Toward autonomous underwater mapping in partially structured 3D environments

by

Mark VanMiddlesworth

A.B. Harvard University (2011)

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

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2014

Author .....
Department of Electrical Engineering and Computer Science
January 31, 2014

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

Accepted by .....
Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

Toward autonomous underwater mapping in partially structured 3D environments

by

Mark VanMiddlesworth

Submitted to the Department of Electrical Engineering and Computer Science
on January 31, 2014, in partial fulfillment of the
requirements for the degree of
Master of Science

Abstract

Motivated by inspection of complex underwater environments, we have developed a system for multi-sensor SLAM utilizing both structured and unstructured environmental features. We present a system for deriving planar constraints from sonar data, and jointly optimizing the vehicle and plane positions as nodes in a factor graph. We also present a system for outlier rejection and smoothing of 3D sonar data, and for generating loop closure constraints based on the alignment of smoothed submaps. Our factor graph SLAM backend combines loop closure constraints from sonar data with detections of visual fiducial markers from camera imagery, and produces an online estimate of the full vehicle trajectory and landmark positions. We evaluate our technique on an inspection of a decomissioned aircraft carrier, as well as synthetic data and controlled indoor experiments, demonstrating improved trajectory estimates and reduced reprojection error in the final 3D map.

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

Acknowledgments

They say "it takes a village...," and sometimes I wonder what it would be like if my academic support network were an actual village. I like to think of it as a pre-Columbian settlement in, say, the Andes, which I imagine to be climatologically indistinguishable from Boston.

My research supervisor, John Leonard, has provided invaluable advice, mentoring, and access to an overwhelming array of research projects and other opportunities. He would be the village wise man or chief, in constant contact with everyone, always available for advice both personal and professional, identifying problems to be overcome in our hunting and gathering activities, and coordinating trade with nearby villages to ensure we had adequate supplies for the winter.

To my advisor Dana Yoerger, whose contagious energy sparked a passion for marine robotics, whose battle-tested wisdom guided me at land and at sea, I owe my entire career at MIT. I see him as the village shaman, dispensing sage advice while magically solving any problem with his beyond-mortal mastery of arcane technical systems.

Michael Kaess has been absolutely essential to my academic progress, and I imagine him as the village's technical wizard. He would be master of emerging technologies like metal tools and mechanical advantage, and would share his knowledge generously to ensure the success of the entire community.

Franz Hover, the chief of the village next door, has provided both long-term opportunities and day-to-day mentoring and advice. I am immensely grateful for his help, and have very much enjoyed our collaboration on both research and operations.

My family is always close to my heart, and their support would be crucial to surviving the harsh alpine environment (although the closeness might be suffocating if we were holed up in a tiny yurt made of animal skins all winter, with only a sooty fire for warmth and the village passtime of pebble-tossing for entertainment). My mother Diane would certainly stay up all night helping me study the various edible roots and practice animal-skinning technique for my rite of passage into manhood, a

twelve-day journey into the icy mountains with nothing but a flint knife and alpaca-skin blanket. Over meals of small game, my father Rex and I would discuss the finer points of glacier navigation, and occasionally get into heated debates about village politics and optimal public policy. My younger brother Paul, whom I look up to in both stature and spirit, would push me to travel farther and climb higher on our hunting expeditions. We would undoubtedly get ourselves into all sorts of trouble climbing the avalanche-prone slopes and crevasse-ridden glaciers, but Rex and Diane would ensure that we were competent and well-equipped enough to get ourselves out.

My girlfriend Natalie, a native of the colder climate, would turn out to be a strong, competent mountaineer (despite her claims to the contrary) and would help me acclimate with hot herbal tea and reminders to avoid frostbite.

My friends, my band, and my labmates would provide necessary psychological relief from the tedium of the long, dark winter months. Their advice and encouragement would motivate me to work hard each morning, and their laughter and high spirits would provide something to look forward at the end of long days.

I couldn't have done it without y'all. If you ever want to go to Peru, I think we'd make a pretty good village.

Contents

List of Figures

Chapter 1

Introduction

The world is not an arbitrary arrangement of points floating aimlessly in Euclidean space. Environments are structured, be it by natural or human forces; they contain repetitive patterns and geometric regularities. The perceptive circuits of the human brain handle this unconsciously: if we see a wall that is partially occluded by a tree, even the most hardened existentialist skeptic would be unlikely to insist that the wall has a tree-shaped hole in it.

Computer perception has made significant progress up to this point, considering its comparative naivete. Robots are able to autonomously map and explore the world with few, if any, implicit assumptions about their surroundings. In the case of occlusion, this means that most approaches avoid "filling in" the wall area occluded by the tree. This elimination of all but the minimum assumptions is mathematically and philosophically appealing, but may not represent the best adaptation to real-world conditions: the human visual cortex was not carved by Occam's razor. Like humans, robots should leverage environmental structure to improve their perceptive abilities.

This thesis addresses the problem of mapping *partially structured* environments. A partially structured environment contains both *unstructured* and *structured* elements. An *unstructured* element is an arbitrary shape that does not lend itself to a higher-level geometric representation. Examples include rocky seafloors, marine growth, and other complex shapes. A *structured* element, on the other hand, can easily be

simplified to a parametric equation such as a plane or cylinder. Concrete pilings, pipelines, and walls could all be considered structured elements.

It is important to note that what is commonly referred to as "structure" is as much a property of the perception and representation system as it is of the environment. A wall could be represented as a structured element, by the equation $ax + by + cz - d = 0$, or it could be represented as an unstructured element, by a collection of points $p_i = (x, y, z)$. The term "structured environment," as it is used here and in related research, more precisely refers to a structured *representation* of an environment.

In our case, a partially structured environment is an environment represented as a mix of unorganized point clouds and parametric equations. We present a system for Simultaneous Localization and Mapping (SLAM) using visual targets and 3D sonar data, improving upon prior work by using parametric surface approximations of noisy point clouds and explicitly tracking planar features in our optimization framework.

Our application scenario is underwater inspection, which exhibits many desirable features for developing a partially structured SLAM system: navigation by dead reckoning is difficult, the environment contains both structured and unstructured elements, and multiple sensing modalities (visual and acoustic) are necessary for complete inspection.

1.1 Underwater inspection

An increasing amount of infrastructure is being installed underwater for scientific research, aquaculture, energy, and defense applications. These installations, as with ships, are subject to bio-fouling and corrosion. Currently, shallow water inspection tasks (ship hulls, floating platforms, hydroelectric) are time consuming, expensive, and sometimes dangerous tasks performed manually by divers. Infrastructure too deep for human divers, such as oil wellheads or ocean science instrumentation, is often inspected by a tethered Remotely Operated Vehicle (ROV). This generally requires ship time, which is expensive, and is tedious for the human operator (considering that the vast majority of inspection tasks should never find anything out of the ordinary).

Figure 1-1: Propeller of the USS Churchill in drydock (U.S. Navy DNSD0409218)

Automating these tasks with an Autonomous Underwater Vehicle (AUV) could provide benefits in cost, safety, and effectiveness. They require little if any support infrastructure (e.g. ship time for an ROV) and fewer personnel. They can operate in dangerous scenarios without risking human life, and because they aren't tethered to a ship, are much more practical in crowded harbors. AUVs are able to operate for hours or, if docking is available, weeks or months without human intervention, providing an unprecedented of level of persistent monitoring.

1.1.1 Inspection targets

Underwater infrastructure is subject to environmental forces such as corrosion, biological growth, and abrasion by suspended particles, which greatly decreases its service life compared to land-based installations. During this service life they must be continually monitored to ensure that levels of fouling, loss of thickness, and deformation are within design parameters.

Ship hulls are exposed to biofouling, corrosion, and abrasion throughout the service life of the vessel. In addition, minor collisions or groundings can damage or deform the outer hull and running gear. These events are routine and generally do not render the ship inoperable; however, damaged areas must be carefully

monitored. The American Bureau of Shipping (ABS) recommends that ship hulls and flooded ballast tanks be inspected a minimum of once a year and requires a full dry-dock inspection every 24 to 36 months [4]. Dry docking is time consuming and expensive, and it can require a lengthy commute from the vessel's port of operation to a specialized dry dock facility. The goal of these inspections is to detect and monitor hull deformation, thinning due to corrosion, and the condition of the ship's anti-fouling coating [14].

Offshore platforms Deep water drilling and oil extraction are sensitive, high-tech operations performed in difficult and unpredictable environments. The environmental impact of a failed valve or collapsed scaffolding could be disastrous. Stationary infrastructure experiences bio-fouling at a much high rate than operational ship hulls, which dislodge some growth during transit. Additionally, anchored structures must withstand the force of waves and wind on the above-water elements, and of ocean currents on the submerged elements. Debris from the superstructure are commonly dislodged and fall to the seafloor [71], potentially damaging submerged elements in the process. Potential failure points such as seafloor anchors, structural scaffolding, and welds are common targets of inspection. In addition, submerged metals commonly require require cathodic protection in the form of "sacrificial" galvanic anodes, which slow corrosion of structural elements. The anode material is consumed in the electrochemical reaction and must therefore be inspected and replaced regularly.

Submerged pipelines are not subject to the physical stresses of platforms, but must withstand erosion of supporting sand or rock as well as gradual motion of the seafloor. A typical inspection program designed by the United States Department of the Interior Minerals Management Service requires full sonar-based evaluation of burial and spanning conditions, structural integrity, protrusions, and damage from external impacts at minimum once every 2 years [56].

Harbors are high-traffic areas with specialized infrastructure for loading and shipping, and there is a strong economic incentive to ensure efficient operation.

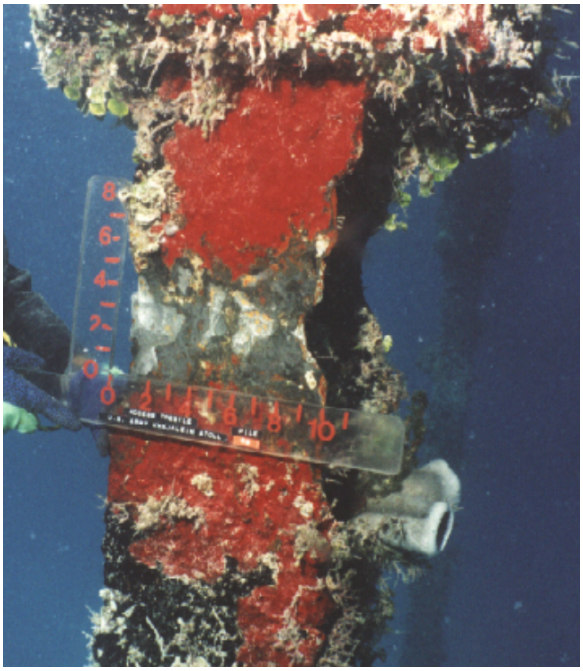


Figure 1-2: Inspection for corrosion on a steel pile (USDOT)

Harbor infrastructure is often made of wood, which softens and rots due to microbial action, and concrete, which cracks due to loading stress and temperature changes. Semi-submerged "dolphins" for mooring and berthing, structural pilings for piers and loading docks, and lock gates must all be monitored regularly [2].

Bridges are perhaps the most critical targets of underwater inspection from a safety standpoint. According to the U.S. Department of Transportation, 83% of the 600,000 bridges in the United States span waterways. Many of these bridges have support structures underwater, which wear quickly due to scouring by abrasive suspended particles carried by the currents or "prop wash" from passing ships. Flooding carries debris which pile up against bridge supports, and washes away supporting soil beneath bridge anchors. Due to their proximity to motor vehicles and other human infrastructure, bridges are often submerged in chemical-laden polluted water, increasing the rate of corrosion [13]. Given these factors, it may come as no surprise that most bridge collapses are due to failure of submerged elements [1]. A string of high-profile bridge collapses in the mid

(a) Schoharie Creek bridge collapse, Fort Hunter, NY, 1987 (USGS)

(b) US51 bridge collapse, Hatchie River, TN, 1989 (USDOT)

Figure 1-3: The Schoharie Creek bridge (a) and the Hatchie River bridge (b) both collapsed due to excessive scour of the support pier foundations, leading the U.S. Department of Transportation to develop improved guidelines for regular bridge inspection.

1980s led the U.S. Department of Transportation to create a set of standards for bridge inspection, including the planning, executing, and documenting the inspection of underwater inspection tasks [13].

1.1.2 Types of inspection

Which, if any, of these inspections could be performed by an AUV? Some, like ultrasound hull thickness measurements, require specialized tools. Others, like flooded member testing, can be done from the surface. AUVs would be most suited for tasks that can be done with standard underwater cameras and sonars. Within these tasks, there are widely varying requirements for coverage and precision. Inspecting welds or microfractures requires close-up imagery with millimeter-scale resolution, but detecting fallen debris, bent support beams, or unwanted attachments (bio, mines) could feasibly be done with centimeter or decimeter scale sonar.

The USDOT Federal Highway Administration defines the following types of inspection for bridges [13]:

Type I The *visual, tactile inspection*, also called a "swim-by" inspection, should be detailed enough to detect "obvious damage or deterioration." The target should

be examined visually or, if visibility is poor, by touch. Generally every element of the structure receives a full level I inspection.

Type II The *detailed inspection with partial cleaning* consists of removing biofouling in small patches to obtain detailed estimates of corrosion and deformation. Generally, a randomized representative subset of structural elements will receive a level II inspection.

Type III If the type II inspection reveals structural issues, it is followed up with a *highly detailed inspection*, usually involving non-destructive or partially-destructive testing.

These inspection tiers have been also adopted for harbors [43] and open water oil platforms [71].

We hope to automate level 1 surveys using a two-stage inspection process. First, the vehicle performs a long-range sonar inspection, or "safe distance survey," to construct a rough map of the inspection target. The safe distance survey consists of a simple lawnmower pattern at sufficient distance to avoid collision with the object, generally between 5 and 10 meters. Using the model created from the safe distance survey, the vehicle plans a trajectory for a detailed inspection at 1 to 5 meter range. The detailed inspection can be designed to use a camera, sonar, or both.

1.2 Partially structured environments

In general, robots tend to operate near humans and human-created infrastructure. As a result, most environments that a robot must map and navigate will exhibit some sort of regular structure. Using prior knowledge of the environments structure can greatly simplify the tasks of mapping and navigation. The most basic example of this is 2D laser mapping by indoor wheeled robots [85], which operates under the assumption that floors are horizontal and walls are vertical. This assumption reduces the complexity of mapping from 3D/6DOF to 2D/3DOF. Work in planar mapping attempts to detect flat surfaces in point cloud data, and use the extracted planes to

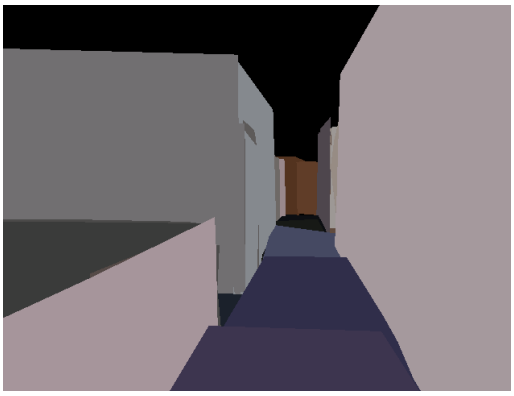


Figure 1-4: Human-occupied environments have predictable features, such as walls and floors, which can be used to improve localization and mapping. The higher-level representation is generally more compact, and produces more accurate map estimates for a given amount of sensor data. (Source: Hordur Johannsson [37], used with permission).

generate constraints on vehicle position (see Section 5.1.1 for examples and analysis of some of these techniques). At a higher level, 3D objects can serve as landmarks for mapping. Salas-Moreno et al. present a system for mapping based on object detection in 3D point clouds [82].

There are several advantages to mapping based on higher-level landmarks. First, these high level features can produce fuller maps from limited sensor data, or data with less overlap. They require only enough data to *detect* the object, such as one view of a chair, to reconstruct the full object in the final map. Second, in addition to operating on a lower quantity of data, these techniques may be able to operate on lower *quality* data. While a wall scanned with a noisy sensor will require the averaging of many measurements to produce a smooth result, a smooth plane could be extracted from the noisy data. Third, they can produce a more compact representation of the environment, by reducing redundancy (e.g. tracking chair locations rather than full point clouds for each chair) or replacing dense point cloud data with parametric surfaces. Finally, these techniques may produce more "meaningful" maps if the features are well-matched to the usage of the final map. For example, a map used for ray-tracing will be more useful if closed surfaces are explicitly represented, instead of being interpolated from an unorganized point cloud.

However, it is not immediately apparent what structure can be leveraged in an underwater environment. They are not occupied by humans, so we can't track features based on their human affordances, such as walls or chairs. However, we can make some assumptions based on what the infrastructure is used for. First, because these structures are submerged, they are generally composed of smooth surfaces. Many underwater structures are designed to be hydrodynamic, and therefore exhibit relatively slow changes in gradient. Even structures for which hydrodynamics is not a consideration are usually built with smooth, solid faces unless the functionality requires otherwise. Second, most of these structures are watertight, for the somewhat obvious reason that they are specifically designed to be a barrier against water. Third, we can make general assumptions based on human design patterns: anthropogenic structures are often built using straight lines and flat faces, unless the application specifically requires otherwise (such as a dam curved inward to increase strength).

In addition to geometric structure, underwater infrastructure may exhibit distinctive visual features. Components in offshore infrastructure must be labeled for inspection and maintenance, for example with a serial number, part ID, or bar code [3]. Many currently deployed tags are high-contrast corrosion-resistant labels (e.g. `http://www.aquasign.com`), which could be recognized by a camera and used as a high accuracy zero-drift waypoint for navigation and mapping. Additionally, the cost of installing robot-specific markers, such as visual fiducial markers, is small compared to the cost of regular manual inspection .

We want to design a system to take advantage of both geometric and visual regularity in the environment.

1.3 Requirements

Full coverage of the target structure is a requirement of most level I inspection tasks. This entails requirements for both planning and navigation. We use existing techniques from coverage planning to generate a trajectory that covers the full structure. In many situations, it is sufficient for the vehicle to have an accurate estimate of its

current position only. In this case, filtering solutions such as EKF or particle filtering, which can run efficiently in real time, are good solutions. However, inspection tasks require that we have an accurate estimate of the entire vehicle trajectory, not just the current pose. This allows us to verify that the entire inspection target has been covered.

We also requires that the robot be able to localize itself to accurately follow the trajectory. Because it is important that the vehicle follow the inspection path as closely as possible, we require these trajectory estimates *online*, rather than in post-processing, as is common with many autonomous underwater operations. Therefore, our system must be designed and implemented for computational efficiency, to be able to provide real-time position estimates on the low-power computers typical of AUVs.

Because many inspection tasks are in water with poor visibility, we require that the vehicle be able to use multiple sensors. Vehicles are commonly equipped with a camera and sonar, and it is desirable to have the ability to generate loop closures from either sensor.

1.4 Contributions

We have developed an algorithm for underwater SLAM in partially structured environments, using sonar and camera data. Our main contributions are:

1. A denoising and parametric surface modeling technique for improved alignment of sonar submaps.

2. A technique for tracking sonar submaps and generating pose-relative constraints based on submap alignment.

3. A framework for visual and acoustic SLAM using factor graphs.

4. Experiments with localization and mapping using sonar and camera data in a swimming pool and on the hull of a decommissioned aircraft carrier.

The remainder of this thesis is structured as follows:

Chapter 2 introduces the vehicle and the specifics of our underwater inspection task.

Chapter 3 introduces our framework for planning inspection tasks, integrating sensor data, and constructing and optimizing a factor graph to estimate vehicle trajectory.

Chapter 4 describes our technique for generating loop closures from submap alignment, including the creation, smoothing, and alignment of submaps from noisy sonar data, and presents evaluation from an inspection task on the *USS Saratoga*, a decomissioned aircraft carrier.

Chapter 5 describes out technique for generating loop closures from structured elements, including detection of visual fiducial markers from camera imagery and extraction of planar segments from sonar submaps, and presents experimental results from an indoor structured environment.

