer switch (not visible) is mounted on the opposite side of the board near Box *A*. The magnetic switches allow operators to turn the computers on or off by briefly placing a small magnet near the top of the enclosure lid.

Device Interface

External connections to devices such as sensors or operator laptops are provided with IP68 rated panel mounted ports. The box provides eleven ports, allowing three ethernet connections and eight USB connections. Ports have identical dimensions, allowing fast conversions such as removing a USB connector to allow another RJ45 ethernet connector and vice-versa.

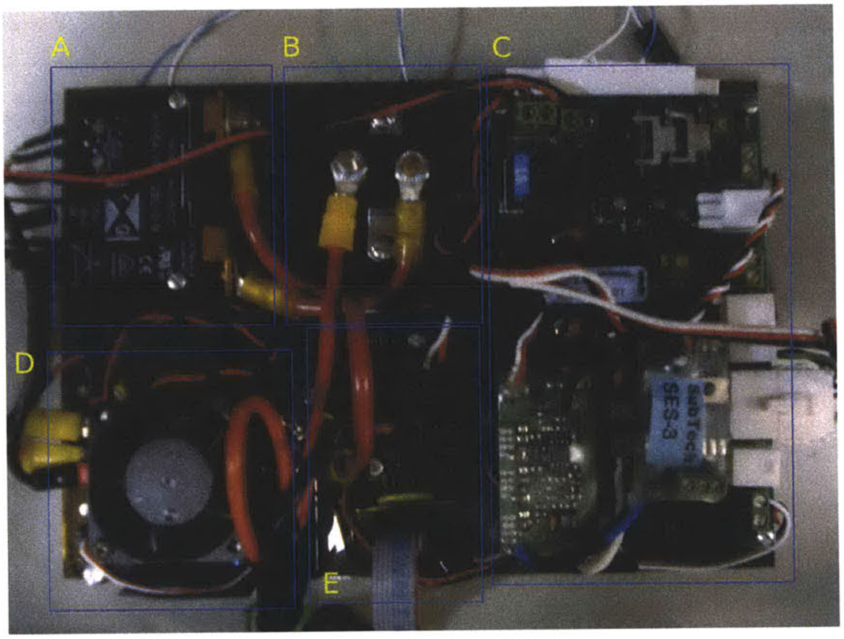


Figure B-5: The vehicle support module removed from the computer enclosure. The box labeled *A* contains the relay for the kill-switch and R/C override. Box *B* is a 50A circuit breaker for safety. Box *C* is the vehicle support board. Box *D* is the thruster controller, and box *E* is the servo motor controller.

One of the design decisions in constructing the vehicle was to minimize the number of connection types to provide as universal an interface as possible. The two connection types chosen were ethernet and USB. While many devices are serial devices, the conversion to USB is achieved externally in sensor support modules.

Networking

The system uses both wireless and wired networking. Each motherboard is equipped with a wireless adapter. Each adapter is connected to an amplifier and routed to an external antenna extending 2m above the hull. The wireless networking permits remote connections to the machine for monitoring or sending commands programmatically when the connectivity is available. While the amplifiers dramatically increase the range at which remote wireless devices may connect to the vehicle, they must amplify their own signal for two-way communication, which can be achieved using a wireless router with a high-gain antenna. Environmental obstructions and extreme distances frequently observed in operations do not permit continuous wireless availability.

The wired ethernet network is heavily utilized. A 5-port switch connects the two motherboards and the three external ethernet connections to a common bus. The common configuration for the external ethernet ports consists of one connection to the IMU/GPS unit, one connection to the Blueview MB2250 imaging sonar, and one connection for use by an operator for downloading data.

Each computer is configured to reside on four networks, one wireless and three wired networks using interface aliasing. The demand for interface aliasing arises due to sensors such as the sonar and IMU having IP addresses assigned by the manufacturer which are not configurable to the user. For consistency, the computers keep the same address within the final octet while the first three octets determine the subnet.

Sensor Support Modules

Sensors have different power and data transmission specifications, and rather than creating custom connections to the main computer box for a single sensor class, modular support boxes perform all the power and data transformations. This allows the use of common cable types from the support modules to the main computer box and does not result in wasted real estate on the computer box for a custom connector when that sensor is not used. Power is provided by a common cable type to the vehicle battery. For sensors such as the SICK laser scanners, a single support module handles up to three such devices, and is shown in Figure B-6. Internally three DC-DC voltage converters provide a clean regulated 24V power source. Fast serial to USB devices convert sensor data from RS-232 to USB, and a wide (voltage) input USB hub aggregates the data for a single USB data cable to the main computer. For Blueview sonars used during experiments, a support module providing power over ethernet (PoE) permits up to two such devices is shown in Figure B-7. Power is provided from a voltage regulator to the pins of the spare twisted pairs of the device ethernet jack. The twisted pairs were blue for positive and brown for negative voltage (pins 4,5 and 7,8). While this non-phantom PoE configuration precludes gigabit speeds, 10/100 Mb/s ethernet speeds are more than sufficient for the bandwidth.

Inertial Measurement Unit

The primary proprioceptive position and motion sensor is the Oxford Technical Solutions (OXTS) Inertial+ integrated Inertial Measurement Unit (IMU) and Global Positioning System (GPS) sensor. This sensor utilizes Microelectromechanical (MEMS) technology for its accclerometers and rate gyroscopes. Alternative technologies such as Ring Laser Gyro (RLG) and Fiber Optic Gyro (FOG) offer greater accuracies at much higher costs. The Inertial+ uses a dual antenna GPS system to estimate

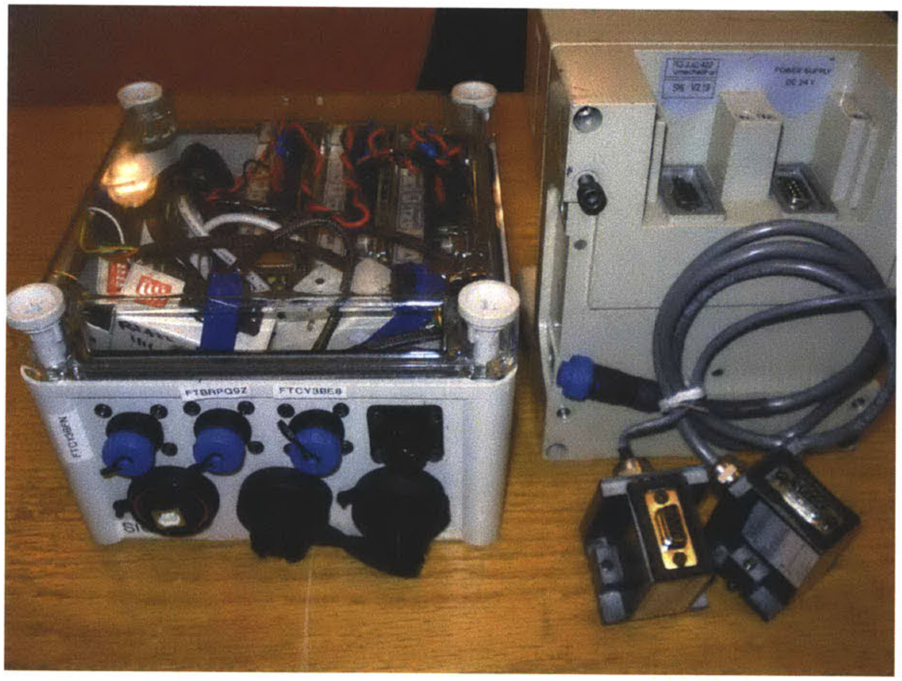


Figure B-6: Sensor support enclosure for three SICK laser scanners and one generic usb device to route through the internal USB hub.