Chapter 6 summarizes our work and presents directions for future research.

Chapter 2

Inspection of Partially Structured Underwater Environments

This chapter presents the challenges in underwater inspection and existing solutions that have informed this work.

2.1 Challenges in underwater navigation

A fundamental challenge of underwater operations is the lack of global positioning information, because water quickly blocks electromagnetic radiation, from visible light through the frequencies used by satellite-based GPS. As a result, many underwater navigation techniques are based on highly accurate dead reckoning. Reduced visibility, strong currents, and the relative lack of distinctive visual or topographical features further complicate underwater localization. However, sound travels rapidly and with low attenuation due to water's relatively high ratio of stiffness (bulk modulus) to density. As a result, acoustic sensors provide some of the best tools for underwater localization.

The sensors commonly used in underwater operations are:

Control inputs The simplest form of navigation is to simply assume an ideal vehicle model that moves exactly as commanded. For land robots with wheel encoders, this can be a fairly accurate solution if there is no wheel slip. For underwater

applications, modeling the relationship between control inputs (such as propeller RPMs) and vehicle motion is more complicated, due to highly nonlinear thruster response and hydrodynamic forces such as drag.

Magnetic compass One of the oldest navigation sensors is still one of the most useful for underwater navigation. Compasses provide an absolute heading reference, which can be combined with an estimate of vehicle velocity or distance traveled to produce a position estimate. A typical compass used in underwater operations provides heading estimates at 1-2Hz with an accuracy of $1-10^{\circ}$ [97]. Unfortunately, magnetic compasses are subject to magnetic interference near large metal structures, making them unsuitable for close inspection of most marine infrastructure.

Pressure sensor A depth gauge is used almost universally in underwater applications. Because water density changes with temperature and salinity, the most accurate systems combine a reading from a pressure sensor with a measured or predetermined profile of temperature and salinity in the water column [19]. A properly calibrated pressure sensor produces zero-drift measurements at 1Hz with accuracy on the order of .01% [97].

Gyroscope A gyroscope provides an orientation estimate by measuring rotation using the conservation of angular momentum. The simplest form is a rotating weighted disc, which resists rotation in any direction other than the spin axis. Combining measurements from multiple orthogonal gyroscopes allows full 3DOF orientation estimate. The most common gyroscope in modern applications is a Micro Electro-Mechanical System (MEMS) gyroscope, which uses a the vibration of a mechanical element to measure angular velocity. MEMS gyroscopes are small enough to fit onto an integrated circuit, and are commonly found in cell phones and other consumer electronics. In aerospace, defense, and underwater applications, more expensive and accurate units are used, based on acceleration-induced phase difference between two light beams traveling in opposite directions around a loop. The two most common implementations

are fiber optic gyroscopes (FOG) and ring laser gyroscopes (RLG). These gyroscopes are accurate enough to detect the rotation of the earth, and can therefore be used for absolute heading reference using gyrocompassing. Gyrocompassing has the added benefit of finding true north, rather than magnetic north.

Gyroscopes tend to exhibit low "jitter" but high drift. A MEMS gyrocompass exhibits precision of under one degree and drift on the order of 10°/hr [97], whereas a high quality RLG has precision of hundredths of a degree and drift of less than .5°/hr [96].

Accelerometer Accelerometers measure linear acceleration, which can be integrated once to find linear velocity, or twice to find position. An array of three orthogonal accelerometers is commonly used to produce 3D position and velocity estimates. Because position is calculated as a double integral, small amounts of noise in the initial acceleration measurement will be amplified and cause large amounts of drift in the position estimate. An uncalibrated stationary accelerometer will measure an acceleration of 9.8 m/s^2 in the direction opposite earth's gravity. Measurement of earth-relative position therefore requires accurate estimates of the direction and magnitude of earth's gravity, which are subtracted out of the estimated acceleration.

Accelerometers are commonly combined with gyroscopes in an *Inertial Measurement Unit*, or IMU. The full system of estimating position and heading using an IMU is called an *Inertial Navigation System* (INS) or *Attitude and Heading Reference System* (AHRS). The heading estimate produced by an AHRS/INS has been shown to be roughly twice as accurate as the estimate produced by an uncorrected gyrocompass [28]. However, the position estimate of these units exhibits drift on the order of several kilometers per hour [52].

Doppler Velocity Log A Doppler Velocity Log, or DVL, uses Doppler shift in acoustic backscatter to measure the relative velocity of the water column or seafloor. DVL units commonly have four highly directional beams measuring the intensity and Doppler shift along a series of range bins. AUV applications

typically use a high frequency (1200 kHz) unit with a range of 30m [97], mounted on the bottom of the vehicle and pointed at the seafloor. For each beam, the bin with the highest amplitude can be used to estimate the distance to the seafloor, and the Doppler shift of this bin can be used to find the component of seafloor-relative sensor velocity along the vector of the beam. Combining the velocity components from at least three beams gives an estimate of the full 3DOF sensor velocity relative to the seafloor; four beams are typically used.

A DVL measuring only the water velocity is called an Acoustic Doppler Current Profiler, or ADCP, and is commonly used to estimate currents for scientific research. The current-measuring ability of the DVL/ADCP has also been been used on an AUV to estimate water current and vehicle position during the vehicle's descent phase, before bottom lock is acquired [84].

A typical navigation-grade 1200 kHz DVL has zero-mean noise with a standard deviation of .2% of measured velocity [97]. Position estimates generated by integrating velocity will therefore drift in a random walk, with the magnitude of position error dependent on vehicle velocity and attitude. Because each beam provides a range to the seafloor, the DVL can also be used as a zero-drift acoustic altimeter.

Acoustic Ranging It is often desirable to find the global position of the vehicle, commonly accomplished by acoustic ranging to objects at known positions. One such system is Long Baseline (LBL) acoustic positioning, in which a network of transponders is deployed around the vehicle's operating area. The vehicle can triangulate its position using ranges from three transponders; additional transponders are generally deployed to increase redundancy and accuracy [29]. To provide absolute position reference, the geo-referenced position of each beacon must be accurately calibrated. In many applications, local position is sufficient, which requires only the relative positions of the transponders. Once installed, the transponder array can operate for long periods of time and provide zero-drift navigation for multiple vehicles in the work area. Therefore, LBL

is commonly used as sites of long term research and monitoring.

A typical long-range LBL unit operates at 12kHz, has a range of up to 10km, and zero-mean range-dependent position error up to 10m [34]. For short ranges, a 300kHz LBL array can provide centimeter-precision positioning accuracy up to about 100m [97]. LBL error is heavily dependent on the calibration error of the beacon positions. Additionally, generating accurate range information from round trip time requires knowledge of the sound speed profile of the water, which is primarily governed by temperature and salinity. In shallow areas, particularly the first tens of meters of the ocean water column, temperature and salinity change rapidly with depth, and may also vary over short time periods due to wave action. LBL is also susceptible to multipath interference due to reflections, and is therefore less accurate in shallow environments, near the surface, or under ice [52].

When it is impractical to install an LBL array, a ship-mounted system, combined with the ship's GPS, can provide accurate globally referenced position. Short Baseline (SBL) localization uses a transponder array mounted along the length of the ship, or hung over the sides. Position is then triangulated as in an LBL system. More commonly used is Ultrashort Baseline, or USBL, in which multiple transducers are mounted in a single transceiver unit. Rather than triangulation used in LBL and SBL systems, USBL measures range and bearing from the transceiver. Range is detected using round trip time from the ship mounted transceiver, as with a single LBL beacon. Bearing is calculated using the phase difference between the transducers.

Ship-mounted systems are more quickly deployed than LBL units, and can provide a similar level of accuracy when the target vehicle is in a narrow working window below the ship. USBL error is dominated by error in the measured bearing, which, although typically under 1°, leads to increasing xyz localization error with increasing distance [83] [89]. Because they require a ship to follow the vehicle, they can be inconvenient for long term deployments. As with all

acoustic ranging systems, USBL is susceptible to errors in the estimated sound speed profile.

Other environmental sensors Other sensors commonly mounted on AUVs include cameras and sonar units. These can provide relative position estimation by tracking objects between subsequent frames, and estimating vehicle motion using an assumption of stationary objects. This visual odometry requires the ability to extract accurate features from camera or sonar images. SIFT [55] [54] are similar gradient-based features are commonly used in camera imagery. The Normal Distribution Transform has been used to align frames from an imaging sonar to aid in AUV navigation [44].

However, as with other incremental measurements, visual and acoustic odometry do not correct for long term navigation drift. When a prior map is available, features from the map can be used to provide zero-drift global position estimates.

These odometry and localization techniques are not limited to cameras and sonar. Other environmental sensors mounted that have been used on AUVs include sheet lasers [75], fluorometers [15], mass spectrometers [36], optical backscatter sensors [100], and magnetometers [88]. If a prior map of sufficient resolution is available, these environmental sensors can be used to navigate within the map.

2.2 Approaches to underwater navigation

The sensors described above are rarely used in isolation; more commonly, they are combined in a multi-sensor estimation framework to provide an improved position estimate. This is particularly beneficial when the sensors have different noise characteristics. A comprehensive overview of AUV navigation techniques can be found in [52]; a short summary is below.

Dead reckoning Dead reckoning is a general term used to refer to navigation us-

ing local measurements projected from a known global position. Any of the relative sensors described above – command inputs, accelerometer, gyroscope, visual odometry – could be considered a dead reckoning solution when used in isolation. In general, combining multiple sensors with uncorrelated noise provides greater accuracy than any individual sensor. In underwater applications, the most common dead reckoning configuration is a DVL and IMU supplemented by a compass and depth sensor. These sensor measurements are generally combined in a filtering framework based on the Kalman filter [42].

Dead reckoning may also be used with other relative position measurements, such as visual odometry or acoustic scan-matching. A well-tuned system using scan-matching from an imaging sonar has been shown to have accuracy on the order of .02% of distance traveled [66].

By definition, dead reckoning lacks an absolute position reference to correct accumulated error. Therefore, the error in dead reckoning estimates will inevitably tend to increase over time.

Acoustic Beacons + Dead Reckoning To correct accumulated error in the dead reckoning position estimate, it is common to combine the DVL+IMU system with a zero-drift acoustic positioning system such as LBL or USBL. This leverages the complementary noise characteristics of dead reckoning, which has low jitter and high drift, and LBL/USBL, which has high jitter and low drift, to produce a significantly improved position estimate. For real time operation, the time-of-flight measurements can be added directly to the filtering framework, with the low-jitter dead reckoning estimate used for outlier rejection on the acoustic position estimates [91]. If navigation is corrected in post-processing, the time domain error characteristics of each sensor can be removed in the frequency domain, and the sensors combined with a complementary filter [97].

Prior Map When a prior map is available, it is possible to localize within this map using available sensors. Gravitational [52], magnetic [28], and bathymetric [38] data have all been used for underwater localization. The localization accuracy

is limited by the spatial resolution of the signal. For example, bathymetric localization will be inaccurate on a flat, featureless bottom. This technique also requires creating a prior map of sufficient resolution, and updating the map to reflect changes in the environment.

SLAM When a prior map is not available, it is possible to navigate relative to natural landmarks using SLAM. See Section 2.4.1 for an overview of prior work in this area.

Of these techniques, only SLAM meets our requirements for high-accuracy positioning near underwater infrastructure. SLAM is well-suited for underwater inspections because it is more accurate than dead reckoning and requires less infrastructure than global acoustic positioning. Additionally, many inspection environments are ill-suited to LBL/USBL acoustic positioning due to multipath interference and the difficulty of installing and maintaining infrastructure in a crowded harbor environment.

2.3 Inspection with the HAUV

The MIT-Bluefin Hovering AUV (HAUV) is designed for shallow-water inspection tasks in complex environments. These inspections are generally performed by divers or ROVs, and would be difficult if not impossible to accomplish with most current-generating AUVs. To create an AUV for complex inspection posed challenges in *vehicle design*, *visual-acoustic mapping*, and *coverage planning*. Prior work of our collaborators in each of these areas is presented below.

2.3.1 Vehicle design

The physical design of most current-generation AUVs is designed primarily for efficient bathymetric mapping and photo surveys, which have vastly different requirements than the inspection of underwater infrastructure. In particular, complex in-

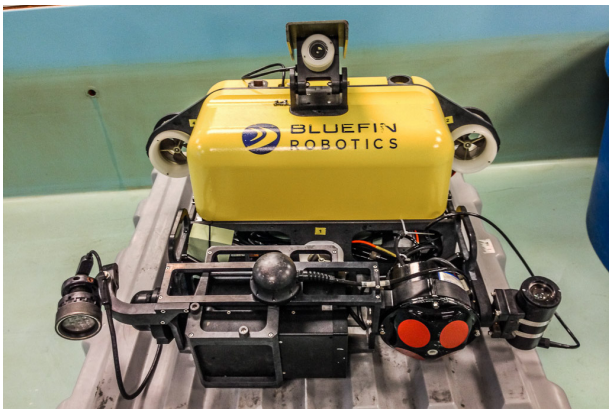


Figure 2-1: Bluefin Hovering Autonomous Underwater Vehicle with Soundmetrics DIDSON sonar.

spection requires greater *maneuverability* and *sensor coverage* than provided by the current generation of AUVs.

Maneuverability Most AUVs designed for long-range or deep water operations have a torpedo shape to reduce hydrodynamic drag, thereby reducing the volume and weight of the propulsion and battery systems. Cornering is accomplished using actuated fins, or by changing the angle of the main rear thruster. Vertical motion is primarily accomplished with buoyancy control, with small changes possible through steering. As a result, tight cornering and arbitrary depth changes are difficult with this form factor. Additionally, the torpedo shape is relatively unstable in the roll direction. Roll stability is particularly important for multibeam mapping operations, where small roll errors translate into large bathymetry errors on the outer edges of the sonar swath. "Sail" shape and dual-hull AUVs increase roll stability by increasing the distance between center of buoyancy and center of mass. The use of differential drive allows for tighter cornering, which allows a tighter spacing between track lines in a bathymetric survey.

On the other end of the spectrum, the common "box" form factor of ROVs increases maneuverability and stability at the expense of increased power and thrust requirements due to hydrodynamic drag. These almost universally em-

ploy differential drive, with thrusters positioned to allow full control of position and heading.

The HAUV was built with the maneuverability of an ROV, with five thrusters allowing arbitrary motion in x, y, z, and heading. The negatively buoyant battery and pressure housing are low in the vehicle, and the flotation is high, providing passive pitch and roll stabilization. Additionally, the placement of the two vertical thrusters allows active correction of vehicle pitch to maintain a level platform for mapping.

Sensor Coverage AUVs designed for mapping typically have downward-facing bathymetric mapping sonars, forward-looking imaging sonars, or sidescan sonars mounted on the underside of the vehicle. Some vehicles have been modified or designed with upward-facing sonar and DVL units for mapping the underside of sea ice [48] [12].

The HAUV carries a DIDSON multibeam sonar, detailed in Section 4.1.1, as well as two underwater cameras and a blue-green LED lighting array. The DIDSON can be used as either a narrow-beam mapping sonar or a wide-beam imaging sonar.

Complex inspection requires the ability to look up, down, or forward, often over the course of a single inspection. To this end, the HAUV was designed with a rotating instrument basket to hold the DVL, sonar, and camera. This allows the DVL to be aimed downward in the traditional bottom lock configuration for seafloor-relative positioning, or to be aimed at the inspection target for target-relative positioning.

The sonar is mounted on a 90° tilt mount within the front basket. It can be held at a fixed position or it can be swept in an arc independent of the DVL. The fixed position is generally used for imaging sonar or standard lawnmower surveys with profiling sonar, while the sweep is used for close-range inspection with profiling sonar.

For navigation, the HAUV is equipped with the standard sensor package of DVL, IMU, and pressure sensor. The IMU is a Honeywell HG1700 Ring Laser Gyro, which gives heading estimates with a bias of 2°/hr, and zero-drift pitch and roll measurements with an accuracy of 0.1°/hr. The DVL is a RDI Workhorse operating at 1200 kHz, with an accuracy of .2% of vehicle velocity. The pressure sensor has an accuracy of .01% with zero bias. [94] A compass is not used because the vehicle is expected to operate near large metal structures such as ship hulls, which create magnetic interference.

The vehicle's main processor performs sensor fusion, status monitoring, and waypoint-based position control in the dead reckoning frame. The estimated position, along the sensor data, is passed off to a second computer through a backseat driver interface. The backseat computer can either be onboard or remotely attached over the fiber-optic tether.

2.3.2 Non-complex area SLAM

Our collaborators have created a SLAM system for surveying the *non-complex* areas of the hull, which can be covered with a standard lawnmower survey. For the non-complex areas, the sonar, DVL, and underwater camera are continually aimed at the hull using the rotating front basket. The DIDSON is operated in imaging mode, with a low grazing angle, aimed laterally across the hull. At each time step, the vehicle's uncertainty is compared to the cost of reducing uncertainty by revisiting a previous location for a loop closure in sonar or camera data. If the gain of returning for a loop closure is greater than the cost, the vehicle pauses its survey and returns to the previous location; this is a variant of active SLAM which the authors refer to as Perception Driven Navigation [46] [45]. The expected information gain from a visual loop closure is estimated using visual saliency [47], which roughly corresponds to the "distinctiveness" of an area, and therefore the probability of generating a reliable loop closure from visual features. Using a piecewise planar assumption, they track features in the imaging sonar, and align them using the Normal Distribution Transform. They optimize the vehicle trajectory and plane locations in real-time using an incremental

factor graph solver [65].

2.3.3 Coverage planning

The remainder of the hull, including the propellers, driveshafts, and rudders, we refer to as the *complex area*. The non-complex areas can be completely surveyed using a standard trackline survey, assuming navigation drift is accounted for. This can be done either by following a preplanned survey and correcting navigation through SLAM, or by adaptively replanning based on updated position estimates [70]. However, fully surveying the complex areas is non-trivial even in the case of perfect navigation. The survey pattern must inspect each surface, taking into account occlusion, and must follow a collision-free path to do so. The problem of generating a trajectory that ensures full coverage of the structure is known as coverage planning, and consists of multiple sub-problems, primarily generating subsets of viewpoints that cover the entire structure, selecting the best subset, and planning a collision-free trajectory between the selected waypoints. The optimal solution can only be found through exhaustive search, which is computationally infeasible. Our colleagues have developed a sampling-based coverage planning algorithm [20] [21] using redundant roadmaps to efficiently generate multiple full-coverage waypoint sets and select the lowest-cost subset of waypoints that guarantees full coverage. They have also developed an iterative smoothing technique [22] to improve the resultant trajectories without loss of coverage.

This prior work has assumed perfect localization of the vehicle. These algorithms could potentially be modified to tolerate bounded navigation error in both collision avoidance, e.g. by inflating the vehicle size by the amount of expected position error, and sensor coverage, e.g. by reducing the field-of-view of the simulated sensor during the coverage planning waypoint generation. However, even a small amount of heading error can quickly lead to position error much larger than the vehicle, or even the sonar field of view, meaning that these ad-hoc solutions would quickly become intractable.

Full-coverage surveys of complex area require a navigation solution to correct accumulated drift in the dead reckoning estimate. In non-complex areas, this is

provided by a SLAM system assuming local planarity, however this assumption does not hold for complex areas. There is a large body of relevant prior work in underwater mapping, 3D SLAM, and surface modeling, however, we did not find a system that met our specific requirements for close-range complex-area inspection. We therefore created a SLAM system for mapping these complex areas that is: 1.) efficient enough for online operation, 2.) able to utilize both visual and acoustic loop closures, 3.) capable of using environmental structure in the form of planes extracted from sonar point clouds, 4.) robust to sonar error.

2.4 Existing Solutions

This paper builds upon a large body of prior research in underwater simultaneous localization and mapping (SLAM), 3D mapping, and dense point cloud alignment.

2.4.1 SLAM

The goal of SLAM is to correct for drift in the vehicle's dead reckoning by using repeated observations of static landmarks in the environment. There are two broad families of approaches: *filtering* and *smoothing*. Both approaches generally assume Gaussian process and measurement error models.