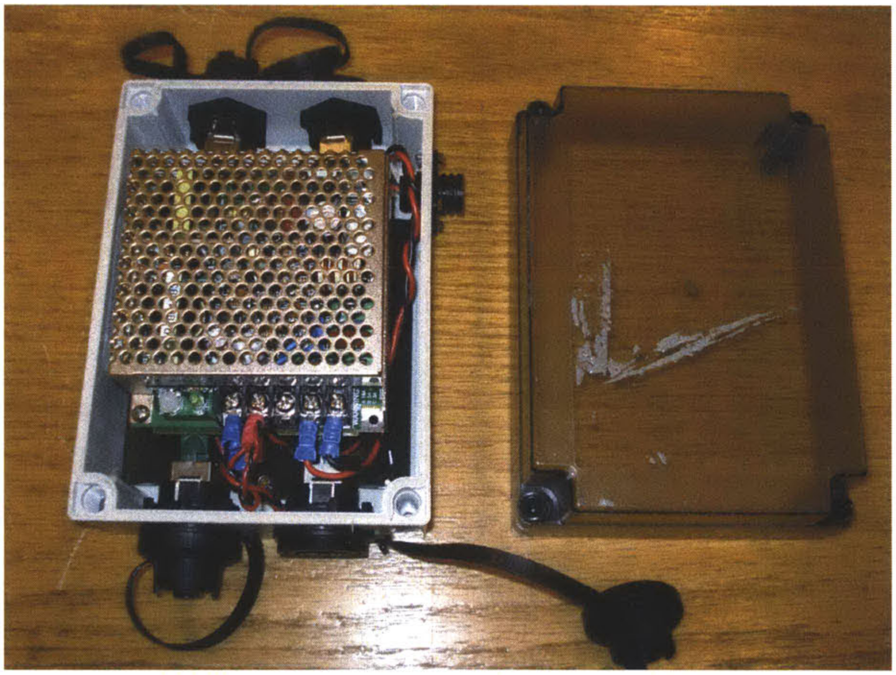


Figure B-7: Sensor support module for two sonar sensors providing Power over Ethernet (PoE) and data transmission.

vehicle heading to reported accuracies of 1 degree, assuming optimal antenna placement, configuration settings, sufficient initialization period, and unobstructed view of many satellites. The OXTS RT2000 IMU reports data packets at 100Hz, although specific data payload packets beyond the main packet are cycled. Available data include accelerations and angular rates (corrected for scale factor and bias), velocities, orientation, and GPS position.

In the initial experimentation, a different, cheaper IMU was used. A Crossbow Nav440 provided a GPS receiver, MEMS technology accelerometers and rate gyroscopes, and magnetometers for orientation. While both the OXTS and Crossbow units used MEMS technology for rate gyros and accelerometers, the OXTS unit provided stabler and more accurate accelerations and rates than the Crossbow. Both units incorporate GPS receivers, although only the OXTS unit utilizes dual antennas for improved heading and velocity estimates. The Crossbow utilizes magnetometers to enable absolute orientation estimates. Experimentally we encountered degraded performance in the target environment with the Nav 400 due to both GPS and magnetic hazards.

Magnetic Compass

Another orientation sensor used is the Ocean Server OS5000 3-axis magnetic compass. This is a low-cost sensor providing heading, pitch, and roll measurements at 5-10Hz. Since it is a magnetic sensor, it reports heading relative to magnetic north, which must be corrected by the magnetic variation of the operating area (-15.6 degrees during experiments) to obtain heading relative to true north. The magnetic compass provides redundancy of orientation measurements offered by the IMU. It provided data for evaluating relative performance of the sensors. The update rate of this sensor in the final driver implementation was approximately 10Hz.

In comparison to the much more expensive IMU, the OS5000 compass is less accurate, and suffers from hazards that do not affect the MEMS-based IMU. Currents from electronics, especially wires supplying power, induce a magnetic field that may interfere with the ability of the device to detect the weak magnetic field of the earth. Placement of the compass on the vehicle took this interference into consideration. subject to constraints of the physically small hull and practical mount points.

Another compass hazard is from magnetic disturbances in the environment. Deviations from the Earth's magnetic field may be due to ferrous metals in the structure and possibly from overhead passing traffic or induction loops in the pavement for demand-based traffic signal detection. This effect was observed experimentally when the vehicle traversed under the Harvard Bridge, which caused a disruption in the compass readings. In Figure B-8, the three axis magnetic compass sensor shows a clear sinusoidal deviation in duration from approximately 1310 seconds to 1340 seconds, which was the approximate time navigating underneath a bridge. During the same interval, the GPS heading, shown in blue, varies widely. In an attempt to establish a true heading, a SLAM system using line-contour matching and constant velocity prediction model was created and the results are shown in black. The cyan colored line is simply the filter results with constant offset to magnetic north for comparison to the magnetic compass. The line-contour localization approach began as a scan-matching system, but due to the corridor geometry and anisotropic sampling, updates reversing the principle direction of motion were noted. To alleviate this problem, line contours were extracted and updates were incorporated along the principal component (the normal). Both approachcs would require more sophisticated policies to permit general operation as the vehicle enters such environments and upon exit, where few, if any, ranges are observed.

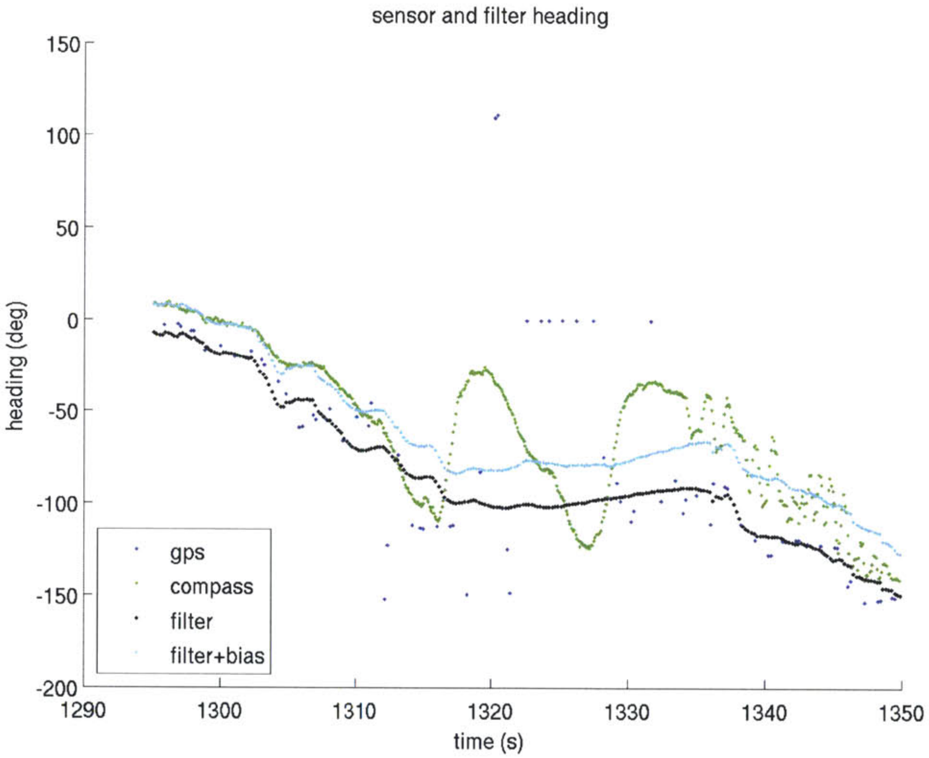


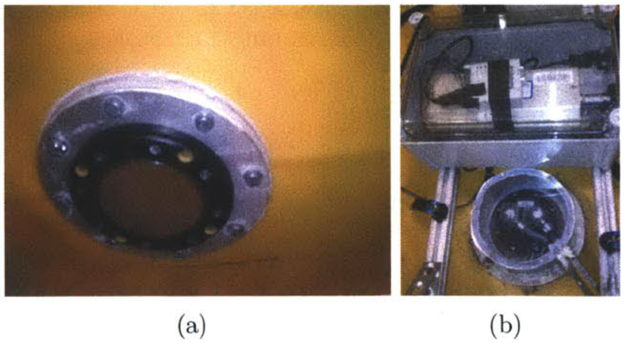
Figure B-8: Magnetic compass (green) experiences disturbances when the vehicle traverses under a bridge from approximately time 1310 to 1335. GPS heading (blue, scattered) varies wildly due to satellite signal loss/distortion. A reference heading (black) is provided from laser localization and also shown adjusted with offset to magnetic north (cyan).

Garmin GPS

A low-cost single antenna Garmin GPS18 sensor provides an additional source of GPS derived positioning. This sensor is a redundant source of GPS measurements in addition to the Inertial+, but at lower accuracy. In addition to evaluating sensor performance characterization via comparison, the Garmin GPS provides an interface with data unavailable from the Inertial+. For example the Garmin GPS will report the Pseudo Random Number (PRN) of satellites it is tracking, estimates of their azimuth and elevation, and the signal to noise ratio. Although the Inertial+ does not report such details, it is probable the Inertial+ is using a subset of those satellites in its calculations under the assumption that proximity of the sensors results in similar satellite constellation views. The Garmin GPS provides updates at approximately 5Hz, and the driver in preliminary experiments updated at 5Hz, but a later version updated at only 1.6Hz because more message types were requested, such as satellite information.

Doppler Velocity Logger

The vehicle is equipped with a Teledyne RD Instruments Explorer Doppler Velocity Logger (DVL). This sensor uses four acoustic beams at 600kHz and measures doppler shift of the beam echos from the bottom (riverbed or ocean floor) and reports an estimate of the device's velocity relative to the bottom. Mounted under the hull of the kayak as shown in Figure B-9 the DVL provides velocity measurements for estimating vehicle motion.



(a) (b)

Figure B-9: The DVL sensor is mounted through the hull of the vehicle. The sensor is shown below the hull in Figure B-9(a). Inside the vehicle, a waterproof enclosure protects the DVL data interface, and a polycarbonate enclosure protects the sensor's logical unit as shown in Figure B-9(b).

There are several hazards to using the DVL as a velocity sensor. The sensor only reports valid data when it achieves *bottom lock*, which means it can accurately track the bottom surface. Loss of bottom lock may occur when operating in depths too deep for the settings or device or too shallow, in muddy environments (such as the Charles River), or in the presence of debris such as marine vegetation or garbage. Marine life such as schools of fish may cause loss of bottom lock or erroneous velocities. For the experiments in this thesis, the DVL driver was configured to operate at 1Hz, but the actual update rate observed was 0.5Hz.

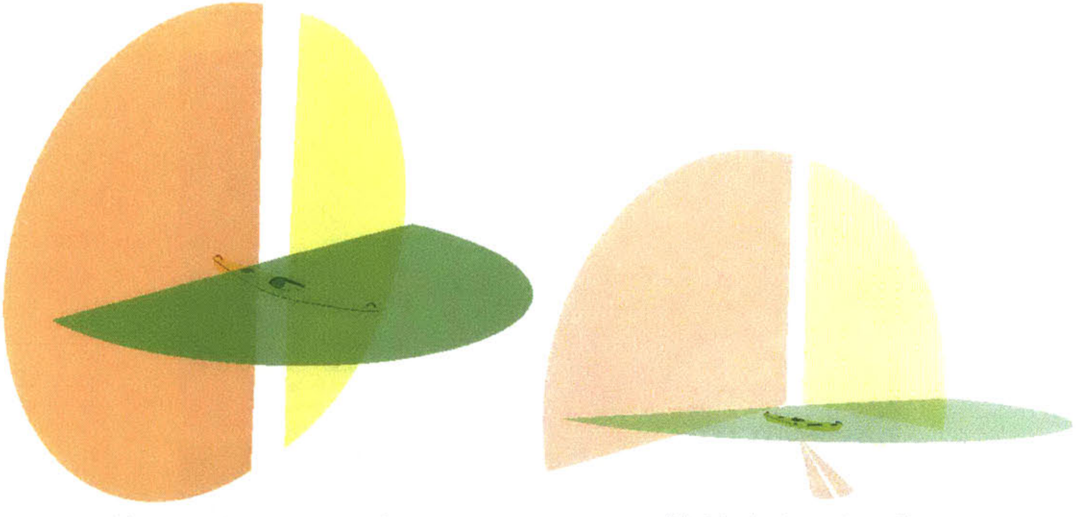
Surface Laser Range Sensors

The vehicle utilizes two types of scanning laser range sensors: the SICK LMS291 and Hokuyo UTM-30LX. These sensors use lasers with wavelengths of 905nm and 870nm, respectively, which is significant when considering performance in a marine environment. The absorption spectrum of water has a medium band near 900nm, which effectively makes the water surface invisible to the sensors. The SICK LMS291 scanner and very similar models from the manufacturer are widely used in mobile robotics [24,26,37,45,72,120,134]. Experiments in this thesis statically mounted up to three SICK laser sensors, as seen in Figure 1-3. The SICK LMS291 is capable of producing scans at 75Hz with millimeter range resolution. In 75Hz operation, the scans are interlaced, and the sensor cycles through four scan configurations. The scan configurations have 1.0 degree separation between samples in the scan and differ in number of samples and beginning angular offset. The number of samples and angular offsets are shown in Table B.2. The maximum range was set at 32 meters (the maximum range permitted with millimeter range resolution) for the configuration employed during experiments, yielding the field of view shown in Figure B-10.

The second type of laser sensor is the Hokuyo UTM-30LX. This sensor is relatively new and has become popular for robotics researchers in recent years. It operates in a

Table B.2: During 75Hz operation, the SICK LMS291 interlaces scans in cycles of four. The scans differ in number of samples and beginning angular offset.

Samples	Offset (degrees)
181	0.00
180	0.25
180	0.50
180	0.75



(a) Static laser sensing planes

(b) Masked sensing planes

Figure B-10: The sensor field of view for the static laser range sensors on the vehicle. Masks are applied as shown on the right for the effective sensing region. For comparison the figure on the right also shows the sensor range of the sonar.

mode providing scans of 1440 samples at .25 degree angular resolution and 30-50mm accuracy with maximum range of 30 meters. Compared to the SICK sensors, the Hokuyo offers a greater field of view at 270 degrees rather than 180 degrees, a smaller form factor, and simpler wiring interface (USB).

The Hokuyo laser scanners are mounted on a rotating shaft configured with velocity control to complete approximately one revolution per second[1]. The configuration is shown in Figure 1-3 and more clearly in Figure B-11. The two scanners were mounted in an asymmetric manner on a plate at the end of the rotation shaft. The center field of view (FOV) of one scanner is parallel with the shaft looking forward. The other scanner's center FOV was directed along rotation of the shaft. The shaft rotates with positive roll. The sensors rotate in the direction of the upper arrow in Figure B-11.