Filtering approaches track the robot's current pose by incrementally adding dead reckoning and loop closure constraints. Because constraints are added incrementally, this approach is naturally suited to real-time operation. The extended information filter (EIF) [87], in which the normal distribution is parameterized in terms of its information vector and information matrix rather than its mean and covariance, has a sparse structure which enables efficient computation.

Barkby et al. [7] used a particle filter along with a bathymetric sonar to produce a 2.5D map of the seafloor in real time. The extended Kalman filter (EKF) has been applied to imaging sonar data [74], and forward-looking sonar [26] collected by AUVs. Walter et al. [93] used a filtering approach to survey an underwater structure using features manually extracted from an imaging sonar.

A disadvantage of filtering approaches is that they estimate only the current vehicle pose. Because information from loop closure constraints is not back-propagated to correct previous pose estimates, these approaches do not provide an accurate estimate of the entire vehicle trajectory. This is particularly problematic when adding constraints from large loop closures, which produces discontinuities in the estimated vehicle path.

Smoothing approaches also include all past poses into the optimization. Exploiting the fact that the information matrix is exactly sparse in view-based SLAM, Eustice et al. [23] applied the information filtering approach to camera data from the RMS Titanic, producing a 6-DOF trajectory estimate. Dellaert and Kaess [17] formulate the SLAM problem as a bipartite factor graph, and provide an efficient solution by smoothing and mapping (SAM). Incremental smoothing and mapping (iSAM) [41] incrementalizes the matrix factorization to efficiently integrate new constraints without refactoring the information matrix.

In the underwater domain, pose graphs have been shown to produce more consistent maps due to their ability to correct prior navigation error and relinearize around the corrected trajectory. Beall et al. [8] used an offline pose-graph based smoothing approach to estimate a full 6-DOF trajectory in a large-scale underwater photo survey. Kunz and Singh [49] applied offline pose graph optimization to visual and sonar data. Pose graphs have been used for real-time mapping of a locally planar complex structures such as ship hulls [32].

In bathymetric and photomosaicing applications, a 2.5-dimensional representation of the environment (depth map) is sufficient, but complex environments require a full 3D representation. Fairfield et al. [24] use evidence grids inside a particle filter to perform real-time 3D mapping of a sinkhole with an imaging sonar.

2.4.2 Submap SLAM

Submap alignment requires generation of loop closure constraints, which, in visual SLAM, are commonly derived using viewpoint-invariant visual features such as SIFT.

However, bathymetric and profiling sonars generally do not produce easily iden-

tifiable viewpoint-invariant features. If the vehicle dead reckoning is accurate over short time periods, as is the case with most IMU and DVL based systems, the sonar data can be aggregated into "submaps" for improved matching.

For terrestrial mapping, point cloud-based approaches [63] [11], which use iterative closest point (ICP) to align submaps, have been applied to depth camera and scanning laser data.

In the underwater domain, McVicker et al. [57] created 2D maps of flooded archaeological sites using a small ROV equipped with a 360-degree scanning sonar. Because the ROV does not have an inertial sensor to produce dead reckoning estimates, sonar data was collected while the vehicle was stationary. The authors note that standard scan-matching techniques developed for laser data perform poorly when applied to noisy sonar data. They instead develop a denoising and alignment pipeline based on particle filter localization [86].

Roman and Singh have used submap alignment for bathymetric (2.5D) mapping with an AUV and multibeam sonar. They use cross-correlation [77] and ICP [76] to provide constraints for EKF SLAM. These are perhaps the most closely related to the work presented here. However, their filtering approach does not optimize the full vehicle trajectory, and is susceptible to linearization errors when initial position estimates deviate from ground truth.

Chapter 3

A Framework for Visual-Acoustic Factor Graph SLAM

This chapter describes our framework for autonomous ship hull inspection using factor graph SLAM. The core of our mapping framework is iSAM [41], which allows efficient online optimization for factor graph SLAM. Our system runs in real time during field operations, and at approximately 4x speedup on a standard consumer laptop. The system consists of various modules for pre-processing sensor data, storing and aligning submaps, and constructing and optimizing the factor graph. The sensor front-end modules are decoupled from the mapping back-end using Lightweight Communications and Marshalling (LCM) [33], a low-overhead publish-and-subscribe architecture for network communications. This decoupling enforces a standard interface and allows us to easily substitute or add new sensors.

3.1 Problem formulation

We formulate poses and constraints into a *factor graph*, a bipartite graph in which nodes represent vehicle and landmark poses, and factors represent constraints on vehicle and landmark poses. Our factor graph will have three types of nodes: vehicle poses X, landmark poses L, and planes E.

Vehicle poses are $X = \{\mathbf{x}_t\}$, for time $t \in 0 \dots T$. Poses are represented in Cartesian

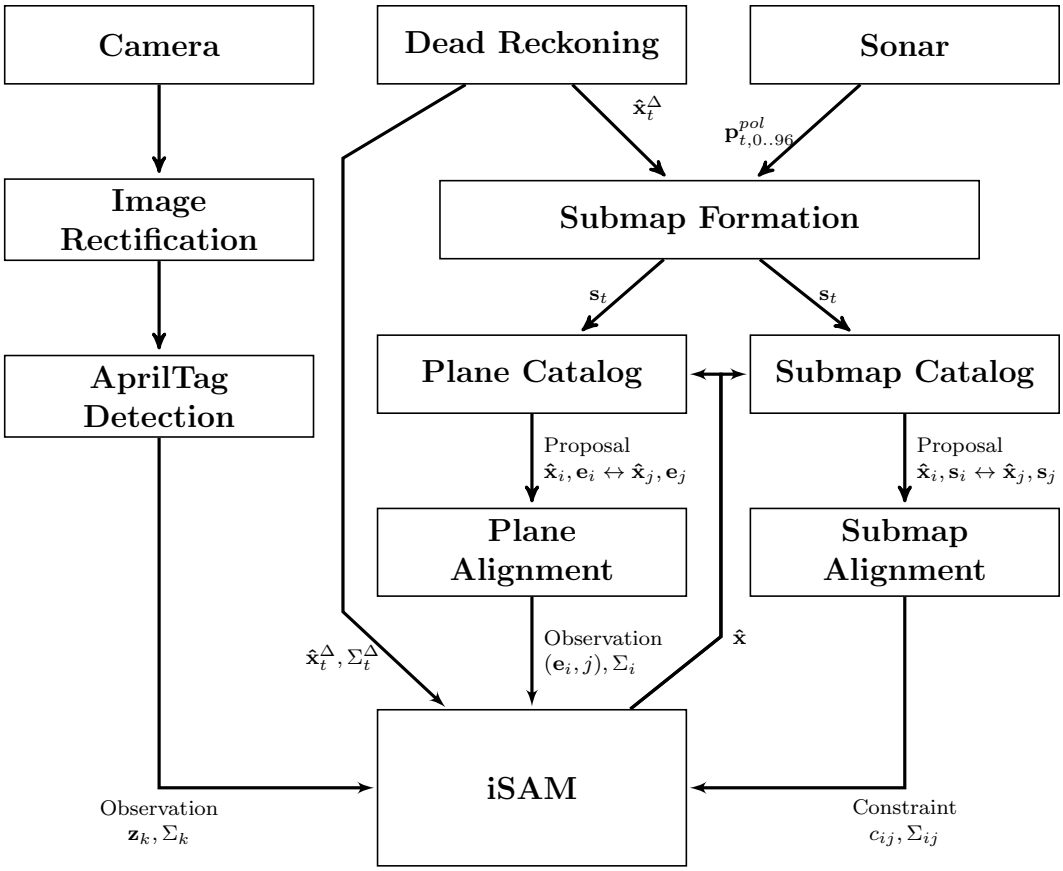


Figure 3-1: Architecture of visual-acoustic SLAM with planar constraints

coordinates as $\mathbf{x}_t = [x, y, z, \psi, \theta, \phi]$, using standard Euler angles for yaw, pitch, and roll. [1] The choice of origin does not affect our position estimates, although due to Euler angle singularities it is more convenient to choose a coordinate system that avoids high pitch and roll angles. We initialize x, y, z, and heading ψ to 0 at the first robot pose, with pitch θ and roll ψ initialized relative to gravity using the estimate produced by the IMU.

We refer to the estimated vehicle poses as $\hat{\mathbf{x}}_t$ to distinguish them from ground-truth poses $\mathbf{x}_t$.

We model the vehicle and landmark positions as a joint probability distribution $P(X, C, Z, L, E, F, U)$, where:

[1] Although we use Euler angles in formulating poses and constraints, they are represented internally as quaternions in iSAM

X represents vehicle positions.

C represents constraints between nonconsecutive vehicle poses.

Z represents 6DOF landmark positions.

L represents constraints between vehicle positions and 6DOF landmark positions.

D represents planar landmark positions.

E represents constraints between vehicle poses and planar landmark positions.

U represents constraints between consecutive vehicle poses from dead reckoning estimates.

We can directly write the joint probability as:

$$\begin{aligned} P(X, C, Z, L, D, E, U) = P(x_0) & \prod_{i=1}^{|X|} P(x_i|x_{i-1}, u_i) \\ & \prod_{j=1}^{|Z|} P(z_j) \prod_{k=1}^{|L|} P(l_k|x_{a_k}, z_{b_k}) \\ & \prod_{n=1}^{|D|} P(d_n) \prod_{o=1}^{|E|} P(e_o|x_{a_o}, d_{b_o}) \\ & \prod_{m=1}^{|C|} P(c_m|x_{a_m}, x_{b_m}) \end{aligned} \tag{3.1}$$

We find the maximum *a posteriori* (MAP) estimate using the standard technique of minimizing the negative log probability:

$$\begin{aligned} \hat{X}, \hat{Z}, \hat{D} &= \arg\max_{X,Z,D} P(C, L, E, U|X, Z, D) P(X, Z, D) \\ &= \arg\max_{X,Z,D} P(X, C, Z, L, D, E, U) \\ &= \arg\min_{X,Z,D} -\log P(X, C, Z, L, D, E, U) \end{aligned} \tag{3.2}$$

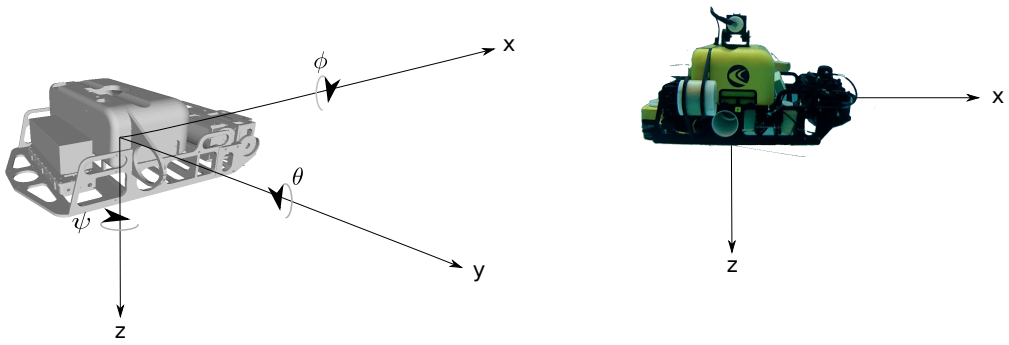


Figure 3-2: Coordinate system of the HAUV [37]

3.2 SLAM server

The role of the SLAM server is to asynchronously collect constraints from various sensors, combine them into a factor graph, and provide updated pose, landmark, and uncertainty estimates to other vehicle services. We use iSAM [41] to formulate and solve the factor graph.

Constraints come from a variety of sensors with varying time delays due to data transmission and post-processing. Full-resolution camera images are large, and may arrive several seconds late due to network congestion. Constraints from camera data may be further delayed due to image processing. Constraints based on submap alignment may arrive late due to the time required to denoise and smooth the sonar data. Our system adds target observations and pose to pose loop closures as they arrive, creating a new node and initializing it with the dead reckoning estimate.

3.2.1 Types of constraints

In general, constraints in factor graph SLAM take three forms: absolute, pose to pose, and pose to landmark. Absolute constraints are unary– they operate on a single node. Pose to pose constraints and post to landmark constraints are binary, representing a relative transformation between two nodes. Our particular implementation uses multiple types of absolute and relative constraints, described below.

Absolute constraints Absolute constraints are constraints on a single pose relative to the world coordinate frame. We initialize our robot position with a weak absolute constraint at the origin. We also use absolute constraints for the depth, pitch, and roll portions of the robot position. See Section 3.2.2 for details.

"Odometry" We refer to constraints between consecutive vehicle poses as odometry constraints, borrowing terminology from wheeled robot mapping. We use $\mathbf{x}_t^{\Delta}$ to refer to the ground truth vehicle motion between time $t-1$ and time t, such that $\mathbf{x}_t = \mathbf{x}_{t-1} \oplus \mathbf{x}_t^{\Delta}$. We use $\hat{\mathbf{x}}_t^{\Delta}$ to refer to the dead reckoning estimate of $\mathbf{x}_t^{\Delta}$ produced by the IMU. Our dead reckoning constraints between consecutive poses are therefore $\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_{t-1} \oplus \hat{\mathbf{x}}_t^{\Delta}$.

Pose to pose loop closures These are loop closures formulated as constraints between two non-consecutive poses, generated from submap alignment. In our implementation, pose to pose constraints from submap alignment are c_{ij}, which represent the relative transform between anchor poses $\mathbf{x}_i$ and $\mathbf{x}_j$. See Section 4.6 for details.

6DOF landmark observations Another form of loop closure is observation of a previously observed landmark. We denote the set of landmarks $L = \{\mathbf{l}_j\}$, with $j \in 0 \ldots N$, where $\mathbf{l}_j = [x, y, z, \psi, \theta, \phi]$ is the position of landmark j. Landmark observations are $Z = \{\mathbf{z}_k\}$ With 6DOF constraints $\mathbf{z}_k = [x, y, z, \psi, \theta, \phi]$ representing the relative position of the landmark from the robot pose . We use 6DOF landmark constraints for observations of our visual fiducial markers; see Section 5.2 for details.

Planar landmark observations Loop closures can also be generated by observation of a previously observed plane. We formulate planar constraints E in terms of the plane equation (unit normal and distance to origin) in the local vehicle frame. See sec 5.1 for details.

3.2.2 Dead reckoning and partial constraints

Dead reckoning loosely refers to vehicle-relative motion estimates from a moving platform. "Odometry" is often used to refer to dead reckoning solutions, e.g. visual odometry, and as a result, the words are often used interchangeably. The canonical example is wheel odometry, either measuring the rotation of the wheels using a rotary encoder, or, in the case of the Soviet *Lunokhod* rovers, an extra wheel acting as a dedicated odometer.

However, many dead reckoning systems combine absolute and relative measurements into the dead reckoning estimate (see Sec. 2.2 for examples of common approaches in the underwater domain). In our case, constraints on depth z come from a pressure sensor, which provides an absolute measurement of vehicle depth. Constraints on pitch and roll come from the IMU's accelerometers, which provide and absolute measurement of the vehicle's orientated with respect to gravity. x, y, and heading are relative constraints, computed by combining IMU and DVL estimates, and are formulated as binary partial factors between consecutive poses. In a factor graph, dead reckoning combining relative and absolute measurements can be formulated as two or more *partial constraints*. Partial constraints represent measurements with fewer degrees of freedom than the full robot state. In our case, depth, pitch, and roll are formulated as *absolute partial constraints*, unary factors applied only to those components of the robot state. The other state variables estimated by the IMU and DVL are x, y, and heading, which are formulated as *relative partial constraints*, binary factors constraining the remaining degrees of freedom between poses.

Note that the degrees of freedom in a partial constraint do not necessarily correspond directly to the constrained node's state variables. A classic example of partial constraints is measurement of a point feature, such as SIFT, in a 2D camera image. A SIFT feature is a point in $\mathbb{R}^3$. Using knowledge of the camera's focal length and principal point, the feature's bearing (azimuth and altitude angles) can be recovered from its position in the 2D image, but its range cannot. If we wish to formulate the relative partial constraint between the camera and the feature in Cartesian co-

ordinates, we cannot do so by simply removing some state variables as in odometry estimates. Instead, we would have to define a factor in terms of azimuth and elevation, and evaluate the cost of a proposed feature location (x, y, z) by computing its azimuth and and elevation relative to the camera. Another example of partial constraints using the full state variables can be seen in the planar constraints described in Section 5.1.

If we tried to optimize the least-squares problem representing the position of this landmark, we would find that it does not have a unique solution. Therefore, unless special provisions are made to handle non-unique solutions within the least squares solver (e.g. choosing an arbitrary point in the unconstrained dimensions), we must always ensure that the robot and landmark poses are fully constrained; that is, that there is a unique solution when all factors are considered. We therefore add our absolute and relative constraints simultaneously, without performing an incremental update between them.

Chapter 4

Sonar smoothing and submap alignment

Unlike laser scanners used in terrestrial mapping, sonars are very noisy. Terrestrial laser scanners typically exhibit error in the sub-millimeter range; even high-quality sonar data generally exhibits error on the order of centimeters. Beyond ensuring that the sonar unit itself is properly serviced and calibrated, there are two primary ways to reduce sonar noise for mapping applications. The first is to remove noise from the range image during the extraction of ranges for each beam. The second is to remove noise from the 3D point cloud after range extraction, during construction of the mesh. "Noise" in a point cloud refers to both points that deviate slightly from their ground-truth position and entirely spurious points that must be removed.

The goal of our noise removal algorithm is twofold. First, we want to remove sonar artifacts to present the most accurate possible final map. Second, we want to condition our submaps for alignment. Part of conditioning is just making the submaps accurate, but we also want to make sure they have roughly uniform density and accurate surface normal estimates.

We use a point cloud to represent submaps. Point clouds are frequently used with range sensors such as LiDAR and RGB-D cameras in terrestrial mapping. There exist a variety of techniques [16] [30] for creating watertight meshes from dense, hi-resolution point clouds produced by a laser scanner. In the case of incomplete data,

algorithms have been developed for hole-filling, smoothing, and denoising. See [95] for a more thorough survey.

Alternatives to point clouds include "implicit surface" representations, where the surface is modeled directly as a function of the input data. Two common implementations are the truncated signed distance function (TSDF) [16], implemented in real-time with an RGB-D camera in KinectFusion [35] [62]. These have the advantage of reducing small measurement errors automatically as more data is accumulated. Recent advances in Gaussian Process Implicit Surfaces [98] provide a nonparametric surface representation that appears to be robust to varying data density and outliers. GPIS has been used for terrain data [92] and 3D modeling [27], but is presently far too computationally expensive to run in real time.

Another alternative to a point cloud representation is volumetric modeling. Elfes introduced the occupancy grid [18], which discretizes the environment into a grid of cells, and tracks occupancy of each cell. Efficient implementations employ a tree structure like OctoMap [99], and can easily be run in real time. Occupancy grids can also be used to track probability of occupancy, and have been employed in underwater cave mapping [24]. By tracking all observations of each cell, probabilistic volumetric techniques will naturally tend to remove spurious observations as more data is added to the model.

We chose to use point clouds attached to anchor poses because this approach is more flexible than volumetric or implicit surface techniques. Point clouds have very fast insert, delete, and merge operations, which allow easy modification of the point cloud to correct for navigation error. Volumetric and implicit surface techniques, on the other hand, make it very difficult to remove or modify sensor data that has already been integrated into the model. It might be interesting to explore volumetric techniques for individual submaps, although the benefit of probabilistically tracking sonar returns would be reduced by not having much overlap. A better application of these techniques would be to apply them outside of our core mapping framework as a "sidechain," to produce an estimate using the corrected trajectory estimate.

4.1 Multi-beam sonar

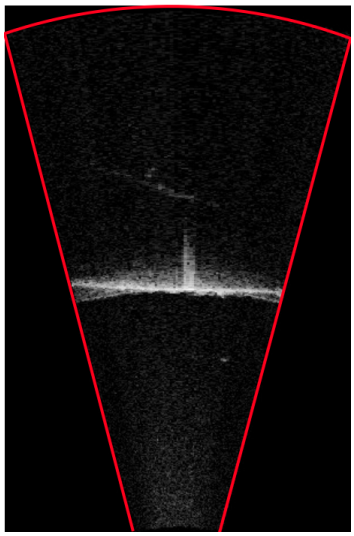
Sonars can be as simple as setting off a loud sound source, like an explosion on the surface, and listening to the response. The amplitude of the response over time gives an indication of the reflectivity of materials at different depths.