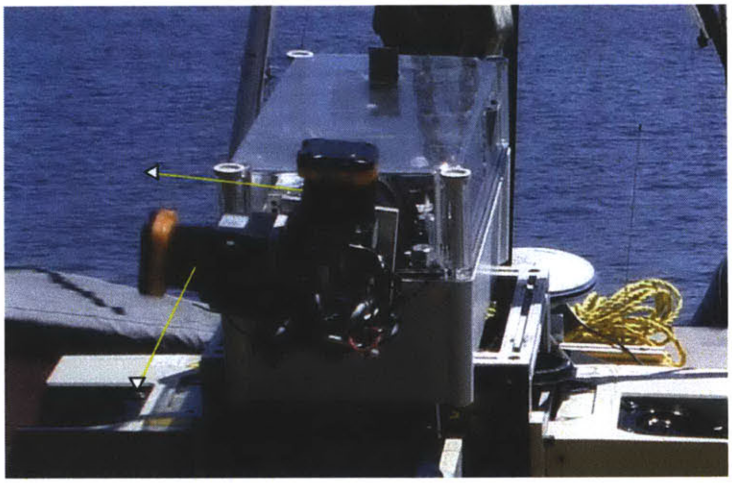


Figure B-11: The Hokuyo laser scanners are mounted on a rotating shaft (counter clockwise relative to reader). Arrows indicate the direction of the center beam of each scanner.

[1] The author acknowledges Luke Fletcher for providing the design of the rotating assembly.

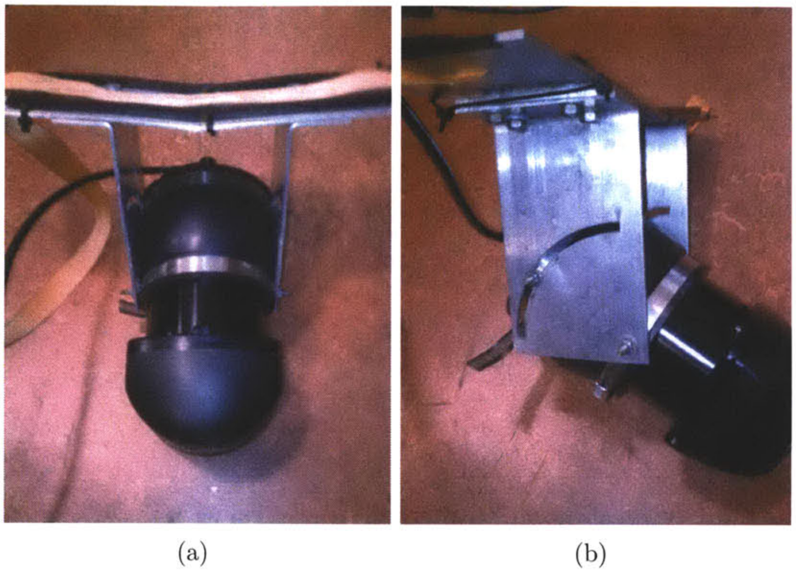
Subsurface Sonar Sensors

Multibeam imaging sonars provided the subsurface sensing. The vehicle permits use of both the Blueview 900kHz and 2250kHz microbathymetry sonar. The 2250kHz version offers greater resolution reported at 1 degree but has a maximum range of 10m rather than 40m for the 900kHz. Also the 900kHz has a vertical main lobe width of 15-20 degrees, which complicates detection estimation since elevation angle is unobservable. Experimental results used the MB2250 sonar.

The actual ping rate for the sonar is variable and depends on process load. The maximum feasible ping rate before inter-ping interference becomes noticeable is approximately 20Hz. Using the faster Quad-Core computer without beamforming the images, a ping rate of 10-12Hz was observed. When beamforming images of low resolution, the ping rate drops to approximately 5Hz.

The mount for the sonar is a custom low-cost aluminum design. Figure B-12 shows the sonar mount unattached to the vehicle. The mount consists of a hull plate conforming roughly to the *vee* of the vehicle hull with a slot at each end for straps. We use cheap and easily available ratcheting tie-down straps to secure the mount to the vehicle. The side of the hull plate in contact with the vehicle is outfitted with rubberized strips to prevent slippage. Two vertical plates approximately 13cm apart extend 16cm from the hull plate for the sonar cradle. Each of the two plates contains a small diameter (6mm) hole for a bolt at the forward corner. An arc of radius 83mm from the forward hole begins, spanning 45 degrees and with 6mm width. The forward hole and arc provide a simple pivoting mechanism for adjusting the sonar angle.

A custom sleeve secures the sonar to the cradle. The sleeve is very simple and composed of very low cost components found in hardware stores. We place two screw clamps on a 4 inch diameter rubber plumbing connecter (11cm length) with separation of the screw clamps matching the hole and slots in the cradle. Two holes are punched



(a) (b)

Figure B-12: The sonar sensor mount provides adjustable pitch angle and a simple strap mechanism for attachment and removal from the vehicle.

through the band of each screw clamp, and bolts are placed through them such that the head of the bolt rests against the rubber sleeve and threads are exposed outside of the clamps. The bolts are placed through the hole and slot for each side of the cradle, and a matching nut and washer secure the sleeve to the cradle.

To mount the sonar, the sonar is first pushed into the sleeve, leaving at least a 4cm gap between the sonar head and sleeve. The sonar/sleeve is pitched to the desired angle before tightening the nuts for the bolts on the rear arc. Finally, all nuts are tightened. When properly tightened, the cradle plates have a visible bend, which means the cradle arms are compressing the sleeve and the screw clamps are compressing the sonar, preventing slippage.

Bibliography

[1] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski. Building Rome in a day. In *Intl. Conf. on Computer Vision (ICCV)*, Kyoto, Japan, 2009.

[2] B. Allen, R. Stokey, T. Austin, N. Forrester, R. Goldsborough, M. Purcell, and C. von Alt. REMUS: a small, low cost AUV; system description, field trials and performance results. In *Proc. of the IEEE/MTS OCEANS Conference and Exhibition*, volume 2, pages 994–1000, October 1997.

[3] C. Von Alt, B. Allen, T. Austin, N. Forrester, R. Goldsborough, M. Purcell, and R. Stokey. Hunting for mines with REMUS: a high performance, affordable, free swimming underwater robot. In *Proc. of the IEEE/MTS OCEANS Conference and Exhibition*, pages 117–122, 2001.

[4] M. Antone and S. Teller. Automatic recovery of relative camera rotations for urban scenes. In *Int. Conf. Computer Vision and Pattern Recognition*, volume 2, pages 282–289, 2000.

[5] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9:698–700, 1987.

[6] Nationa Marine Electronics Association. NMEA 0183 standard. http://www.nmea.org/content/nmea_standards/nmea_083_v_400.asp, 2010.

[7] A. Bachrach, R. He, and N. Roy. Autonomous flight in unstructured and unknown indoor environments. In *European Micro Aerial Vehicle Conference*, Sep 2009.

[8] A. Bahr. *Cooperative Localization for Autonomous Underwater Vehicles.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, Feb 2009.

[9] T. Bailey and H.F. Durrant-Whyte. Simultaneous localisation and mapping (SLAM): Part I the essential algorithms. *Robotics and Automation Magazine*, 13(2):99–110, June 2006.

[10] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan. *Estimation with Applications to Tracking and Navigation.* John Wiley & Sons, Inc., New York, NY, USA, 2001.