The simplest form of sonar used for mapping is a sidescan sonar. Sidescan sonar consists of two directional beams aimed perpendicular to the vehicle path. The return intensity along each beam indicates the reflectivity of the area below the vehicle, with high-intensity areas corresponding to reflective objects and surfaces sloped towards the vehicle, low intensity area corresponding to surfaces that are level or sloped away from the vehicle [39]. Objects that occlude the area behind them will produce a "shadow" in which no return is received. This shadow is often the most reliable method of finding a target, as it is more easily visible in the return intensity, and provides an indication of the targets height. Using the sonar geometry and vehicle altitude, and assuming a flat seafloor, sidescan intensity can be reprojected to create a reflectivity map of the area below the vehicle.

A more recent design is the multibeam sonar, which provides return intensity along an fan-shaped array of beams. These beams can be either highly directional transducers, or can be "virtual beams" created by digital beam-forming. The return intensity along each beam is discretized into "bins." Each ping of the sonar provides a two-dimensional array of return intensity, with a value representing the return strength for each bin of each beam. This array is called the *range image*.

There are two general types of multibeam sonar: imaging sonar and mapping sonar. Imaging sonars have a wide vertical field of view, akin to a sidescan sonar. Each ping provides an image of the reflectivity of the target area, similar to the map created by reprojecting sidescan data. As with sidescan sonar, imaging sonars are generally operated at a low grazing angle. Because of the large vertical field of view, the physical location of a target located in the range image (in beam-bin coordinates) is ambiguous. Target locations can be estimated using a flat bottom assumption.

Mapping sonar, also called profiling sonar or bathymetric sonar, has a narrow



(a) DIDSON Frame

(b) DIDSON Sonar

vertical field of view, and is analogous to a 2D laser scanner commonly found on wheeled robots. Contrary to an imaging sonar, mapping sonars work best when striking a target perpendicularly, as this provides the most distinct return in the range image. Because of the narrow field of view, a particular beam-bin location can be converted to a 3D Cartesian location using the sonar geometry. Due to the narrow field of view, each beam generally reflects off of only one target area, and return intensity along each beam is usually concentrated in a small number of contiguous bins at the target range. In practice, this range is taken to be the distance from sonar to object, and the amplitude information for other bins is discarded. Because this "range extraction" is dependent on the geometry and noise characteristics of a particular sonar unit, it's usually performed by software bundled with the sonar unit.

4.1.1 DIDSON

The sonar used for this work is the SoundMetrics Dual-Frequency Identification Sonar (DIDSON) [9], used in "identification mode" at 1.8 MHz. At this frequency, the window is comprised of 512 bins per beam, covering a range of 1.25 to 10m. The range can be extended by increasing the window start distance (the distance of the first reported return) from .42m to 26.1m in half-meter increments. We generally use

a window start distance of 2m, and a window size of 10m for the safe distance survey and 5m for the close inspection.

The DIDSON has a 96-element transducer array, with each transducer corresponding to a single beam. The beams are aimed using a fluid-filled lens array, which directs the beams into a fan pattern with a 29° horizontal field of view. Each beam has a width of .3° in the horizontal direction. The vertical field of view is user-configurable with "spreader" or "concentrator" lenses. We use a concentrator lens with a 1° vertical field of view.

Note that the specified beam widths are not absolute cut-offs; they represent the angle offset that causes a 3dB reduction in return strength of the main lobe. In addition, the beams may have side lobes, which pick up off-axis returns. This can cause "crosstalk" interference between adjacent beams. To alleviate this, the DIDSON uses an 8-phase transmit/receive cycle, with each phase firing 12 non-adjacent beams.

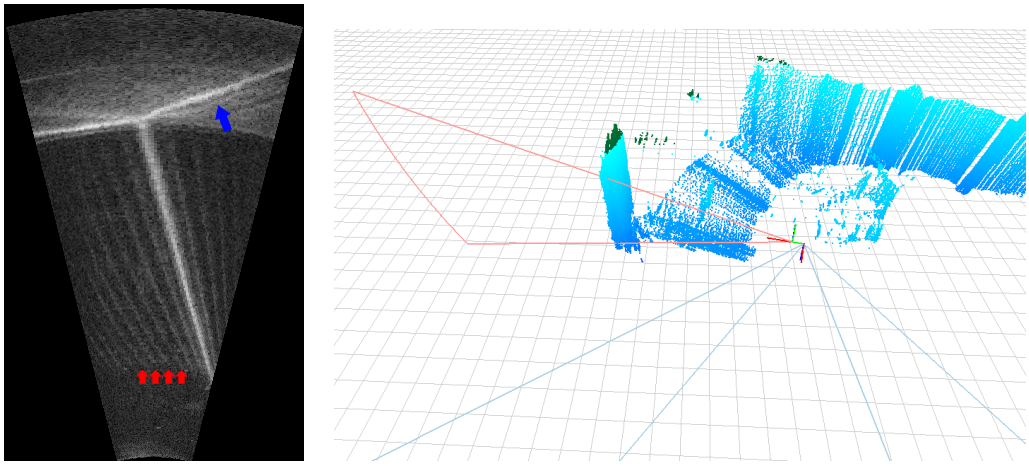
4.2 Sources of error

The most common types of multibeam sonar errors are described below.

Acoustic and electrical noise floor The noise floor is primarily a function of environmental and electrical noise. Although marine infrastructure, particularly harbors, tends to have a high level of ultrasonic background noise, it is generally far lower than weakest return produced by reflection off of a target. Therefore, the noise floor is not generally a source of significant error in profiling sonar.

In some cases, artifacts arise from interference with other underwater acoustic instruments, such as sub-bottom profilers [60] or acoustic projector arrays [50].

Zero beam width approximation Multibeam mapping operations typically approximate the beams as having zero thickness. In practice, the beam ensonifies a larger area of the target. When the beam is perpendicular to the target, as is the case in most bathymetric surveys, the ensonified area for each beam lies at roughly uniform distance from the sonar. However, when the beam strikes

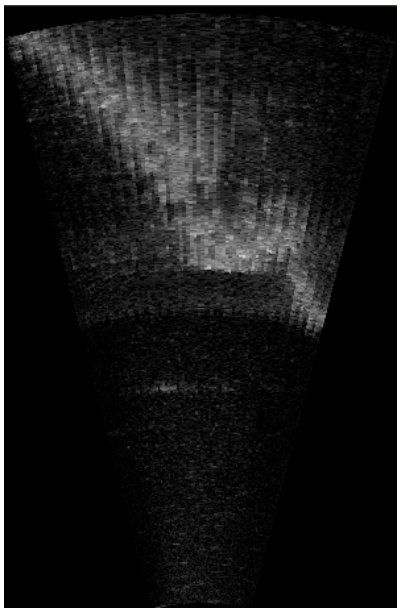


(c) (d)

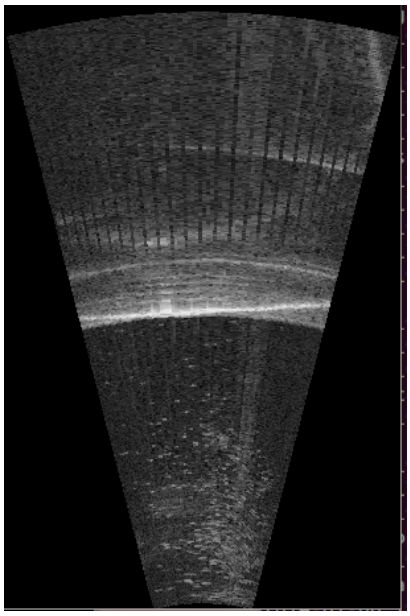
Figure 4-1: (a) A DIDSON return in the corner of a swimming pool, showing "ghosting" due to sidelobe interference (blue) and false returns due to multipath (red). The actual return from the vertical wall is reflected off the sloped pool floor, creating a "false wall" extending below the bottom of the pool. The scene geometry and false wall can be seen in (b), with the red outline representing the sonar cone, and the blue frustum showing the forward-looking camera field of view.

the target with a low grazing angle (high angle of incidence), it ensonifies an area spanning a larger range of distances. This produces rangewise "blurring" in the sonar beam, shown in Fig. 4-2(a).

Crosstalk The ideal sonar model also neglects the effect of "crosstalk" between beams. Beams on the DIDSON are spaced at .3°, and have only 3dB of attenuation at 1° off axis. Therefore each beam would receive a nearly full-strength return from its neighbors if they transmitted and received simultaneously. The DIDSON avoids this with its 8-phase staggered firing, which gives 2.4° spacing between simultaneously fired beams. Even at this spacing, a particularly strong return in one beam, such as that caused by pinging a highly reflective metal ship hull at zero grazing angle, will cause interference in all other beams of the same firing phase. In most cases, this interference is of lower intensity than the legitimate on-axis return, resulting only in a slight "ghosting" or blurring. In extreme cases, strong returns can cause overwhelming interference across the entire array, leading to semi-circular artifacts shown in Figure 4-2(a)



(a) Sonar ping showing "blurring" due to nonzero beam width



(b) Sonar ping showing "tunneling" artifact due to crosstalk

These artifacts are described in prior work [60] as "tunneling" because of the semi-cylindrical artifacts in the resulting bathymetric map.

Vehicle motion Another source of of error is vehicle motion during the time between when the ping is transmitted and when the return is received. The speed of sound is roughly $1500 m/s$ in water, meaning that the time between transmission and reception of a ping at 15m range is around $20ms$. The vehicle is unlikely to travel a significant distance in this amount of time, but it may pitch or roll rapidly enough to corrupt the data (e.g., due to wave motion). Even a small error in sensor orientation, well under 1°, can cause a significant error in the estimated position of the return. In our $15m$ example above, a .5° orientation error could cause a $13cm$ error in the estimated position of the return.

Furthermore, the DIDSON and other mapping sonars have a staggered timing, such that subsets of the beams are fired at different times for the same sonar frame, which compounds motion artifacts if not compensated. The DIDSON's 8-phase timing is combined into a single virtual return with a single timestamp, thereby compounding motion artifacts by a factor of 8. Fig. 4-2 shows the

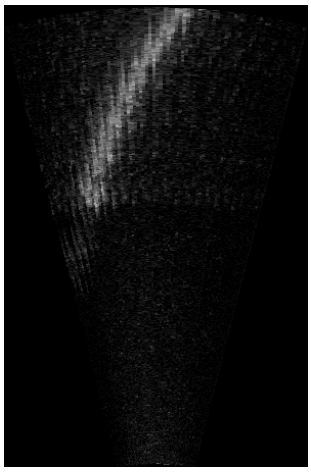


Figure 4-2: Sonar ping showing "combing" artifact due to vehicle motion

"combing" artifacts produced by vehicle motion over the course of one frame.

Multipath interference Multipath reflections are a common problem in underwater acoustic communications, ranging, and mapping. Deepwater AUV bathymetry is relatively free of multipath interference because of the lack of reflective surfaces in the environment. In operations near the surface, reflections from the almost perfectly reflective air-sea boundary can be nearly as strong as the original return. This is exacerbated by a shallow seafloor, particularly if the seafloor is a highly reflective material such as gravel, large rocks, or concrete. Other flat, reflective surfaces, such as concrete pilings or metal ship hulls, can also produce multipath interference.

Dual-path interference, produced by a single reflection, can appear as false returns in the sonar range image. If the additional path length is within the range of the sonar window, these artifacts will appear at a longer range the actual target. See Fig. 4-3 for an example of dual-path reflections. Rarely, multipath interference will produce interference between subsequent pings. These reflected returns are generally of lower magnitude than the primary, meaning that in a simple dual-path case, they will not necessarily interfere with thresholded range extraction. However, the reflections can intersect with legitimate returns or with other reflections and thereby artificially increase the amplitude,

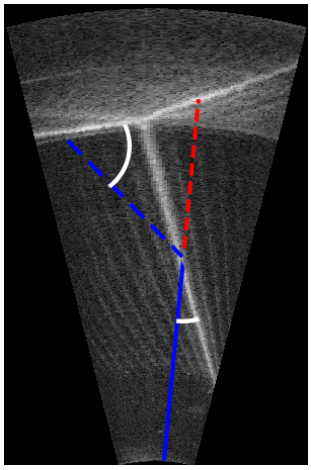


Figure 4-3: Artifact from a dual-path reflection. Actual sound path is shown in blue, with the solid line representing the primary path and the dashed line representing the path of the reflection. The dashed red line shows the apparent path of the reflection, where the artifact appears in the range image. In this case, because the grazing angle of the reflected return is much high than that of the actual return, the reflection is stronger than the primary signal. See Fig 4-1 for the scene geometry that produced this image.

creating artifacts in the extracted ranges.

Sound speed gradient As with any acoustic measurement device, sonar is sensitive to errors in the estimated sound speed profile [58]. In seawater, the speed of sound can vary between approximately 1450 and 1550m/s, depending primarily on salinity and temperature. In long range acoustic operations, a full water column profile can accurately compensate for depth-dependent changes. However, AUV mapping operations generally rely on a point estimate, either hardcoded in pre- or post-processing, or directly measured by the vehicle. For deep water, comprised of large masses of roughly uniform density, the point approximation is accurate. However, the first few tens of meters of the water column exhibit high variance in sound speed profile. The surface layer is warmed by solar radiation, and has high salinity due to evaporation. Because temperature diffuses faster than salinity, the temperature-salinity distributions in the few tens of meters below the surface is highly uneven. Wind and wave action mix water layers, causing spatial disturbance in the depth-dependent distribution of water

density. As a result, the sound velocity profile of shallow water is unpredictable, both between and across depth layers. This causes errors in the estimated range to sonar targets, up to a magnitude of a few percent of the distance traveled. For the DIDSON, which has a range over $10m$ and centimeter-level accuracy, this could cause measurable distortions in the extracted ranges.

Systematic issues All of the above artifacts can be predicted using the beam-forming equations of a mathematically ideal sonar. In real-world applications, a variety of small calibration issues contribute with varying degrees to the overall measurement noise. Issues such as minor lens imperfections, slight variance in position after re-mounting the sonar, and flex between vehicle and sonar all contribute small amounts of measurement noise, which can usually be ignored. With our particular vehicle, some of these issues occasionally cause errors large enough to appear in our final data: a bubble in the acoustic lens array, non-rigid mounting of the outer concentrator lens, and slop in the actuators are the most common examples.

It is worth asking why these types of error are being considered in the design of a mapping system, rather than being pushed aside as "implementation issues." Compared to other mapping AUVs, ours has relatively high-end hardware, and is well designed and maintained. Hundreds of hours of engineering have been devoted to eliminating them, yet they frequently recur in our operations. If these errors persist in a system that is arguably state of the art, we cannot ignore them when designing a general-purpose sonar mapping algorithm. Therefore, we consider these "sonar noise" because they are nearly inevitable when operating a vehicle in harsh environments.

"Legitimate" outliers Harbor environments are dense in biochemical energy due to runoff from surrounding land and growth on marine structures. As a result, surfaces are often heavily biofouled, either with hard growth, such as mussels and barnacles, or soft growth, such as plants and algae. This attracts species higher up the food chain, such as fish and crabs, in large numbers. In our op-

erations, marine life causes "noise" by attaching to surfaces, creating reflective bubbles, or swimming between the sonar and target. Though not strictly "outliers," these returns interfere with our 3D mapping operations, and we consider them a source of noise to be filtered out when possible.

These noise sources are found to some degree in all mapping operations. However, complex-area mapping exhibits some specific characteristics that exacerbate common types of noise. Harbors have a generally high *noise floor* due to the presence of ships, loading and fueling machinery, and other acoustic instruments. Shallow water and flat, reflective infrastructure, such as ship hulls and concrete pilings, contribute to both *crosstalk* and *multipath* interference. Because lines of sight are limited in cluttered environments, complex area inspections often require low grazing angles, which cause *blurring* due to nonzero beam width. The shallow harbor environments complicate *sound velocity measurement*, and tides and wave action can cause unpredictable vehicle sway, contributing to to *motion errors*. In many ways, complex inspection in a shallow harbor is a worst case scenario for common type of sonar errors.

4.3 Range extraction

The standard approach to extracting target ranges from the intensity image is to simply choose the bin with the highest intensity above a certain threshold. This approach is highly effective because, although many of the artifacts described above can be seen in the intensity image, most artifacts do not reach the intensity of the true return. In particular, this simple approach removes the noise floor, multipath returns that do not intersect with the true return, and small artifacts from marine life.

The remaining noise sources are: blurring due to the zero beam width approximation, crosstalk, motion artifacts, soundspeed error, calibration issues, and marine life. The latter three produce artifacts that cannot be distinguished from valid returns using the range image alone. Soundspeed error can only be corrected with ground truth knowledge of the sound velocity profile between source and target. Calibration

issues are extrinsic to the range image and should be addressed elsewhere. Marine life could feasibly be removed from the intensity image if it could be accurately identified from a single frame; we leave this to future work.

Prior work ([59], [53], [31]) on removing multibeam sonar artifacts has primarily focused on bathymetry data. These techniques are based on identifying bad beams by comparing the ranges extracted from neighboring beams. This can be done either by averaging beam ranges across many pings to create a sliding-window seafloor model [60] or by simply detecting discontinuities between neighboring beams in a single ping [59]. This is effective at removing small numbers of bad beams with large range errors, such as those caused by acoustic interference or crosstalk.

Because our complex area inspection task cannot assume a continuous seafloor with bounded slope, sliding-window techniques are not appropriate. When imaging multiple occluding targets from arbitrary angles, discontinuities between adjacent beams will be common and will not generally represent outliers or artifacts.

Furthermore, bathymetric outlier rejection does not attempt to de-noise inlier pings, which are reprojected as-is and taken to be ground truth. As a result, these prior approaches do not address "combing" motion artifacts or the "blurring" caused by the zero beam width approximation.

Noise in the inlier beams could conceivably be addressed by a parametric line or curve fitting, either to the extracted ranges or to the raw backscatter data. However, to avoid smoothing out "valid" discontinuities would require active tracking of surface edges, which would be difficult using only the 2D backscatter image.

For these reasons, we believe that denoising sonar data for complex inspection should not be done at the level of an individual ping, or a window of consecutive pings. Instead, outlier detection and smoothing should be addressed primarily *after* range extraction, in the reprojected 3D point cloud. The primary advantage of this approach is that discontinuities, gradients, and outliers will be analyzed relative to the full 3D structure. This allows us to preserve arbitrary gradients and points that may appear to be outliers within a single ping, but which are revealed to be valid by their spatial relationship to prior or subsequent data. It also allows us to perform

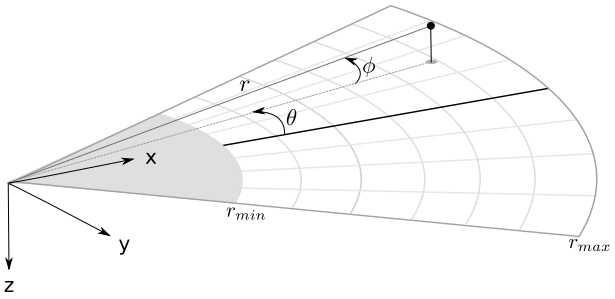


Figure 4-4: Multibeam sonar geometry (modified from [37])

a more informed de-noising of the inlier points, by evaluating them relative to their spatial neighbors.

4.4 Submap formation

We use a simple thresholding operation to produce point estimates in the range image.

This gives a series of points in beam-bin coordinates, which can be translated into standard polar coordinates and represented as $\mathbf{p}_{t,i}^{pol} = [r, \theta]$, where t is the timestep of the ping and i is the beam index. We transform them from sonar coordinate frame into vehicle coordinate frame, getting $\mathbf{p}_{t,i} = [x, y, z]$.