[11] C. Beall, B. Lawrence, V. Ila, and F. Dellaert. 3D reconstruction of underwater structures. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2010.

[12] E.O. Belcher, H.Q. Dinh, DC Lynn, and TJ Laughlin. Beamforming and imaging with acoustic lenses in small, high-frequency sonars. In *Proc. of the IEEE/MTS OCEANS Conference and Exhibition*, volume 3, pages 1495–1499, 1999.

[13] O. Bengtsson and A. Baeveldt. Robot localization based on scan-matching–estimating the covariance matrix for the IDC algorithm. *IEEE Transactions on Robotics and Automation*, 2004.

[14] M. Benjamin. The pHelmIvP module: A description and guide for users. Technical report, www.moosivp.org, 2006.

[15] M. Benjamin and J. Curcio. COLREGS-based navigation in unmanned marine vehicles. In *AUV-2004*, Sebasco Harbor, Maine, June 2004.

[16] M. Benjamin, J. Curcio, J. Leonard, and P. Newman. Navigation of unmanned marine vehicles in accordance with the rules of the road. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Orlando, Florida, May 2006.

[17] M. Benjamin, J. Curcio, J. Leonard, and P. Newman. Protocol-based COLREGS collision avoidance navigation between unmanned marine surface craft. *J. of Field Robotics*, 23(5), May 2006.

[18] J. L. Bentley. K-d trees for semidynamic point sets. In *Proceedings of the sixth annual symposium on Computational geometry*, SCG '90, pages 187–197, New York, NY, USA, 1990. ACM.

[19] J. L. Bentley and M. I. Shamos. Divide-and-conquer in multidimensional space. In *Proceedings of the eighth annual ACM symposium on Theory of computing*, STOC '76, pages 220–230, New York, NY, USA, 1976. ACM.

[20] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin. The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):349–359, 1999.

[21] V. Bertram. Unmanned surface vehicles–a survey. *Skibsteknisk Selskab*, 2008.

[22] P.J. Besl and N.D. McKay. A method for registration of 3-d shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 14(2):239–256, Feb 1992.

[23] M. Bosse, P. Newman, J. Leonard, and S. Teller. Simultaneous localization and map building in large-scale cyclic environments using the Atlas framework. *Intl. J. of Robotics Research*, 23(12):1113–1139, Dec 2004.

[24] M. Bosse and R. Zlot. Map matching and data association for large-scale two-dimensional laser scan-based SLAM. *Intl. J. of Robotics Research*, 27:667–691, 2008.

[25] M. Bosse and R. Zlot. Continuous 3D scan-matching with a spinning 2D laser. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Kobe, Japan, May 2009.

[26] M. Bosse and R. Zlot. Keypoint design and evaluation for place recognition in 2d lidar maps. *Journal of Robotics and Autonomous Systems*, 2009.

[27] M. C. Bosse. *ATLAS: a framework for large scale automated mapping and localization*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2004.

[28] M. Caccia, M. Bibuli, R. Bono, Ga. Bruzzone, Gi. Bruzzone, and E. Spriandelli. Unmanned surface vehicle for coastal and protected waters applications: the charlie project. *Marine Technology Society Journal*, 41(2):62–71, 2007.

[29] J. Canny. A computational approach to edge detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8(6):679–698, Nov 1986.

[30] J. D. Carlson. *Mapping Large, Urban Environments with GPS-Aided SLAM*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, August 2010.

[31] J. Castellanos, J. Neira, and J. Tardós. Multisensor fusion for simultaneous localization and map building. *IEEE Transactions on Robotics and Automation*, 17(6):908–914, December 2001.

[32] T. Chang, L. Wang, and F. Chang. A solution to the ill-conditioned GPS positioning problem in an urban environment. *IEEE Trans. on Intelligent Transportation Systems*, 10:135–145, March 2009.

[33] R. Chellappa and A. Jain. *Markov Random Fields: Theory and Applications*. Academic Press, San Diego, CA, 1993.

[34] P. Cignoni, M. Corsini, and G. Ranzuglia. Meshlab: an open source 3D mesh processing system. http://meshlab.sf.net, April 2008.

[35] P. Cignoni, F. Ponchio, N. Pietroni, M. Corsini, G. Ranzuglia, M. Di Benedetto, M. Tarini, and M. Dellepiane. Visual computing library. http://vcg.sourceforge.net, 2004.

[36] J. T. Cobb, B. Schulz, and G. Dobeck. Forward-looking sonar target tracking and homing from a small AUV. In *Proc. of the IEEE/MTS OCEANS Conference and Exhibition*, pages 1–7, September 2005.

[37] D. Cole and P.Newman. Using laser range data for 3D SLAM in outdoor environments. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Orlando, Florida USA, May 2006.

[38] R. G. Cowell, A. P. Dawid, S. L. Lauritzen, and D. J. Spiefelhalter. *Probabilistic Networks and Expert Systems, Statistics for Engineering and Information Science.* Springer Verlag, 1999.

[39] Y. Cui and S. Sam Ge. Autonomous vehicle positioning with GPS in urban canyon environments. *IEEE Trans. Robotics and Automation*, 19, Feb 2003.

[40] J. Curcio, J. Leonard, and A. Patrikalakis. SCOUT — a low cost autonomous surface platform for research in cooperative autonomy. In *Proc. of the IEEE/MTS OCEANS Conference and Exhibition*, September 2005.

[41] J. Curcio, J. Leonard, J. Vaganay, A. Patrikalakis, A. Bahr, D. Battle, H. Schmidt, , and M. Grund. Experiments in Moving Baseline Navigation using Autonomous Surface Craft. In *Proc. of the IEEE/MTS OCEANS Conference and Exhibition*, volume 1, pages 730–735, September 2005.

[42] F. Dellaert. Square root SAM: Simultaneous location and mapping via square root information smoothing. In *Robotics: Science and Systems (RSS)*, 2005.

[43] F. Dellaert and M. Kaess. Square Root SAM: Simultaneous localization and mapping via square root information smoothing. *Intl. J. of Robotics Research*, 25(12):1181–1204, Dec 2006.

[44] F. Delleart, J. Carlson, V. Ila, K. Ni, and C. Thorpe. Subgraph-preconditioned conjugate gradients for large scale SLAM. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2010.

[45] A. Diosi and L. Kleeman. Uncertainty of line segments extracted from static sick laser scans. In *Australasian Conference on Robotics and Automation*, pages 1–6, 2003.

[46] M.W.M.G. Dissanayake, P. Newman, S. Clark, H.F. Durrant-Whyte, and M. Csorba. A Solution to the Simultaneous Localization and Map Building (SLAM) Problem. *IEEE Transactions on Robotics and Automation*, 17(3):229–241, June 2001.

[47] H.F. Durrant-Whyte and T. Bailey. Simultaneous localisation and mapping (SLAM): Part I. *Robotics & Automation Magazine*, 13(2):99 –110, Jun 2006.

[48] A. Elfes. Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation*, RA-3(3):249–265, June 1987.

[49] A. Elfes. Integration of sonar and stereo range data using a grid-based representation. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 1988.

[50] A. Elfes. *Occupancy grids: A probabilistic framework for robot perception and navigation.* PhD thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, 1989.

[51] A. Eliazar and R. Parr. Dp-SLAM: Fast, robust simultateous localization and mapping without predetermined landmarks. In *Intl. Joint Conf. Artificial Intelligence*, 2003.

[52] A. Eliazar and P. Ronald. Hierarchical linear/constant time slam using particle filters for dense maps. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 339–346. MIT Press, Cambridge, MA, 2006.

[53] R. Eustice. *Large-Area Visually Augmented Navigation for Autonomous Underwater Vehicles.* PhD thesis, Massachusetts Institute of Technology / Woods Hole Oceanographic Joint-Program, June 2005.

[54] R. Eustice, H. Singh, J. Leonard, and M. Walter. Visually mapping the RMS Titanic: conservative covariance estimates for SLAM information filters. *Intl. J. of Robotics Research*, 25(12):1223–1242, 2006.

[55] R. Eustice, L. Whitcomb, H. Singh, and M. Grund. Experimental results in synchronous-clock one-way-travel-time acoustic navigation for autonomous underwater vehicles. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Rome, Italy, April 2007.

[56] R.M. Eustice, L.L. Whitcomb, H. Singh, and M. Grund. Recent advances in synchronous-clock one-way-travel-time acoustic navigation. In *Proc. of the IEEE/MTS OCEANS Conference and Exhibition*, pages 1–6, Boston, MA, USA, September 2006.

[57] N. Fairfield. *Localization, Mapping, and Planning in 3D Environments.* PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, January 2009.

[58] N. Fairfield, G. A. Kantor, and D. Wettergreen. Three dimensional evidence grids for SLAM in complex underwater environments. In *Proceedings of the 14th International Symposium of Unmanned Untethered Submersible Technology (UUST)*, Lee, New Hampshire, August 2005. AUSI.

[59] N. Fairfield, G. A. Kantor, and D. Wettergreen. Real-time SLAM with octree evidence grids for exploration in underwater tunnels. *J. of Field Robotics*, 2007.

[60] N. Fairfield and D. Wettergreen. Active localization on the ocean floor with multibeam sonar. In *Proc. of the IEEE/MTS OCEANS Conference and Exhibition*, Quebec, Canada, September 2008.

[61] J.P. Fish and H.A. Carr. *Sound underwater images: a guide to the generation and interpretation of side scan sonar data.* Lower Cape Pub Co, 1990.

[62] J. Folkesson. *Simultaneous Localization and Mapping with Robots.* PhD thesis, Computer Science and Communication, Royal Institute of Technology, SE-100 44 Stockholm, Sweden, 2005.

[63] J. Folkesson and H. I. Christensen. Graphical SLAM - a self-correcting map. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, volume 1, pages 383–390, 2004.

[64] J. Folkesson, J. Leedekerken, R. Williams, A. Patrikalakis, and J. Leonard. A feature based navigation system for an autonomous underwater robot. *The Intl. Conf. on Field and Service Robots (FSR)*, July 2007.

[65] E. Frazzoli, M. Dahleh, and E. Feron. Real-time motion planning for agile autonomous vehicles. *AAIA Journal of Guidance, Control, and Dynamics*, 25(1), 2002.

[66] M. Gary, N. Fairfield, W. C. Stone, and D. Wettergreen. 3D mapping and characterization of sistema zacatón from DEPTHX (DEep Phreatic THermal eXplorer). In *Proceedings of KARST08: 11th Sinkhole Conference ASCE*, 2008.

[67] Google. Google maps with street view. http://www.google.com/intl/en_ALL/help/maps/streetview/index.html, 2007.

[68] G. Grisetti, C. Stachniss, and W. Burgard. Improved grid-based SLAM with rao-blackwellized pariticle filters by adaptive proposals and selective resampling. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, volume 1, pages 2432–2438, 2005.

[69] G. Grisetti, C. Stachniss, S. Grzonka, and W. Burgard. A tree parameterization for efficently computing maximum likelihood maps using gradient descent. In *Robotics: Science and Systems (RSS)*, Atlanta, GA, 2007.

[70] J. Gutmann and C. Schlegel. AMOS: comparison of scan matching approaches for self-localization in indoor environments. In *Proc. of the First Euromicro on Advanced Mobile Robot*, pages 61–67, 1996.

[71] M. Hadwiger, C. Berger, and H. Hauser. High quality two-level volume rendering of segmented data sets on consumer graphics hardware. In *Proc. of the 14th IEEE Visualization Conference (VIS03)*, pages 301–308, Seattle, Washington, USA, October 2003.

[72] A. Harrison and P. Newman. High quality 3D laser ranging under general vehicle motion. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Pasadena, California, April 2008.

[73] B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America*, 4:629–642, April 1987.

[74] D. Horner and O. Recent developments for an obstacle avoidance system for a small AUV. In *Proc. of the 7th IFAC Conf. On Control Applications in Marine Systems*, 2007.

[75] A. Howard, D. F. Wolf, and G. S. Sukhatme. Towards 3D mapping in large urban environments. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 419–424, Sendai, Japan, Sep 2004.

[76] T. Huntsberger, H. Aghazarian, A. Howard, and D. Trotz. Stereo visionâĂŞbased navigation for autonomous surface vessels. *J. of Field Robotics*, 28(1):3–18, 2011.

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

[78] S. Julier and J. Uhlmann. A new extension of the kalman filter to nonlinear systems. In *The Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls*, Orlando,Florida, 1997. SPIE. Multi Sensor Fusion, Tracking and Resource Management II.

[79] M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Incremental smoothing and mapping. *IEEE Trans. Robotics*, 24(6):1365–1378, Dec 2008.

[80] R.E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME - Journal of Basic Engineering*, 82D:35–45, March 1960.

[81] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, SGP '06, pages 61–70, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association.

[82] R. Kinderman and J. Snell. *Markov Random Fields and their Applications.* American Mathematical Society, Providence, RI, 1980.

[83] J. Kinsey, R. Eustice, and L. Whitcomb. A survey of underwater vehicle navigation: Recent advances and new challenges. In *IFAC Conf. on the Manoeuvring and Control of Marine Crafts*, Lisbon, Portugal, September 2006.

[84] S. Koenig and M. Likhachev. Incremental A*. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1539–1546, 2002.

[85] K. Konolige. Sparse sparse bundle adjustment. In *British Machine Vision Conference*, Aberystwyth, Wales, August 2010.

[86] K. Konolige and J. Bowman. Towards lifelong visual maps. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 1156–1163, 2009.

[87] K. Konolige, J. Bowman, J. D. Chen, P. Mihelich, M. Calonder, V. Lepetit, and P. Fua. View-based maps. *Intl. J. of Robotics Research*, 29(10), 2010.

[88] K. Konolige, G. Grisetti, R. Kümmerle, W. Burgard, B. Limketkai, and R. Vincent. Sparse pose adjustment for 2D mapping. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan, October 2010.

[89] S. Laine and T. Karras. Efficient sparse voxel octrees. In *Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games*, I3D '10, pages 55–63, New York, NY, USA, 2010. ACM.

[90] S. LaValle. Rapidly-exploring random trees: A new tool for path planning. Technical Report 98-11, Dept. of Computer Science, Iowa State University, October 1998.

[91] S. LaValle and J. Kuffner. Randomized kinodynamic planning. *Intl. J. of Robotics Research*, 20(5):378–400, May 2001.

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

[93] J. Leonard, J. How, S. Teller, M. Berger, S. Campbell, G. Fiore, L. Fletcher, E. Frazzoli, A. Huang, S. Karaman, O. Koch, Y. Kuwata, D. Moore, E. Olson, S. Peters, J. Teo, R. Truax, M. Walter, D. Barrett, A. Epstein, K. Maheloni, K. Moyer, T. Jones, R. Buckley, M. Antone, R. Galejs, S. Krishnamurthy, and J. Williams. A perception driven autonomous urban vehicle. *J. of Field Robotics*, 25(10):727–774, 2008.