We aggregate points over a time range to produce a submap $\mathbf{s}_t$ associated with anchor pose $\mathbf{x}_t$. A submap $\mathbf{s}_t = [\mathbf{p}_0 \dots \mathbf{p}_{|\mathbf{s}_t|}]$ consists of points with associated normal vectors, $p = [x, y, z, n_x, n_y, n_z]$ stored in the local frame of the anchor pose.

Our submap catalog is $S = \{\mathbf{s}_t\}$,

4.5 Submap smoothing

To remove the artifacts described above, we have devised a smoothing and outlier rejection algorithm. Our smoothing is based on a parametric surface approximation using moving least-squares [5]. In the general case, there is no reason to expect arbitrary input data to form a smooth, watertight surface. However, in the underwater inspection domain, we find this technique justified for two reasons (beyond its empir-

ical effectiveness). First, we make the general observation that the seafloor is almost by definition a watertight manifold. Second, many of our inspection targets are anthropogenic structures which do, in fact, have a good polynomial approximation. We use the implementations provided by the Point Cloud Library ([80]) for many of the steps below.

The first step is a voxel filter, in which the submap is divided into cubes of size v_1, and if multiple points occupy the same cube, they are removed and replaced with a single point at their centroid. We choose v_1 to be 2cm, which is approximately the spacing between DIDSON beams at medium range. This spacing serves to remove redundant data, where the sonar scanned the same physical location multiple times, without significantly reducing the resolution of the submap.

Second, we perform k-nearest-neighbor outlier rejection, in which points are rejected if the average distance to their k_{nn} nearest neighbors is $> \sigma_{nn}$ standard deviations above the mean (for details and implementation, see [81]). This step is designed to remove artifacts caused by electrical noise, minor multipath reflections, and small fish. $k_n n$ can be considered a target cluster size: it is related to an isolated cluster of points must reach to be considered inlier. Because this cluster size is based on number of points, rather than metric size or volume, our outlier rejection is sensitive to changes in data density. Consider the case in which the vehicle stops to plan or await operator input, and the sonar images the same area repeatedly before continuing the inspection. In this case, we may encounter a submap in which most points are concentrated around a small area and therefore have very small nearest-neighbor distances. The rest of the points in the submap would be considered outliers and discarded. By filtering out duplicate data in the prior step, we ensure a minimum spacing among clusters of points and avoid this problem.

We found values of $k_{nn} = 50$ and $\sigma_{nn} = 2$ to be effective at distinguishing outliers from legitimate returns. This is approximately half of the points from a single ping in which every beam strikes a target. With our voxel spacing of $v_1 = 2$cm, our "target cluster size" of 50 voxels is approximately $O(.02m^2)$ of surface reflection. For a detailed listing of parameters, see Table 1 in the appendix.

The third step is a larger voxel filter of size v_2. Before this step, point clouds are free of duplicate data, but still often show "scan line" artifacts: the points from withing a single ping are significantly closer to each other than to points from adjacent pings. These scan lines interfere with ICP alignment by creating strong local minima in the cost function, biasing the results towards transformation estimates in which the scan lines are aligned. Scan lines also interfere with our moving least squares surface reconstruction, which is robust only to *smooth* changes in data density. Therefore, we choose v_2 to be roughly the distance covered between DIDSON frames (5-10 Hz) when the vehicle is moving at speed (.5-1 m/s), giving us $v_2 \approx$ 10cm. This step has the effect of combining adjacent points from within DIDSON frames, so that their spacing more closely matches the spacing between frames.

Finally, the resulting points are smoothed using a local parametric approximation as described in [5]. For each point, a polynomial surface is constructed which minimizes the mean squared error to points within a radius r. The point normal is estimated as the normal to the parametric surface, and the point is projected onto the surface. The surface normal is also stored for each point, as it will be utilized in the alignment step. This step helps to remove noise resulting from the zero beamwidth approximation and vehicle motion, as well as reduce the effect of surface biofouling.

Data: s_t is the input submap, S is the submap catalog, c_{it} is a 6DOF transform, Σ is a 6x6 covariance matrix

$\hat{x}_t$ = queryMAPVehiclePose(t);
foreach $s_i \in S$ **do**
    $\hat{x}_i$ = queryMAPVehiclePose(i);
    c_{it}, Σ_{it}, fitnessScore = align(s_i,s_t,initialEstimate = $\hat{x}_t \ominus \hat{x}_i$);
    if *fitnessScore* $< \alpha$ **then**
        addConstraint(x_i,x_t,c_{it}, Σ_{it});
    end
end
addSubmap(s_t);

Algorithm 1: Generating submap constraints with ICP

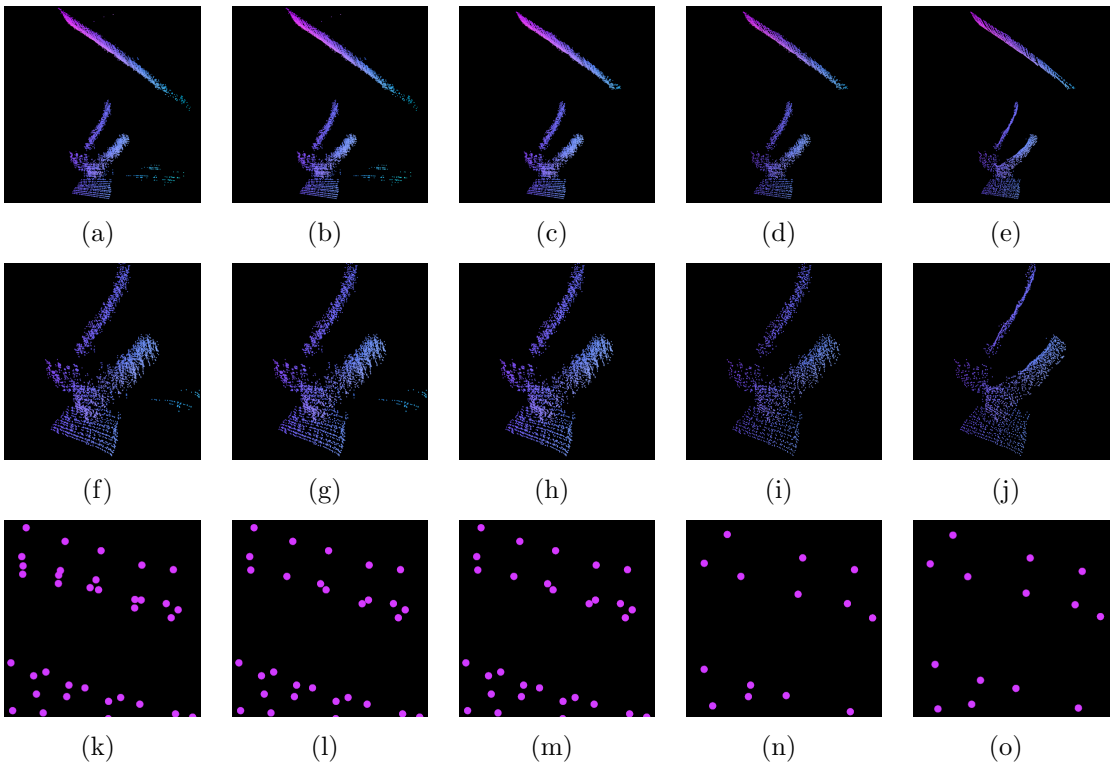
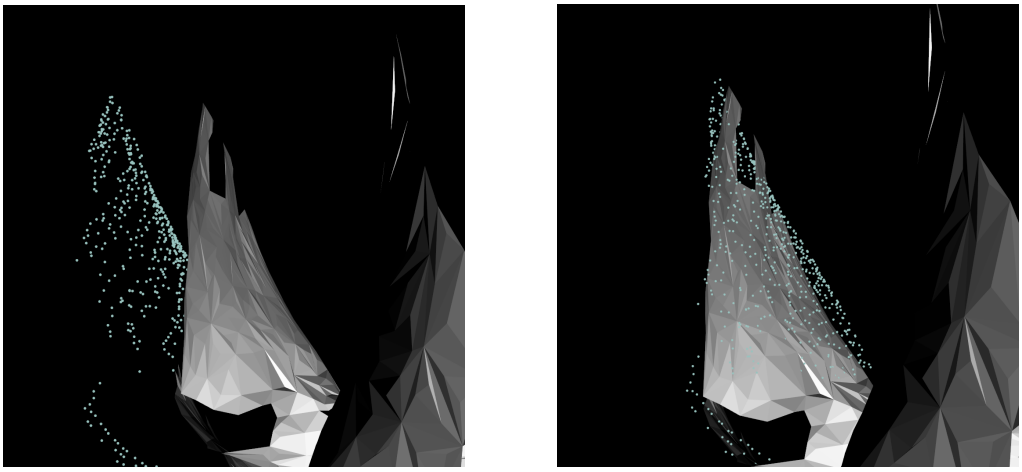


Figure 4-5: A submap as it progresses through the four steps of the smoothing process. This submap consists of a vertical portion of the middle of the propeller and a segment of the hull, viewed from the starboard aft side. From left to right: the raw submap, submap after voxel filter 1, submap after outlier rejection, submap after voxel filter 2, and submap after parametric surface modeling. (a-d) wide view showing outliers. (f-j) detail shows smoothing of propeller blades. (k-o) illustrate the relative point density at each step.



(a) (b)

Figure 4-6: Two submaps before and after alignment. One submap is triangulated for visualization.

4.6 Submap alignment

We align submaps using Iterative Closest Point (ICP) [10], an algorithm for aligning a set of *measurement points* (also called *source points*) to a *target model* by iteratively computing rigid transformations that minimize the sum of squared distances between the points and target model. Each measurement point is represented in Cartesian coordinates as $p = [x, y, z]$, and the target can be any model that allows computation of distances to a point, such as a parametric surface, line, or another point cloud. In our case, we have normal estimates for each submap, so we use point-to-plane distances in computing the transform.

For measurement points p'_i corresponding to target points p_i with normals n_i in the target, ICP computes the rigid transform T that minimizes the sum of square errors between the measurement set and the target set:

$$\sum_i ||(p'_i - Tp_i) \cdot n_i||^2 \tag{4.1}$$

ICP is known to be highly sensitive to initialization, due to the local minima which are often present in the cost function. If not initialized close to the correct solution, it will converge to the wrong local minimum. We address this issue in

two ways. First, we initialize ICP with the most recent estimate of the relative position of the source and target poses. This is equivalent to assuming that the dead reckoning between the last correctly aligned pose and the current pose is within the region of attraction of the correct alignment. Second, we assign a *fitness score* to each alignment that represents the normalized sum of squared distances between corresponding points. Smaller fitness scores represent better alignments. If the fitness score exceeds a threshold α, we reject the alignment and do not add a loop closure constraint to the pose graph. This threshold is dependent on the scope of the data set, point density, and noise levels; we found $\alpha = .1$ to be a reasonable value for our experiments.

When a suitable alignment is found, we transform the anchor pose of the source submap, and formulate a relative constraint between the source and target anchor poses. This constraint is added as a factor in the pose graph.

For the purposes of this work, we use a simple exhaustive search to find potential loop closures. Future work may address more advanced techniques for data association.

4.7 Relation to previous work

Underwater mapping has historically focused on 2.5D bathymetry, which is sufficient to represent terrain features for navigation, petroleum exploration, geological surveys and other scientific purposes. Compared to bathymetric submap-based SLAM [77] [76], our data is noisier, submap alignment has a additional degrees of freedom, and the resulting model is more complex. For this reason, we have introduced a novel technique for denoising and smoothing submaps to improve alignment. Whereas most previous work uses filtering approaches [7] [74] [26] [93] or offline pose graph optimization [8] [49], which are sufficiently accurate for planning and obstacle avoidance in bathymetric surveys, complex area inspection requires online position and trajectory updates for obstacle avoidance and coverage planning. To our knowledge, our technique is the first to utilize loop closures from 3D submap alignment

in a factor graph framework to provide online position and trajectory for mapping complex 3D underwater environments.

4.8 Experimental Results

4.8.1 USS *Saratoga*

We tested our system in an underwater inspection scenario on the USS *Saratoga*, a decommissioned aircraft carrier in Newport, RI. We first performed a survey of the running gear from a safe distance, approximately 7-8m. See Figure 4-7 for a reconstruction of the outboard port propeller, drive shaft, support struts, and rudder based on our initial survey.

Figure 4-7: Running gear of USS *Saratoga*, outboard port side.

We elected to test our SLAM system on the rear face of the outboard port propeller, approximately 7m in diameter, and a portion of adjacent hull. To generate a mesh for our coverage planning algorithm, we ran a safe distance survey, consisting of three vertical track lines from the aft side, at 5m range, with the DIDSON fixed in position (not actuated). We used this mesh to generate a trajectory for full coverage of the target area using the sampling-based coverage planning algorithm described in [22]. The planned path, consisting of 24 waypoints at which the DIDSON is swept vertically, is shown in Fig. 4-8. The safe distance survey took approximately 15 minutes, and the close inspection trajectory took approximately 20 minutes. Based on the reprojection error between the two surveys, we estimate that navigation drift was

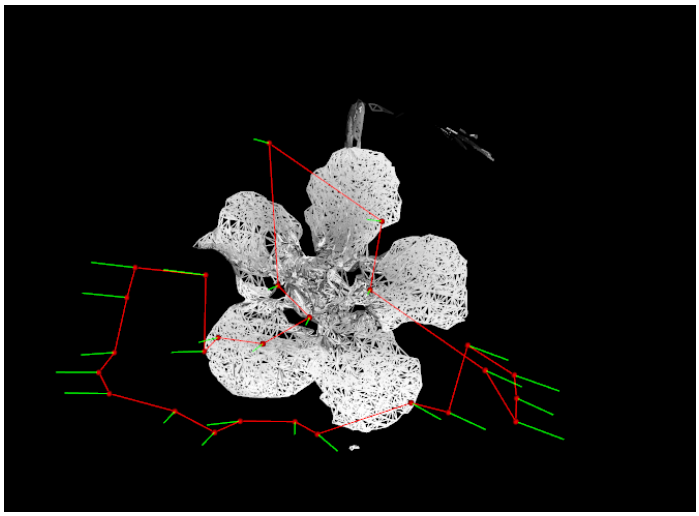


Figure 4-8: Planned path for inspecting the propeller of the USS *Saratoga*.

on the order of 30-50cm. This an atypically large amount of drift for a short survey. A possible explanation is that the tidal outflow being directed under the narrow gap between the ship and seafloor, which was strong enough to nearly stop the vehicle at some points during the inspection, stirred up silt and degraded the quality of the DVL motion estimates. This, combined with the heading drift of our compass-less IMU, resulted in large errors in our dead reckoning trajectory estimate.

For this survey, we attached AprilTags to the hull of the ship, approximately 2m above the water line (see Fig. 4-9). AprilTags were captured in multiple frames before and after the underwater surveys, resulting in 22 vehicle-landmark constraints from successful tag detections. The tags were angled sharply relative to the vehicle (approximately $60-70^{\circ}$ off axis), and were captured in the edge of the camera frame, both of which degrade positioning accuracy and compound the effects of camera calibration errors.

A mesh generated from our inspection trajectory is shown in Fig. 4-10(b). The raw point cloud was denoised and smoothed using a variant of our submap smoothing algorithm described in Section 4.5. It was then meshed with a simple greedy triangulation. The raw sonar returns, reprojected from the optimized vehicle trajectory, can be seen in Fig. 4-10(a). Behind the propeller, the drive shaft is visible, along

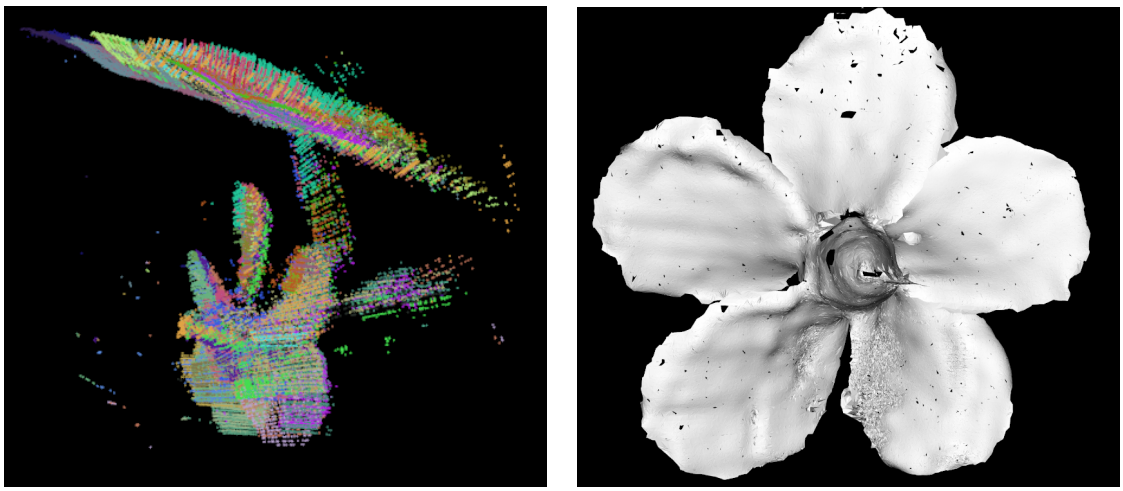
Figure 4-9: An AprilTag captured by the periscope camera on the hull of the *Saratoga*, Newport, RI. This is a cropped detail of the upper left quadrant of the camera frame.

with a support strut. Note the high number of spurious sonar returns. Because the ship was not operational, it had a significant amount of growth on the propeller and shaft, and had become a habitat for fish, oysters, and other marine life. Marine life was visible in the sonar range images, and often appeared as spurious returns in the extracted point cloud.

While we can visually confirm that our technique produced improved maps, it was difficult to quantify the improvement in the absence of ground truth trajectory or 3D models of the inspection target. To address the lack of ground truth, we created a high resolution map combining data from multiple surveys. We utilized only data with low angles of incidence (under approx. 30°) due to its improved noise characteristics, and hand-corrected for navigation drift. This produced a high quality, drift-free map that, although it is not absolute ground truth, provides a reasonable baseline for evaluating reprojected maps. Figure 4-11 shows the reprojection error of the dead reckoning and SLAM trajectories evaluated against the handmade mesh.

4.8.2 Synthetic data

To isolate the effect of pose graph alignment, we also generated a partially synthetic data set based on the actual vehicle trajectory. We corrupted the vehicle trajectory with navigation drift, simulated by incrementally adding zero mean Gaussian noise



(a) (b)

Figure 4-10: (a): The final point cloud from the propeller inspection trajectory. Raw submaps have been reprojected according to the SLAM-corrected trajectory, and points are colored by submap. (b): A smoothed mesh generated from the raw point cloud.

to dead reckoning measurements of x, y, and ψ state variables. We used a standard deviation of .01m for x and y, and .00001 radian for ψ, accumulating at 20Hz. We also added zero mean Gaussian noise to the absolute measurements of z, θ, and ϕ to simulate increased measurement noise without drift. We also added synthetic sonar measurements created from the actual vehicle trajectory and a high resolution sonar map of the propeller gathered in a previous experiment. To simulate sonar pings at each waypoint, we extracted points in the sonar field of view, downsampled them to the sonar resolution, and added Gaussian noise with a standard deviation of 5cm.

Fig. 4-12 shows the reprojected clouds from our synthetic data set, using the synthetic dead reckoning and SLAM trajectories. As is apparent, the dead reckoning error caused misalignment in the reprojection, which is corrected in our SLAM framework. Fig. 4-13 shows the average per-pose trajectory error over time. At each loop closure event, the SLAM error is reduced, while the dead reckoning error continues to accumulate. Note that, in the SLAM reprojection, the sonar noise dominates the alignment error, even though the trajectory is estimated based on the noisy sonar returns. We attribute this to our smoothing procedure detailed in Section 4.5.

This graph shows the average error over *the entire estimated trajectory* at each time

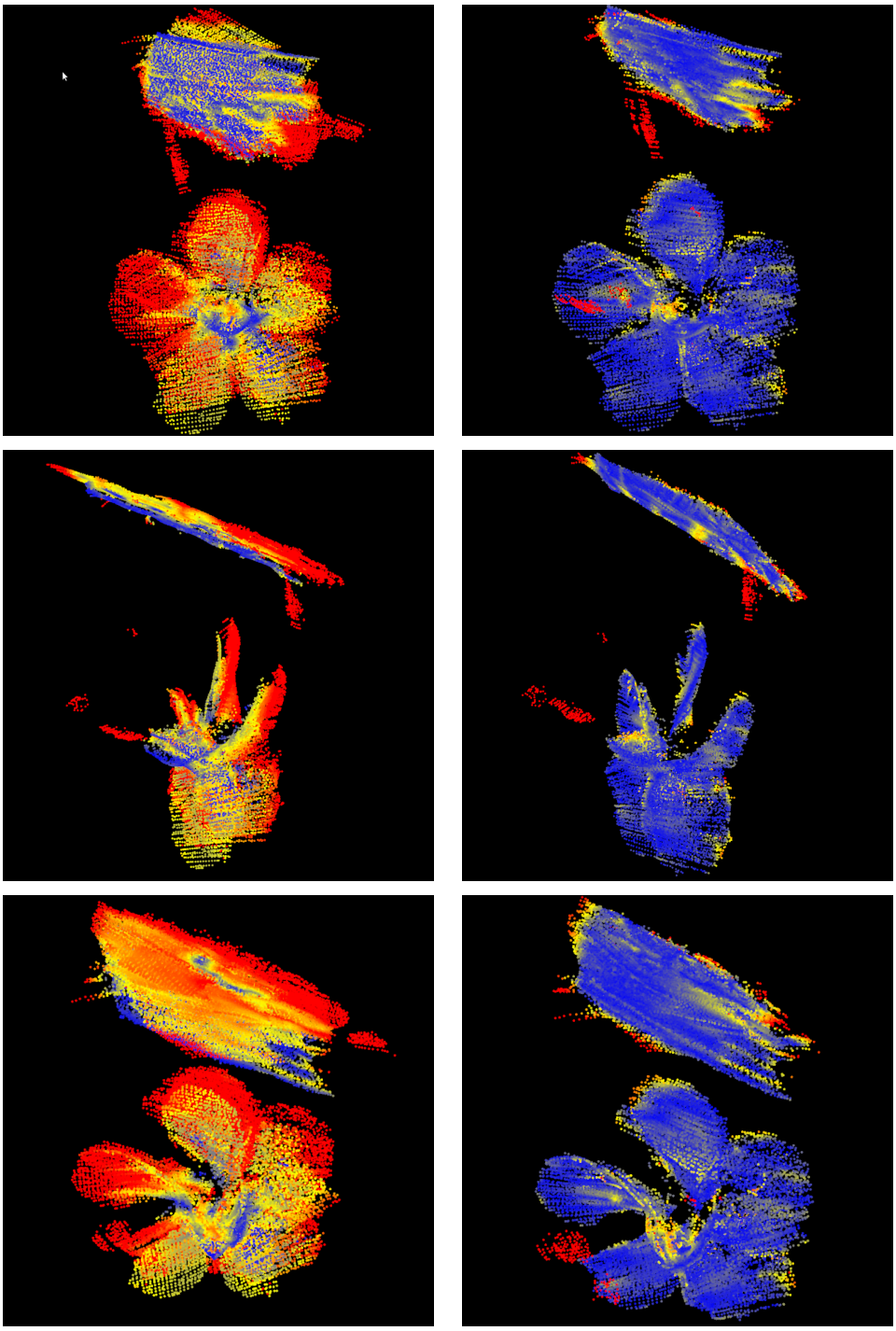


Figure 4-11: Comparison of reprojection error from dead reckoning and SLAM trajectories. On the left are three viewpoints of the map created using dead reckoning, and on the right are the same viewpoints of the SLAM solution. Error is evaluated based on distance from a manually aligned and denoised mesh which serves as our ground truth. Points are colored by distance from the handmade mesh, with blue, yellow, and red representing error of 0, 0.5, and 1.0 meters, respectively.

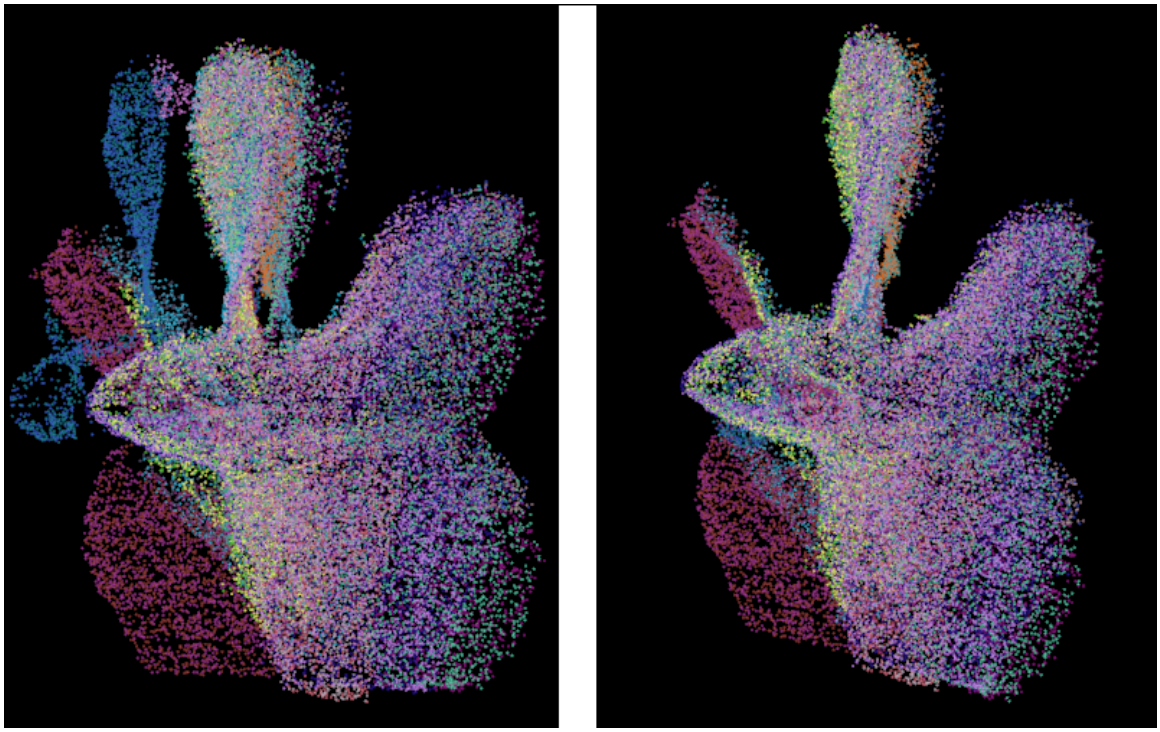


Figure 4-12: Comparison of point clouds generated from synthetic data: dead reckoning (left) and SLAM (right).

step, rather than the instantaneous error. This is a more informative metric for our purposes than instantaneous position error, because we are concerned with ensuring coverage of the entire inspection target. The first loop closure occurs approximately halfway through the simulated trajectory. At this point, the average error is significantly reduced, although there remains substantial error accumulated during the first half of the trajectory that cannot be corrected by a single loop closure. Subsequent loop closures further reduce the average per-pose error. Around $t = 1200$s, our random walk simulating navigation drift began to converge back towards ground truth, reducing average error. Our field tests generally produced monotonically increasing error, and we expect that such error would show an even greater improvement in the SLAM solution over dead reckoning. However, the Gaussian random walk used to produce our synthetic data makes minimal assumptions and provides an unbiased basis for comparison.

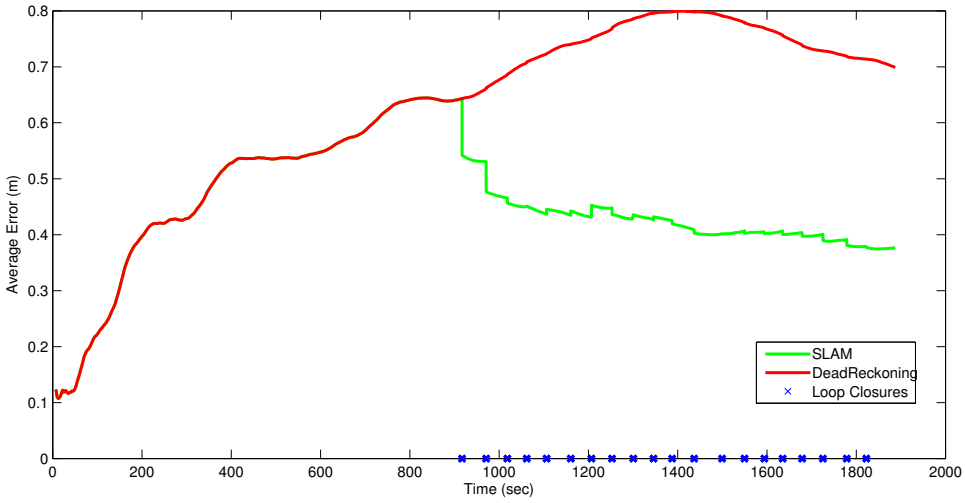


Figure 4-13: Average per-pose error over time for the synthetic data set.

Chapter 5

Structured visual and acoustic features

Computer perception generally makes as few assumptions as possible about the environment. Popular visual feature detectors such as SIFT [55] operate at the level of clusters of pixels, tracking distinctive features without attempting to reconstruct their physical properties such as symmetry or parallelism. Three dimensional laser and LiDAR data is commonly represented as a point cloud, a simple unordered collection of points. These representations are well matched to the output of mobile robot sensors, and there are solid theoretical reasons for operating on low level data structures beyond their practical utility: simpler representations with fewer assumptions are well suited to a wider range of conditions, and tend to perform well even when operating in unfamiliar environments.

However, most environments feature some geometric structure. A very simple example is continuity of objects: given a picture of an object on a desk, one can reasonably assume that the desk surface continues underneath the object. However, this structure is not captured directly in the image, nor would it be captured in a 3D point cloud of the scene.

In addition to 3D geometric structure, many environments contain distinctive visual features. We consider a visual feature "structured" if it provides some information beyond that of a pixel-based point feature such as SIFT or SURF. For

example, a flat text label, such as a serial number, on an underwater valve could provide information about the valve's location relative to the pipeline, and relative to other valves. Even in the absence of prior information, the structure of the text could be exploited, for example to determine the relative angle of the marking surface relative to the vehicle.

We aim to exploit both geometric and visual structure of the environment to improve our localization and mapping abilities for underwater inspection tasks. We introduce a technique for extracting planes from sonar data, formulating constraints between vehicle poses and planar landmarks, and jointly optimizing vehicle and plane locations within our factor graph SLAM framework. We also present a technique for integrating 6DOF constraints from visual features into our sonar mapping framework. We use an existing visual fiducial marker system to stand in for the structured visual features, such as bar codes or serial numbers, commonly found on underwater inspection targets.

5.1 Planes

Many environments contain manmade features with predictable geometric properties that can be leveraged to improve localization. These features are well suited for use in SLAM because they can be represented and optimized much more efficiently than dense point cloud data. We have created a system for tracking planar landmarks within a factor graph SLAM framework. Below we present related work, followed by our system of extracting and aligning planes.

5.1.1 Related work

Previous work has demonstrated the effectiveness of using planar landmarks for both localization and SLAM. Planar features are a compact representation of an environment, and by representing closed surfaces, rather than unorganized points, they are more useful for tasks such as ray-casting and trajectory planning. Furthermore, they can be used to generate constraints from noisy data with little overlap. This makes

them particularly attractive to mapping with multibeam sonar units, which are noisier than laser sensors used in terrestrial mapping, and which generally yield very little overlap between measurements due to narrow beam width and relatively slow data rates.

Fallon et al. [25] used planar approximations to performed 6DOF particle filter localization efficiently within the known map. Lee et al. [51] present a system for planar mapping using an RGB-D camera and hierarchical pose graphs. The authors introduce a fast plane extraction and boundary tracking algorithm, a technique for estimating frame-to-frame constraints using the parameters and spatial boundaries of extracted planes, and a method for merging planes. A local map is formed using frame-to-frame constraints, and when the local map is sufficiently large, it is used to generate loop closure constraints within the global map.

Ribas et al. [73], [74] created a system for SLAM in structured environments using an AUV and 360-degree scanning imaging sonar. The sonar data from a each scan is motion-corrected using the IMU, meaning that short term drift is "baked in" to the measurements (as in our and previous systems). Walls are detected using line extraction on the motion-corrected sonar image. By assuming walls are vertical, they reduce the mapping problem to two dimensions. Vehicle and wall poses are tracked using an EKF.

Ozog et al. [65] use a piecewise planar assumption in a hull inspection task to generate constraints based on the four beams of the vehicle's DVL.

Pathak et al. [69] present a system for 6DOF scan matching and covariance estimation using planes extracted from 3D point clouds. The go on [68] to demonstrate its use in an online SLAM system for a mobile robot with a 3D laser scanner. They show a substantial benefit in both speed and accuracy over ICP alignment. The same authors point out that, while their and other techniques are successful with highly accurate laser rangefinders, there has been little evaluation of applying these techniques to noisier sensor data such as sonar. As a preliminary investigation, they apply their algorithm to data collected at a series of waypoints with a stationary multibeam sonar [67]. They show that offline registration of these planes can produce

a geometrically consistent map without odometry data, but do not demonstrate a full online SLAM system.

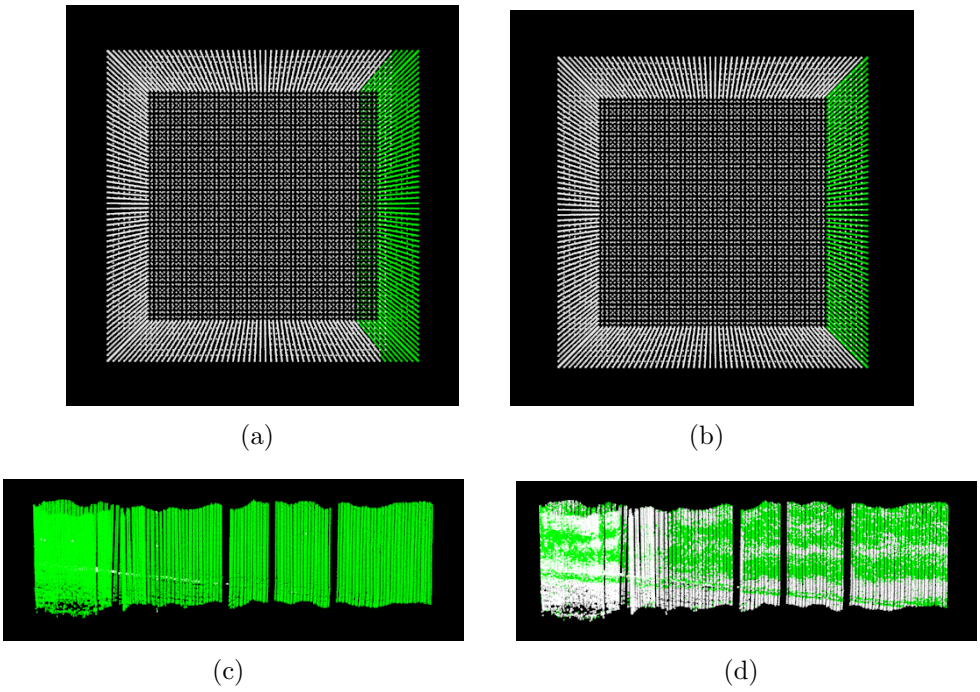
Trevor et al. [90] presents the planar SLAM technique most closely related to our work. The authors introduce a system to track vehicle poses and plane locations in a factor graph. They track convex hulls of planes and generated constraints when the measured planes overlap.

These previous efforts suggest that when planar features are present they can be leveraged to improve performance of SLAM. The benefits of planar alignment coincide nicely with the drawbacks of sonar. Plane extraction increases robustness when aligning noisy data, and sonar data is particularly noisy. Planar alignment can produce loop closure constraints even with little overlap between measurements, as is often the case when using profiling sonar.

5.1.2 Plane fitting

For each submap, we attempt to fit a plane with the well known RANSAC algorithm, implemented in PCL [80]. RANSAC begins by randomly selecting a set of points to be inliers. It then fits a least squares model to the inliers, and recomputes the set of inliers based on the model. This is repeated to improve the model. If RANSAC finds a model that has more than k^{RANSAC} inliers, and if these inliers make up more than α^{RANSAC} of the point cloud, we keep the plane. We use an absolute threshold to ensure that the fit is good, and the relative threshold to ensure that we're actually imaging a plane. If we're not imaging a plane, we expect loop closures to come from submap alignment.

RANSAC works best when there is only one model present in the point cloud. However, in our case, we may be imaging e.g. a corner, so a single submap may contain two planes. (In fact, this is the best way to constrain plane-plane angles). Rather than explicitly searching for multiple planes in the full submap, we simply remove the inliers after fitting each plane, and attempt the model fitting procedure again. In conjunction with tight thresholds k^{RANSAC} and α^{RANSAC}, this is effective at fitting multiple planes without excessive false positives.



(a) (b)

(c) (d)

Figure 5-1: Results of fitting a plane with different inlier distance thresholds. 5-1(a) and 5-1(b) show the inliers (green) resulting from fitting a plane to a synthetic point cloud of a cube. 5-1(c) and 5-1(d) show the results of fitting a plane to real data collected from a sonar scan of a swimming pool. To high of an inlier distance threshold (5-1(a)) results in a less accurate fit due to the inclusion of points that are not part of the fitted plane. Too low of an inlier threshold (5-1(d)) results in a less accurate fit due to the exclusion of points that belong to the plane. The threshold should match the noise characteristics of the data: tight thresholds for highly accurate data (5-1(b)), and loose thresholds for noisier data (5-1(c)).

Although RANSAC is natively designed to fit noisy data, we fit planes to the smoothed submaps. The differences between fitting smoothed and unsmoothed submaps were minor in our testing, but there were two advantages to fitting the smoothed submaps. The first benefit is uniform density and outlier rejection, which lead to a better fit. The second benefit of fitting to smoothed submaps is that it allowed us to use a very small inlier distance threshold for RANSAC. A large threshold is generally not a problem when fitting a single plane, but when fitting two intersecting planes, a large threshold causes the first extracted plane to include points from the second plane, and slightly bias the fit in that direction.

After we fit a plane, we have its equation in the local frame of the anchor pose of the associated submap.

We specify a plane using Hessian Normal Form, as a vector $\mathbf{n}$ and distance to origin d in the local frame of the vehicle. The i^{th} plane in submap t is $e_{ti} = [\mathbf{n}, d]$. Because this representation is symmetric, that is, $[\mathbf{n}, d]$ represents the same plane as $[-\mathbf{n}, -d]$, we use the convention that d is positive. For each detected plane, we also store the centroid $\mathbf{c}_{ti}$ of the inliers.

5.1.3 Planar constraints

We then have to determine whether this is a repeat or new observation. We get the updated pose $\hat{\mathbf{x}}_t$ from iSAM (see fig.). From this, we compute the homogeneous 4x4 transformation matrix $T_{\hat{\mathbf{x}}}$ of the vehicle coordinate frame in the global coordinate frame. We then transform the centroid $\mathbf{c}^{local} = [x, y, z, 1]^T$ into the global frame using $\mathbf{c}^{global} = T_{\hat{\mathbf{x}}}\mathbf{c}^{local}$ And transform the normal similarly to find $\mathbf{n}^{global}$. The distance to origin in the global frame is then $d^{global} = \mathbf{n}^{global} \cdot \mathbf{p}$ for any point $\mathbf{p}$ on the plane.