[94] J. J. Leonard and H. F. Durrant-Whyte. *Directed Sonar Sensing for Mobile Robot Navigation.* Kluwer Academic Publisher, Boston, 1992.

[95] A. Leonessa, J. Mandello, Y. Morel, and M. Vidal. Design of a small, multipurpose, autonomous surface vessel. In *Proc. of the IEEE/MTS OCEANS Conference and Exhibition*, pages 544–550, 2003.

[96] F. Lu and E. Milios. Globally Constent Range Scan Alignment for Environment Mapping. *Autonomous Robots*, 4:333–349, April 1997.

[97] J. E. Manley. Evolution of the autonomous surface craft autocat. In *Proc. of the IEEE/MTS OCEANS Conference and Exhibition*, Providence, RI, 2000.

[98] J.E. Manley. Unmanned surface vehicles, 15 years of development. In *OCEANS 2008*, pages 1–4. IEEE, 2008.

[99] E. Mazor, A. Averbuch, Y. Bar-Shalom, and J. Dayan. Interacting multiple model methods in target tracking: a survey. *Aerospace and Electronic Systems, IEEE Transactions on*, 34(1):103–123, January 1998.

[100] P.Y. Mignotte, S. Reed, A. Cormack, and Y. Petillot. Automatic ship hull inspection - the detection of mine-like targets in sonar data using multi-CAD fusion and tracking technologies. In *Proceedings of Institute of Acoustics International Conference on the Detection and Classification of Underwater Targets*, Sep 2007.

[101] M. Montelermo, S. Thrun, and D. Koller nad B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proc. of the National Conference on Artificial Intelligence (AAAI-02)*, Edmonton, Canada, 2002.

[102] M. Montemerlo and S. Thrun. Large-scale robotic mapping of urban structures. In Marcelo Ang and Oussama Khatib, editors, *Experimental Robotics IX*, volume 21 of *Springer Tracks in Advanced Robotics*, pages 141–150. Springer Berlin / Heidelberg, 2006.

[103] D. Moore, A. Huang, M. Walter, E. Olson, L. Fletcher, J. Leonard, and S. Teller. Simultaneous local and global state estimation for robotic navigation. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2008.

[104] F. Moosman and C. Stiller. Velodyne SLAM. In *IEEE Intelligent Vehicles Symposium*, Baden-Baden, Germany, June 2011.

[105] H. P. Moravec and D. W. Cho. A Bayesean method for certainty grids. In *AAAI Spring Symposium on Robot Navigation*, pages 57–60, 1989.

[106] R. R. Murphy, E. Steimle, M. Hall, M. Lindemuth, D. Trejo, S. Hurlebaus, Z. Medina-Cetina, and D. Slocum. Robot-assisted bridge inspection after Hurricane Ike. In *IEEE Intl. Workshop on Safety, Security, and Rescue Robots (SSRR)*, Nov 2009.

[107] R. R. Murphy, E. Steimle, M. Hall, M. Lindemuth, D. Trejo, S. Hurlebaus, Z. Medina-Cetina, and D. Slocum. Robot-assisted bridge inspection. *Journal of Intelligent and Robotic Systems*, pages 1–19, 2011.

[108] NASA. CDDIS GNSS archive. ftp://cddis.nasa.gov/pub/gps/products, Jan 2011.

[109] U. S. Navy. The navy unmanned surface vehicle (USV) master plan. http://www.navy.mil/navydata/technology/usvmppr.pdf, July 2007.

[110] P. Newman, J. Leonard, and Richard Rikoski. Towards constant-time SLAM on an autonomous underwater vehicle using synthetic aperture sonar. In *Proc. of the Intl. Symp. of Robotics Research (ISRR)*, 2003.

[111] P. Newman, G. Sibley, M. Cummins, M. Smith, A. Harrison, C. Mei, and I. Reid. Toward life long mapping. In *Proc. of the Intl. Symp. of Robotics Research (ISRR)*, Lucerne, Switzerland, Aug 2009.

[112] K. Ni and F. Dellaert. Multi-level submap based slam using nested dissection. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2010.

[113] K. Ni, D. Steedly, and F. Dellaert. Out-of-core bundle adjustment for large-scale 3D reconstruction. In *Intl. Conf. on Computer Vision (ICCV)*, Rio De Janeiro, Brazil, October 2007.

[114] K. Ni, D. Steedly, and F. Dellaert. Tectonic SAM: Exact, out-of-core, submap-based SLAM. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 1678–1685, Rome, Italy, April 2007.

[115] A. Nüchter, K. Lingemann, J. Hertzberg, and S. Thrun. 6D SLAM with application in autonomous mine mapping. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 1998–2003, New Orleans, LA, USA, Apr 2004.

[116] Library of Congress. Harvard bridge, spanning charles river at massachusetts avenue, boston, suffolk county, ma. http://www.loc.gov/pictures/item/MA1293/.

[117] Office of Geographic, Executive Office of Energy Environmental Information (MassGIS), Commonwealth of Massachusetts, and Environmental Affairs. MassGIS USGS color ortho imagery. http://www.mass.gov/mgis/colororthos2008.htm, 2008.

[118] E. Olson. Real-time correlative scan matching. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 4387–4393, 2009.

[119] E. Olson, J. Leonard, and S. Teller. Fast iterative optimization of pose graphs with poor initial estimates. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 2262–2269, Orlando, FL, USA, 2006.

[120] Edwin Olson. *Robust and Efficient Robotic Mapping.* PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, June 2008.

[121] Edwin Olson, John Leonard, and Seth Teller. Robust range-only beacon localization. In *IEEE Autonomous Underwater Vehicles (AUV '04)*, Sebasco Estates, ME, June 2004.

[122] M. Paterson and F. Yao. Efficient binary space partitions for hidden-surface removal and solid modeling. *Discrete and Computational Geometry*, 5:485–503, 1990. 10.1007/BF02187806.

[123] M. Pauly, M. Gross, and L.P. Kobbelt. Efficient simplification of point-sampled surfaces. In *Visualization, 2002. VIS 2002. IEEE*, pages 163–170, Nov 2002.

[124] P. Pfaff, R. Triebel, and W. Burgard. An efficient extension to elevation maps for outdoor terrain mapping and loop closing. *Intl. J. of Robotics Research*, 26(2):217–230, 2007.

[125] K.S. Pratt, R.R. Murphy, J.L. Burke, J. Craighead, C. Griffin, and S. Stover. Use of tethered small unmanned aerial system at berkman plaza II collapse. In *IEEE Intl. Workshop on Safety, Security and Rescue Robotics*, pages 134–139, October 2008.

[126] D. Ribas, P. Ridao, J. Neira, and J. Tardos. SLAM using an imaging sonar for partially structured underwater environments. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2006.

[127] D. Ribas, P. Ridao, J.D. Tardós, and J. Neira. Underwater SLAM in man-made structured environments. *Journal of Field Robotics*, 25(11-12):898–921, 2008.

[128] G. N. Roberts and R. Sutton. Unmanned surface vehicles – game changing technology for naval operations. In *Advances in unmanned marine vehicles*, pages 311–359. Institution of Electrical Engineers, 2006.

[129] C. Roman. *Self Consistent Bathymetric Mapping from Robotic Vehicles in the Deep Ocean.* Phd thesis, Massachusetts Institute of Technology and Woods Hole Oceanographic Institution, June 2005.