We define a simple distance metric based on the weighted sum of (1) the distance from each centroid to the other plane and (2) the difference in normal vectors. The distance between planes c_i and e_j is defined as:

$$f(e_{ti}, e_j) = \alpha^p(\mathbf{n}_i \cdot \mathbf{c}_j + \mathbf{n}_j \cdot \mathbf{c}_i) + \beta^p||\mathbf{n}_i - \mathbf{n}_j|| \tag{5.1}$$

For plane e_{ti}, we search all candidate planes to find the closest plane e_j. If $f(e_{ti}, e_j)$ is less than the threshold τ^p, we consider e_{ti} and e_j to represent observations of the same plane.

This is effectively a thresholded Nearest-Neighbor technique, common in tracking and mapping problems [6]. A more sophisticated approach such as the Probabilistic Data Association Filter, Joint Compatibility Branch and Bound [61], or plane-specific convex hull tracking [90] could possibly provide improved performance. For the purposes of this work, we have found our simple technique sufficient for tracking and disambiguating planes.

Because our data association algorithm does not consider spatial overlap, there is no requirement that two observations of a plane overlap for them to be associated. The

ability to generate loop closure constraints without overlap was an important design goal of our system. This is particularly beneficial for sonar inspection tasks, which generally have less overlap than e.g. terrestrial mapping with an RGB-D camera. As a consequence, our system does not distinguish between "coplanar" surfaces, for example tables of the same height, which would have a distance of $f(table_i, table_j) = 0$ using our distance metric.

When a loop closure is detected, we add a node to the factor graph representing the plane, and a factor representing the constraint.

For a list of parameters used in plane fitting, see Table 1 in the appendix.

5.1.4 Relation to previous work

Ours is not the first system for mapping using planar constraints. However, our work is distinguished from other efforts by the requirements of our inspection task—specifically, that we produce an online estimate of the *full vehicle trajectory* with correct linearization points, and create a *full 3D* map of the environment using *noisy sonar data*.

Our work is related to recent developments in terrestrial mapping, particularly those by Trevor et al. [90]. Terrestrial applications have the advantage of RGB-D cameras, which allow planes to be extracted from a single frame, thereby greatly simplifying the data association problem. RGB-D cameras and laser rangefinders are much more accurate than sonar units, and provide data with greater overlap at higher rates. We have introduced techniques for submap tracking and smoothing that have allowed us to use these state of the art techniques with only the limited sensors available to an underwater vehicle.

Some prior work in the underwater domain has assumed vertical walls, thereby effectively reducing the problem to two dimensions [73]. Other systems for underwater 3D mapping with planar constraints are batch processed offline [67], which would not be suitable for our application because we require an updated position estimate for planning and collision avoidance. Online SLAM using filtering approaches [74] can provide an estimate of the current vehicle position, but does not accurately estimate

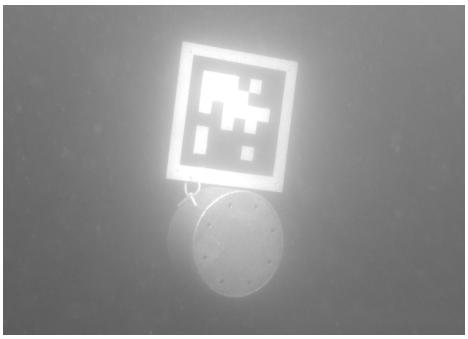


Figure 5-2: An AprilTag on the running gear of the SS Curtiss, San Diego, CA

the past trajectory, which we require to ensure full coverage of the inspection target. Our pose graph approach provides a 6DOF estimate of the full vehicle trajectory, and the efficient least-squares techniques used in iSAM allow real-time operation.

5.2 AprilTags

We use AprilTags [64] to stand in for regular visual features that we expect to encounter on ship hulls and other inspection targets.

The HAUV carries two Prosilica 12-bit cameras in custom waterproof housings: the "periscope" camera mounted on top of the vehicle, and the "underwater" camera mounted on the front basket. When the vehicle is at the surface, we use the periscope camera, otherwise we use the underwater camera. Camera images are sent uncompressed over LCM, rectified, and passed to the AprilTag detector.

We first rectify the images using OpenCV [72], correcting for two types of distortion: radial and tangential. Radial distortion, so called because it is radially symmetric, is a change the apparent size of objects as they move away from the *principal point*, or focal center, of the image. Radial distortion is common in camera lens systems, and is particularly noticeable in wide angle "fisheye" lenses. Tangential distortion is a dilation or compression of apparent size along the vertical or horizontal axis of the image. In our case, a slight misalignment between the camera lens and the transparent plate of the underwater housing will cause tangential distortion due to nonuniform diffraction, akin to looking into a pool of water from the water's edge. We

generate a simple quadratic model for radial and tangential distortion using OpenCV, by imaging a checkerboard pattern at various distances and positions in the camera frame. This gives us the principal points and distortion coefficients, which are used to remove distortion artifacts from images before visual fiducial marker detection.

The AprilTags library first calculates corner positions using fast subsampling, and detects candidate tag matches using line and quadrilateral detection. Once the 2D coordinates of the tag corners are detected, the library computes a 3x3 homography matrix that projects the tag's coordinates into the image coordinates. The 3x3 homography matrix is written as product of 3x4 camera matrix P and 4x3 truncated extrinsics matrix E. The extrinsics matrix is usually a full 4x4 transformation matrix, but in this case, because all of the points on the tag surface are at z=0, it is possible ot used a reduced representation. This produces an equation with an unknown scale factor, which is solved using the focal length of the camera, giving the tag's 6DOF position relative to the camera.

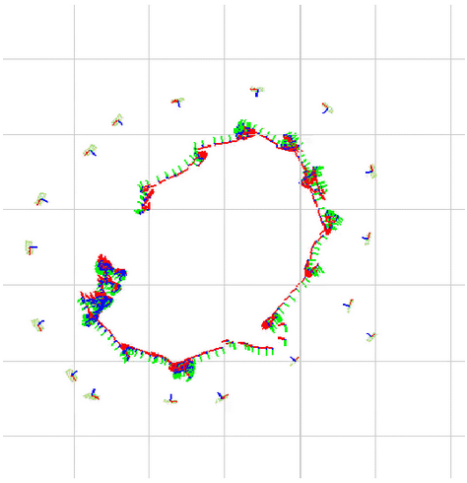
The relative pose from the camera to the tag is given in camera coordinates, where z represents distance from the camera. To use the AprilTag detections as loop closures, we must translate tag detections from the camera frame to the vehicle frame, and compensate for the offset between camera position and the vehicle's coordinate center.

We then add a pose-landmark constraint to iSAM representing the tag detection.

5.3 Evaluation

To test the effect of structured loop closures, we evaluated our system in two indoor experiments. We tested AprilTag-only localization using ground robots, and we tested the combined system of AprilTags and planar constraints using the HAUV in a controlled pool experiment. Below, we present the results of these two tests.

(a) TurtleBot in AprilTag array



(b) TurtleBot and AprilTag positions from iSAM

5.3.1 Wheeled ground robots

To test our implementation of AprilTag localization, we ran an experiment on a simple wheeled robots called TurtleBot, made by ClearPath Robotics. The TurtleBots were equipped with Kinect RGB-D cameras, although only the RGB channels were used for tag detection and localization. The robot performed SLAM using AprilTags and iSAM, with no wheel odometry, and were able to localize within an array of AprilTags. The tag setup is shown in Fig. 5-3(a), and the resulting vehicle and tag poses are shown in Fig. 5-3(b).

5.3.2 HAUV

To evaluate the effect of planar constraints in isolation, we operated the HAUV in the 25-meter MIT Alumni Pool using the DIDSON sonar and Prosilica camera. The pool is approximately 5m deep at one end, with a sloping floor beginning at approximately the midpoint and ending at the opposite end at approximately 2m depth. The sonar was fixed in the "half-split" configuration, in which beams are oriented in a vertical fan, rather than the horizontal orientation used for inspection tasks. In this configuration, the sonar could image the vertical pool walls, as well as the sloped floor of the shallow end. We mounted 4 AprilTags on the walls along the length of the pool, which were visible from the vehicle's periscope camera (Figure 5-3).

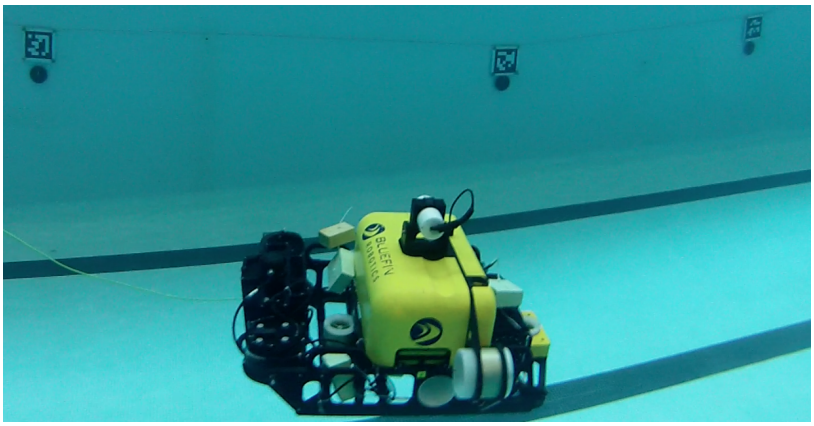


Figure 5-3: HAUV imaging AprilTags in MIT Alumni Pool

The vehicle performed laps of the four walls, at a 2-4m standoff, for approximately 1 hour. The DVL maintained 4-beam bottom lock throughout the course of the experiment, resulting in a dead reckoning position estimate with relatively less drift than the *Saratoga* experiments. Planar SLAM constraints were generated by observations of the four walls and sloped floor of the pool; the locations of these five landmarks were jointly optimized with the vehicle trajectory in iSAM.

A comparison of the dead reckoning and SLAM trajectories can be seen in Figure 5-4. The dead reckoning trajectory exhibits heading drift on the order of 10°, which is corrected in the SLAM solution. Planar constraints proved well suited to correcting heading drift, by explicitly constraining the relative orientation of the vehicle and landmarks.

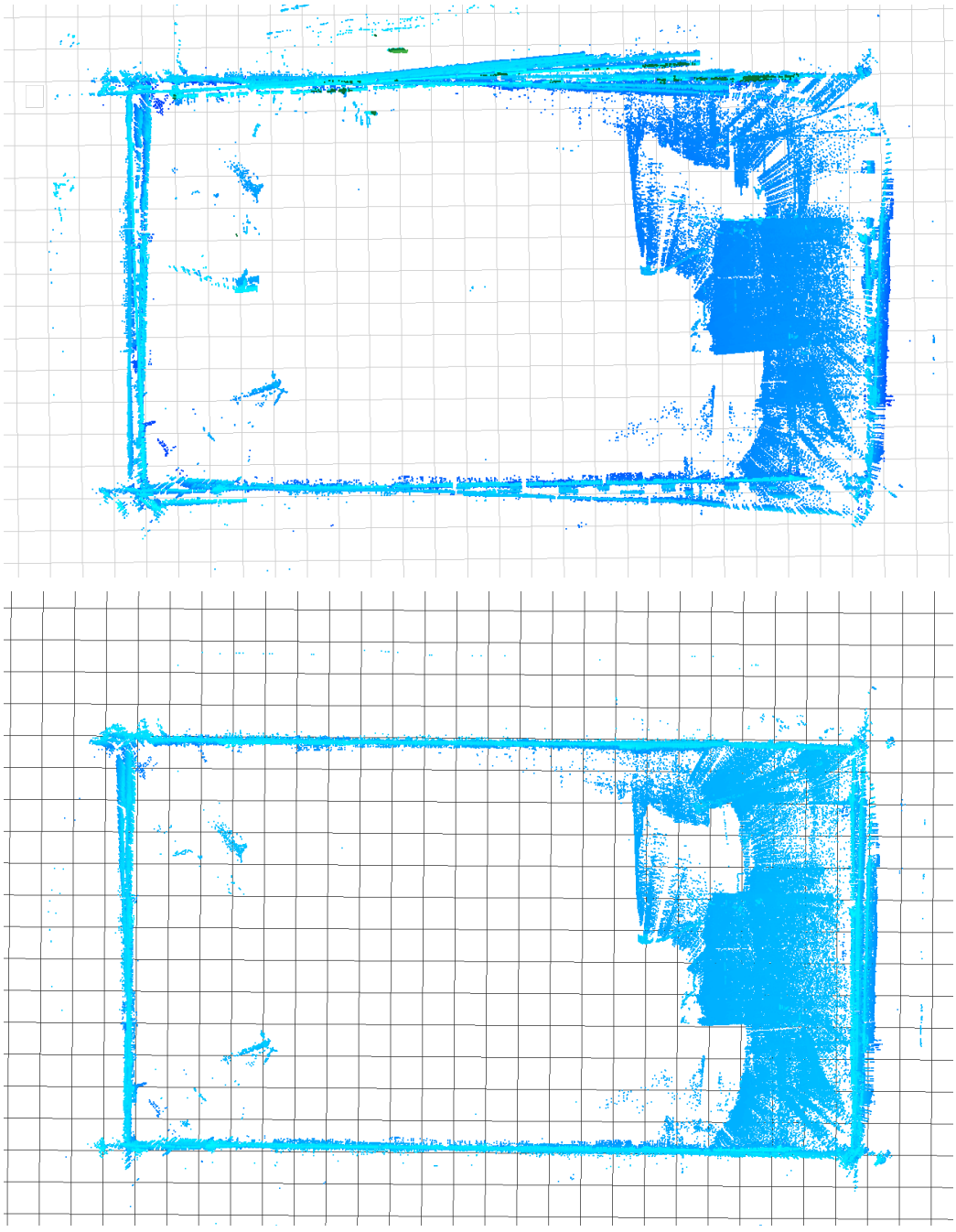


Figure 5-4: Orthographic projection of pool map produced by dead reckoning (top) and SLAM (bottom) trajectories. The SLAM map contained five planes: the four walls of the pool, and the sloped floor that appears in the right side of these figures.

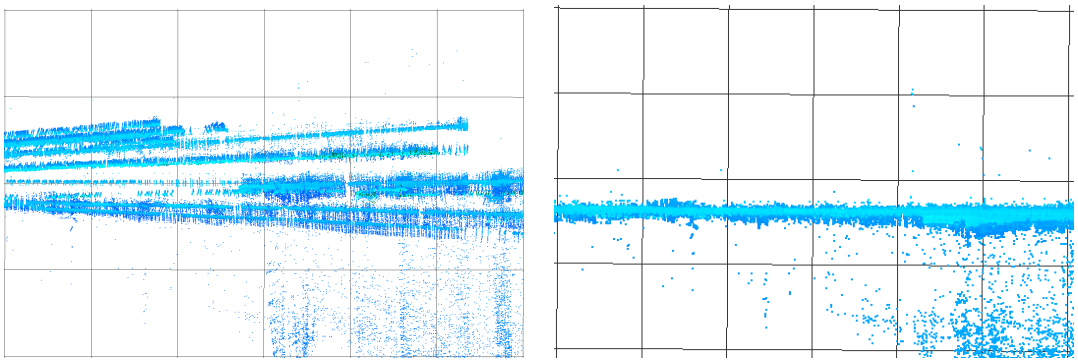


Figure 5-5: Detail of a wall section in pool experiment. Dead reckoning (top) exhibits heading errors, which are corrected in the SLAM solution (bottom).

Chapter 6

Conclusion

Recent advances in underwater perception, navigation, and planning suggest that autonomous inspection of underwater infrastructure could be feasible in the near future. High resolution sonar units, carried on AUVs designed for complex area inspection tasks, provide centimeter-level accuracy at near video frame rates. Modern approaches to SLAM allow efficient online mapping for long term operation, as would be required in a repeated inspection scenario. Techniques adapted from terrestrial coverage planning allow us to plan collision-free trajectories to inspect complex 3D structures.

Our work builds upon the current state of the art in each of these areas to create a system for underwater inspection, and introduces novel techniques which can be applied beyond the underwater domain.

6.1 Review of contributions

At a high level, our contribution is a *system for visual-acoustic SLAM in partially structured 3D environments*. The specific contributions are (1) a framework for multi-sensor factor graph SLAM, (2) an algorithm for denoising and surface modeling in sonar data, (3) a catalog for tracking and aligning sonar submaps, and (4) a technique for tracking planar features in a factor graph.

Our mapping framework combines data from the HAUV's dead reckoning sys-

tem with loop closures generated from camera and sonar data. It supports multiple loop closure modalities, including pose-to-pose, pose-to-6DOF landmark, and pose-to-plane. The resulting factor graph is optimized efficiently using iSAM, and runs easily within our time constraints.

Our sonar processing technique utilizes the full 3D scene geometry, rather than the 2D range image or 2.5D topological denoising common in bathymetric mapping. We perform statistical outlier rejection and multi-step downsampling tuned to reject common sonar errors, along with parametric surface modeling to reconstruct the inspection target, and demonstrate the effectiveness of these techniques in both field experiments and synthetic data sets.

Our submap catalog tracks and aligns submaps, generating constraints for our SLAM framework. This is inspired by previous work in bathymetric mapping, but built around 3D point clouds, rather than elevation maps, to capture the complex geometry of underwater structures.

We introduce a technique for extracting planar features from sonar data, generating pose-relative constraints from these features, and optimizing the plane equation directly within the SLAM framework. To our knowledge, this is the first such system for jointly optimizing full 6DOF vehicle trajectory and plane location using constraints derived from a mapping sonar.

6.2 Future work

We hope to investigate several directions of potential future work. These include alternative surface modeling techniques and submap representations, more complex geometric shapes in our factor graph framework, and parametric primitives beyond simple geometric shapes.

Our current submap smoothing pipeline is based on 3D point clouds, which, compared to volumetric techniques, have the advantage of being arbitrarily complex, because smaller features can be represented by simply adding points at the appropriate level of detail, and easily modifiable, enabling outlier rejection and navigation

corrections to be incorporated into the submap via simple add, delete, and transform operations. However, volumetric techniques such as the Truncated Signed Distance Function (TSDF) have the advantages of native denoising, which mitigates the effect of both small errors and outliers through averaging, and of directly representing a closed surface, which is well suited to the ray-tracing operation used in our coverage planning framework. Much of the improved performance of the TSDF can be attributed to its use of sensor viewpoint in tracking free and occupied space (in contrast, our point cloud smoothing is agnostic to sensor viewpoint). We hope to investigate a hybrid system which provides the flexibility of point cloud submaps with the native surface smoothing and modeling of the TSDF.

Preliminary investigation suggests two possible approaches to combining volumetric techniques with our current unstructured point cloud submaps. The first is to use a high resolution TSDF for each submap, attached to an anchor pose as in the current system, but to avoid combining data from different submaps. This would allow easy map updates (by simply changing the estimated position of the anchor pose), but, by keeping submaps separate, would not fully utilize the native denoising that results from combining multiple overlapping readings in the TSDF. The second is to perform alignment and map updates using the current point cloud submaps, but to maintain a second volumetric representation of the full scene, which could be updated asynchronously after each update of the anchor poses. Although the TSDF is designed to be implemented efficiently, recomputing the entire volume on each update could be expensive. A possible solution would be to tag areas of the volumetric representation with the submaps from which their data originated, and only update the areas for which the location of associated anchor poses has changed significantly.

We are also interested in tracking more complex geometric shapes in our factor graph. For example, cylinders are a reasonable approximation of many wooden support beams, and rectangular prisms are a reasonable approximation of man concrete bridge pilings. We hope that tracking these features would provide many of the same benefits we saw in our planar SLAM system: improved reconstruction from fewer measurements, jointly optimized estimates of vehicle and feature locations, and

higher-level representations that are more natively useful to both human and robotic operators and planners.

Adding features to our current framework requires (1) a mathematical representation of the feature location and parameters which can be efficiently optimized using least squares techniques, and (2) a measurement model, and associated cost and covariance, that represents an observation of the feature from a vehicle pose as a pose-feature constraint.

For the feature representation (1), simple equations for the shape in Euclidean 3-space could suffice, although they are often overparameterized. A better solution would be to create an exponential map, which reduces the number of free parameters and often results in a smoother cost function which is better suited to gradient descent optimization.