[130] C. Roman and H. Singh. Improved vehicle based multibeam bathymetry using sub-mapping and SLAM. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2005.

[131] C. Roman and H. Singh. Consistency based error evaluation for deep sea bathymetric mapping with robotic vehicles. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Orlando, FL, May 2006.

[132] N. Roy, W. Burgard, D. Fox, and S. Thrun. Coastal navigation – robot motion with uncertainty. In *Proceedings of the AAAI Fall Symposium: Planning with POMDPs*, Orlando, Florida, October 1998.

[133] R.B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz. Towards 3D point cloud based object maps for household environments. *Robotics and Autonomous Systems*, pages 927–941, 2008.

[134] J. Ryde and H. Hu. 3D laser range scanner with hemispherical field of view for robot navigation. In *Proceedings of IEEE/ASME Advanced Intelligent Mechatronics International Conference on Mechatronics and Automation (AIM 2008)*, Xi'an, China, July 2008.

[135] J. Ryde and H. Hu. 3D mapping with multi-resolution occupied voxel lists. *Autonomous Robots*, 28(2), Feb 2010. url http://dx.doi.org/10.1007/s10514-009-9158-3.

[136] H. Samet. Implementing ray tracing with octrees and neighbor finding. *Computers and Graphics*, 13(4):445–460, 1989.

[137] P. Sereda, A. V. Bartroli, I. Serlie, and F. Gerritsen. Visualization of boundaries in volumetric data sets using LH histograms. In *IEEE Trans. on Visualization and Computer Graphics*, volume 12, March 2006.

[138] A. J. Shafer. Autonomous cooperation of heterogenous platforms for sea-based search tasks. Masters of engineering thesis, Massachusetts Institute of Technology, June 2008.

[139] R. Shekhar, E. Fayyad, R. Yagel, and J. F. Cornhill. Octree-based decimation of marching cubes surfaces. In *IEEE Visualization*, 1996.

[140] G. Sibley, C. Mei, I. Reid, and P. Newman. Vast-scale outdoor navigation using adaptive relative bundle adjustment. *Intl. J. of Robotics Research*, 29(8):958–980, July 2010.

[141] R. Sim, G. Dudek, and N. Roy. Online control policy optimization for minimizing map uncertainty during exploration. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 1758 – 1763, New Orleans, LA, April 2004.

[142] R. Smith, M. Self, and P. Cheeseman. A stochastic map for uncertain spatial relationships. In S.S. Lyengar and A. Elfes, editors, *Autonomous Mobile Robots*, pages 323–330. IEEE Computer Society Press, 1991.

[143] S. M. Smith and D. Kronen. Experimental results of an inexpensive short baseline acoustic positioning system for AUV navigation. In *Proc. of the IEEE/MTS OCEANS Conference and Exhibition*, volume 1, pages 714–720, Halifax, NS, Canada, Oct 1997.

[144] N. Snavely. *Scene Reconstruction and Visualization from Internet Photo Collections.* PhD thesis, University of Washington, 2008.

[145] N. Snavely, S. M. Seitz, and R. Szeliski. Skeletal graphs for efficient structure from motion. In *Computer Vision and Pattern Recognition*, 2008.

[146] A. Stenz. Map-based strategies for robot navigation in unknown environments. In *AAAI Spring Symposium on Planning with Incomplete Information for Robot Problems*, 1996.

[147] R. Stokey, T. Austin, B. Allen, N. Forrester, E. Gifford, R. Goldsborough, G. Packard M. Purcell, and C. von Alt. Very shallow water mine countermeasures using the REMUS AUV: a practical approach yielding accurate results. In *Proc. of the IEEE/MTS OCEANS Conference and Exhibition*, pages 149–156, 2001.

[148] W. Stone, N. Fairfield, and G. A. Kantor. Autonomous underwater vehicle navigation and proximity operations for deep phreatic thermal explorer (DEPTHX). In G. Griffiths and K. Collins, editors, *Masterclass in AUV Technology for Polar Science: Proceedings of an International Masterclass Held at the National Oceanography Centre, Southampton, UK, 28-39 March 2006*, London, March 2007. Society for Underwater Technology.

[149] S. Subramanian, X. Gong, J. N. Riggins D. J. Stilwell, and C. L. Wyatt. Shoreline mapping using an omni-directional camera for autonomous surface vehicle applications. In *Proc. of the IEEE/MTS OCEANS Conference and Exhibition*, Boston, MA, 2006.

[150] H. Surmann, A. Nüchter, and J. Hertzberg. An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments. *Journal of Robotics and Automous Systems (JRAS)*, 45:181–198, Dec 2003.

[151] S. Teller, M. Antone, Z. Bodnar, M. Bosse, S. Coorg, M. Jethwa, and N. Master. Calibrated, registered images of an extended urban area. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–813. IEEE, 2001.

[152] S. Thrun, W. Burgard, and D. Fox. A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, volume 1, pages 321–328, San Francisco, CA, USA, 2000.

[153] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics.* MIT Press, 2005.

[154] S. Thrun, Y. Liu, D. Kollar, A. Ng, Z. Ghahramani, and H. Durrant-White. Simultaneous localization and mapping with sparse extended information filters. *Intl. J. of Robotics Research*, 23(8):690–717, 2004.

[155] S. Thrun, S. Thayer, W. Whittaker, C. Baker, W. Burgard, D. Ferguson, D. Hähnel, M. Montemerlo, A. Morris, Z. Omohundro, C. Reverter, and W. Whittaker. Autonomous exploration and mapping of abandoned mines. *IEEE Robotics and Automation Magazine*, pages 79–91, December 2004.

[156] C. Urmson and R. Simmons. Approaches for heuristically biasing RRT growth. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, volume 2, pages 1178–1183, October 2003.

[157] J. Vaganay, J. Leonard, J. Curcio, and S. Wilcox. Experimental validation of moving long-baseline concept. In *AUV 2004 IEEE/OES*, pages 59—65, June 2004.

[158] M. Walter, R. Eustice, and J. J. Leonard. Exactly sparse extended information filters for feature-based SLAM. *Intl. J. of Robotics Research*, 26(4):335–359, April 2007.

[159] M.R. Walter, R.M. Eustice, and J.J. Leonard. A provably consistent method for imposing sparsity in feature-based SLAM information filters. In *Proc. of the Intl. Symp. of Robotics Research (ISRR)*, San Francisco, CA, October 2005. In Press.

[160] R. Westermann, L. Kobbelt, and T. Ertl. Real-time exploration of regular volume data by adaptive reconstruction of iso-surfaces. *The Visual Computer*, 15:100–111, 1999.

[161] L. Whitcomb, D. Yoerger, and H. Singh. Combined Doppler/LBL based navigation of underwater vehicles. In *Proc. of the Intl. Symp. on Unmanned Untethered Submersible Technology*, May 1999.

[162] J. Wilhelms and A. Van Gelder. Octrees for faster isosurface generation. *ACM Transactions on Graphics*, 11:201–227, July 1992.

[163] J. M. Wozencraft. Complete coastal mapping with airborne lidar. In *Proc. of the IEEE/MTS OCEANS Conference and Exhibition*, volume 2, pages 1194–1198, October 2002.

[164] J. Yuh. Design and control of autonomous underwater robots: A survey. *Autonomous Robots*, 8(1):7–24, 2000.

[165] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *Intl. Journal of Computer Vision*, 13(2):119–152, 1994.

[166] Z. Zhang, J. Wang, and H. Zha. Adaptive manifold learning. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 99, 2011.