The measurement model (2) encodes the likelihood of a given observation as a function of vehicle and landmark position. A simple measurement model for a parametric feature is to measure distance in the *parameter space* of the feature's equation in the local vehicle frame. This is our current approach for planar measurements. However, if the feature equation is overparameterized (as is the case with Euclidean representations of many shapes) distance in parameter space is a poor choice, because identical shapes could have nonzero distance in parameter space.

The measurement model also entails accurately estimating the noise characteristics of the observation. With simple sensors, such as laser range measurements, the measurement noise can be derived directly from the sensor geometry and measurement properties (e.g. the noise of a sensor measuring photon arrivals will approach a normal distribution, due to the law of large numbers.) Sonar data provides no such obvious model, as it is distorted by unknown environmental factors, and heavily processed both within the sonar unit and during our range extraction, smoothing, and shape fitting procedures. Therefore, it is incorrect to assume Gaussian noise in the range image. The consequences of an incorrect noise model could become amplified when extracting features such as parametric surfaces from the sensor data. In formulating our measurement model for parametric shapes, we may benefit from using

non-Gaussian error models following Rosen et. al. [79], which have been optimized efficiently using trust region methods [78] [40].

Because many inspection targets have components with a known geometry, such as I-beams, lattice beams, or bilge outlets, we see potential for further benefit in tracking the locations and parameters of these higher-level features in the same way we track low-level geometric primitives such as planes or cylinders. In addition to providing the benefits of geometric feature tracking — creating better maps from sparser data – the error residuals in the optimized factor graph could provide an indication of to what degree certain features deviate from their design parameters. In this formulation, structural deterioration from corrosion and deformation could be inferred *directly from the factor graph itself*, providing an indication of which elements require further inspection and testing.

k_{sub}	500	Submap formation: min. points per submap
t_{sub}	25 sec	Submap formation: max. time per submap
v_1	2 cm	Voxel filter 1: size
k_{nn}	50	Outlier rejection: # neighbors considered
σ_{nn}	2	Outlier rejection: std. dev. limit
v_2	10 cm	Voxel filter 2: size
r	0.3 m	Polynomial surface modeling: radius
α	.1	Submap alignment: fitness threshold
d^{ICP}	.5 m	ICP max. correspondence distance
ϵ^{ICP}	10^{-8}	ICP fitness epsilon
τ^{ICP}	.05	ICP fitness threshold
d^{RANSAC}	.1 m	Plane fitting: inlier distance threshold
k^{RANSAC}	350 pts	Plane fitting: min. points per plane
α^{RANSAC}	.4	Plane fitting: min. fraction of points per plane
α^p	1.0	Distance metric: centroid distance weight
β^p	20.0	Distance metric: normal distance weight
τ^p	5.0	Distance metric: association threshold

Table 1: Key parameters for submap formation, smoothing, and alignment.

Bibliography

[1] Underwater Inspection and Evaluation of New Jersey Bridges. Technical report, New Jersey Department of Transportation, June 1994.

[2] Waterfront Facilities Maintenance Management System. Technical report, New York City Economic Development Corporation, October 1999.

[3] ABS guide for building and classing gravity-based offshore LNG terminals. Technical report, American Bureau of Shipping, June 2010.

[4] ABS hull inspection and maintenance program. Technical report, American Bureau of Shipping, December 2013.

[5] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva. Computing and rendering point set surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 9(1):3 – 15, January 2003.

[6] Y. Bar-Shalom and T. Fortmann. *Tracking and Data Association*. Mathematics in Science and Engineering. Academic Press Professional, Inc., San Diego, CA, USA, 1988.

[7] S. Barkby, S. Williams, O. Pizarro, and M. Jakuba. An efficient approach to bathymetric SLAM. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, October 2009.

[8] C. Beall, F. Dellaert, I. Mahon, and S. Williams. Bundle adjustment in large-scale 3D reconstructions based on underwater robotic surveys. In *Proc. of the IEEE/MTS OCEANS Conf. and Exhibition*, 2011.

[9] E. Belcher, W. Hanot, and J. Burch. Dual-frequency identification sonar (DIDSON). In *Proceedings of the 2002 International Symposium on Underwater Technology*, pages 187–192, 2002.

[10] Paul J. Besl and Neil D. McKay. A method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Machine Intell.*, 14(2):239–256, February 1996.

[11] D. Borrmann, J. Elseberg, K. Lingemann, A. Nuchter, and J. Hertzberg. Globally consistent 3D mapping with scan matching. *J. of Robotics and Autonomous Systems*, 56(2):130–142, 2008.

[12] A. Bowen, M. Jakuba, D. Yoerger, C. German, J.C. Kinsey, L. Whitcomb, and L. Mayer. Lightly tethered unmanned underwater vehicle for under-ice exploration. In *IEEE Aerospace Conference*. IEEE, 2012.

[13] T.M. Browne, T.J. Collins, M.J. Garlich, J.E. OLeary, D.G. Stromberg, and K.C. Heringhaus. Underwater Bridge Inspection. Technical Report FHWA-NHI-10-027, U.S. Department of Transportation Federal Highway Administration, 2010.

[14] C. Cabos, D. Jaramillo, G. Stadie-Frohbos, P. Renard, M. Ventura, and B. Dumas. Condition assessment scheme for ship hull maintenance. In *Proc. of Intl. Conference on Computer Applications and Information Technology in the Maritime Industries (COMPIT)*, April 2008.

[15] R. Camilli, C.M. Reddy, D.R. Yoerger, B. Van mooy, M.V. Jakuba, J.C. Kinsey, C.P. McIntyre, S.P. Sylva, and J.V. Maloney. Tracking hydrocarbon plume transport and biodegradation at deepwater horizon. *Science*, 330(6001):201–204, 2010.

[16] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *SIGGRAPH*, pages 303–312, August 1996.

[17] F. Dellaert and M. Kaess. Square Root SAM: Simultaneous localization and mapping via square root information smoothing. *Intl. J. of Robotics Research*, 25(12):1181–1203, December 2006.

[18] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57, June 1989.

[19] W.J. Emery and R.E. Thomson. *Data analysis methods in physical oceanography*. Elsevier Science, 2001.

[20] B. Englot and F. Hover. Inspection planning for sensor coverage of 3D marine structures. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2010.

[21] B. Englot and F. Hover. Planning complex inspection tasks using redundant roadmaps. In *Proc. of the Intl. Symp. of Robotics Research (ISRR)*, 2011.

[22] B. Englot and F. Hover. Sampling-based coverage path planning for inspection of complex structures. In *Proc. of Intl. Conf. on Automated Planning and Scheduling*, 2012.

[23] R. Eustice, H. Singh, J. Leonard, M. Walter, and R. Ballard. Visually navigating the RMS Titanic with SLAM information filters. In *Robotics: Science and Systems (RSS)*, June 2005.

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

[25] M.F. Fallon, H. Johannsson, and J.J. Leonard. Efficient scene simulation for robust Monte Carlo localization using an RGB-D camera. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 1663–1670, St. Paul, MN, May 2012.

[26] J. Folkesson and J. Leonard. Autonomy through SLAM for an underwater robot. In *Proc. of the Intl. Symp. of Robotics Research (ISRR)*, 2009.

[27] M.P. Gerardo-Castro, T. Peynot, and F. Ramos. Laser-radar data fusion with gaussian process implicit surfaces.

[28] E. Geyer, P. Creamer, J. D'Appolito, and R. Gains. Characteristics and capabilities of navigation systems for unmanned untethered submersibles. In *Proc. Int. Symp. on Unmanned Untethered Submersible Technology*, pages 320–347, 1987.

[29] D. B. Heckman and R. C. Abbott. An acoustic navigation technique. In *IEEE Oceans '73*, pages 591–595, 1973.

[30] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. pages 71–78, 1992.

[31] T Hou, L C Huff, and L Mayer. Automatic detection of outliers in multibeam echo sounding data. *US HYDRO*, 1, 2001.

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

[33] A. Huang, E. Olson, and D. Moore. Lightweight communications and marshalling for low-latency interprocess communication. Technical Report MIT-CSAIL-TR-2009-041, Computer Science and Artificial Intelligence Laboratory, MIT, September 2009.

[34] M. Hunt, W. Marquet, D. Moller, K. Peal, W. Smith, and R. Spindel. An acoustic navigation system. Technical Report WHOI-74-6, Woods Hole Oceanographic Institution, 1974.

[35] S. Izadi, R.A. Newcombe, D. Kim, O. Hilliges, D. Molyneaux, S. Hodges, P. Kohli, J. Shotton, A.J. Davison, and A. Fitzgibbon. KinectFusion: Real-time dynamic 3D surface reconstruction and interaction. In *SIGGRAPH*, page 23, August 2011.

[36] M.V. Jakuba, D. Steinber, J.C. Kinsey, D. Yoerger, R. Camilli, O. Pizarro, and S.B. Williams. Toward automatic classification of chemical sensor data from autonomous underwater vehicles. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 4722–4727. IEEE, 2011.

[37] H. Johannsson. *Toward Lifelong Visual Localization and Mapping*. PhD thesis, Computer Science and Artificial Intelligence Laboratory, MIT, 2013.

[38] Hordur Johannsson. Toward autonomous harbor surveillance. Master's thesis, Massachusetts Institute of Technology, 2010.

[39] A. Johnson and M. Hebert. Refinement of seafloor elevation using acoustic backscatter. Technical Report CMU-RI-TR-95-01, Robotics Institute, Pittsburgh, PA, March 1995.

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

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

[42] R. Kalman. A new approach to linear filtering and prediction problems. *ASME Journal of Basic Engineering*, 1960.

[43] S.W. Kelley. Underwater inspection criteria. Technical Report March, Naval Facilities Engineering Service Center, 1999.

[44] A. Kim. *Active Visual SLAM with Exploration for Autonomous Underwater Navigation*. PhD thesis, The University of Michigan, 2012.

[45] A. Kim and R.M. Eustice. Next-best-view visual SLAM for bounded-error area coverage. In *IROS Workshop on Active Semantic Perception*, 2012.

[46] A. Kim and R.M. Eustice. Perception-driven navigation: Active visual SLAM for robotic area coverage. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2013.

[47] A. Kim and R.M. Eustice. Real-time visual SLAM for autonomous underwater hull inspection using visual saliency. *IEEE Trans. Robotics*, 2013. To appear.

[48] C. Kunz, C. Murphy, R. Camilli, H. Singh, J. Bailey, R. Eustice, M. Jakuba, K. Nakamura, C. Roman, T. Sato, R.A. Sohn, and C. Willis. Deep sea underwater robotic exploration in the ice-covered arctic ocean with auvs. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 3654–3660. IEEE, 2008.

[49] C. Kunz and H. Singh. Map building fusing acoustic and visual information using autonomous underwater vehicles. *J. of Field Robotics*, 30(5):763–783, 2013.

[50] J.C. Lanzoni and T.C. Weber. High-resolution calibration of a multibeam echo sounder. In *Proc. of the IEEE/MTS OCEANS Conf. and Exhibition*, pages 1–7. IEEE, 2010.

[51] T. Lee, S. Lim, S. Lee, S. An, and S. Oh. Indoor mapping a planes extracted from noisy RGB-D sensors. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 1727–1733, 2012.

[52] J. J. Leonard, A. A. Bennett, C. M. Smith, and H. J. S. Feder. Autonomous underwater vehicle navigation. Technical Report Marine Robotics Laboratory Technical Memorandum 98-1, MIT, 1998.

[53] CB Lirakis and KP Bongiovanni. Automated multibeam data cleaning and target detection. In *Proc. of the IEEE/MTS OCEANS Conf. and Exhibition*, volume 1, pages 719–723. IEEE, 2000.

[54] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004.

[55] D.G. Lowe. Object recognition from local scale-invariant features. In *International Conference on Computer Vision*, volume 2, pages 1150–1157. Kerkyra, Greece, 1999.

[56] M.D. McRary, D.E. Panzer, and M.O. Pierson. Oil and gas operations offshore California: status, risks, and safety. *Marine Ornithology*, (31):43 – 49, 2003.

[57] W. McVicker, J. Forrester, T. Gambin, J. Lehr, Z.J. Wood, and C.M. Clark. Mapping and visualizing ancient water storage systems with an ROV – an approach based on fusing stationary scans within a particle filter. In *IEEE Intl. Conf. on Robotics and Biomimetics*, pages 538–544, 2012.

[58] H. Medwin and C. S. Clay. *Fundamentals of Acoustical Oceangraphy*. Academic Press, San Diego, CA, USA, 1998.

[59] N.C. Mitchell. Processing and analysis of Simrad multibeam sonar data. *Marine Geophysical Researches*, 18(6):729–739, 1996.

[60] Christian Moustier and Martin C Kleinrock. Bathymetric artifacts in Sea Beam data: How to recognize them and what causes them. *Journal of Geophysical Research*, 91(B3):3407–3424, 1986.

[61] J. Neira and J. D. Tardos. Data association in stochastic mapping using the joint compatibility test. *IEEE Trans. Robotics and Automation*, 17(6):890–897, December 2001.

[62] R. A. Newcombe, A. J. Davison, S. Izadi, P. Kohli, O. Hilliges, J. Shotton, D. Molyneaux, S. Hodges, D. Kim, and A. Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. In *IEEE and ACM Intl. Sym. on Mixed and Augmented Reality (ISMAR)*, pages 127–136, Basel, Switzerland, October 2011.

[63] A. Nüchter and J. Hertzberg. Towards semantic maps for mobile robots. *J. of Robotics and Autonomous Systems*, 56(11):915–926, 2008.

[64] E. Olson. Apriltag: A robust and flexible visual fiducial system. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, May 2011.

[65] P. Ozog and R.M. Eustice. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2013.

[66] J. G. Paglia and W. F. Wyman. DARPA's autonomous minehunting and mapping technologies (AMMT) program: An overview. In *IEEE Oceans*, pages 794–799, Ft. Lauderdale, FL, USA, September 1996.

[67] K. Pathak, A. Birk, and N. Vaskevicius. Plane-based registration of sonar data for underwater 3D mapping. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, Taipei, Taiwan., October 2010.

[68] K. Pathak, A. Birk, N. Vaskevicius, M. Pfingsthorn, S. Schwertfeger, and J. Poppinga. Online three-dimensional SLAM by registration of large planar surface segments and closed-form pose-graph relaxation. *J. of Field Robotics*, 27(1):52–84, January 2010.

[69] K. Pathak, N. Vaskevicius, J. Poppinga M. Pfingsthorn, S. Schwertfeger, and A. Birk. Fast 3D mapping by matching planes extracted from range sensor point-clouds. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, St. Louis, USA, October 2009.

[70] L. Paull. *Robust Online Adaptive Sensor-Driven Survey Planning for Single and Multiple Autonomous Underwater Vehicles*. PhD thesis, Fredericton: University of New Brunswick, November 2013.

[71] N.S. Potty, B.M. Akram, and M. Kabir. Structural integrity management for fixed offshore platforms in malaysia. *World Academy of Science, Engineering and Technology*, (34), 2009.

[72] Kari Pulli, Anatoly Baksheev, Kirill Kornyakov, and Victor Eruhimov. Real-time computer vision with opencv. *Commun. ACM*, 55(6):61–69, June 2012.

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

[74] 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.

[75] C. Roman, G. Inglis, and J. Rutter. Application of structured light imaging for high resolution mapping of underwater archaeological sites. In *Proc. of the IEEE/MTS OCEANS Conf. and Exhibition*. IEEE, 2010.

[76] C. Roman and H. Singh. Improved vehicle based multibeam bathymetry using sub-maps and SLAM. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 3662–3669, August 2005.

[77] C. Roman and H. Singh. A self-consistent bathymetric mapping algorithm. *J. of Field Robotics*, 24(1):23–50, 2007.

[78] D.M. Rosen, M. Kaess, and J.J. Leonard. An incremental trust-region method for robust online sparse least-squares estimation. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 1262–1269, St. Paul, MN, May 2012.

[79] D.M. Rosen, M. Kaess, and J.J. Leonard. Robust incremental online inference over sparse factor graphs: Beyond the Gaussian case. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Karlsruhe, Germany, May 2013.

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

[81] Radu Bogdan Rusu, Zoltan Csaba Marton, Nico Blodow, Mihai Dolha, and Michael Beetz. Towards 3D point cloud based object maps for household environments. *J. of Robotics and Autonomous Systems*, 56(11):927–941, 2008.

[82] R.F. Salas-Moreno, R.A. Newcombe, H. Strasdat, P. H. J. Kelly, and A. J. Davison. SLAM++: Simultaneous localisation and mapping at the level of objects. In *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, Portland, Oregon, June 2013.

[83] H. Singh, J. Catipovic, R. Eastwood, L. Freitag, H. Henriksen, F F. Hover, D. Yoerger, J. G. Bellingham, and B. A. Moran. An integrated approach to multiple AUV communications, navigation and docking. In *IEEE Oceans*, pages 59–64, 1996.

[84] M. J. Stanway. Water profile navigation with an acoustic doppler current profiler. In *Proc. of the IEEE/MTS OCEANS Conf. and Exhibition*, pages 1–5, 2010.

[85] S. Thrun, M. Beetz, M. Bennewitz, W. Burgard, AB Cremers, F. Dellaert, D. Fox, D. Hahnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. Probabilistic algorithms and the interactive museum tour-guide robot minerva. *Intl. J. of Robotics Research*, 19(11):972–999, 2000.

[86] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press, Cambridge, MA, 2005.

[87] S. Thrun, Y. Liu, D. Koller, A. Ng, Z. Ghahramani, and H. Durrant-Whyte. Simultaneous localization and mapping with sparse extended information filters. *Intl. J. of Robotics Research*, 23(7), 2004.

[88] F. Caratori Tontini, C. Ronde, D. Yoerger, J. Kinsey, and M. Tivey. 3-d focused inversion of near-seafloor magnetic data with application to the brothers volcano hydrothermal system, southern pacific ocean, new zealand. *Journal of Geophysical Research: Solid Earth (1978–2012)*, 117(B10), 2012.

[89] B.H. Tracey. Design and testing of an acoustic ultra-short baseline navigation system. Master's thesis, MIT, Cambridge, MA, September 1992.

[90] A.J.B. Trevor, J.G. Rogers, and H.I. Christensen. Planar surface SLAM with 3D and 2D sensors. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 3041–3048, May 2012.

[91] J. Vaganay, J. G. Bellingham, and J. J. Leonard. Outlier rejection for autonomous acoustic navigation. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 2174–2181, April 1996.

[92] S. Vasudevan, F.T. Ramos, E.W. Nettleton, and H.F. Durrant-Whyte. Gaussian process modeling of large scale terrain. *J. of Field Robotics*, 26(10):812–840, 2009.

[93] M. Walter, F. Hover, and J. Leonard. SLAM for ship hull inspection using exactly sparse extended information filters. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 1463–1470, May 2008.

[94] M.R. Walter. *Sparse Bayesian information filters for localization and mapping*. PhD thesis, Massachusetts Institute of Technology, 2008.

[95] T. Weyrich, M. Pauly, R. Keiser, S. Heinzle, S. Scandella, and M. Gross. Post-processing of scanned 3D surface data. In *Proc. IEEE Eurographics Symp. Point-Based Graphics*, pages 85–94, 2004.

[96] L. Whitcomb, D. Yoerger, and H. Singh. Advances in doppler-based navigation of underwater robotic vehicles. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, volume 1, pages 399–406, 1999.

[97] L. Whitcomb, D. Yoerger, and H. Singh. Combined Doppler/LBL based navigation of underwater vehicles. In *Proceedings of the International Symposium on Unmanned Untethered Submersible Technology (UUST)*, May 1999.

[98] O. Williams and A. Fitzgibbon. Gaussian process implicit surfaces. In *Proc. Gaussian Processes in Practice Workshop*, 2007.

[99] K. M. Wurm, A. Hornung, M. Bennewitz, C. Stachniss, and W. Burgard. OctoMap: A probabilistic, flexible, and compact 3D map representation for robotic systems. In *Proc. of the ICRA 2010 Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation*, Anchorage, AK, USA, May 2010.

[100] Y. Zhang, R.S. McEwen, J.P. Ryan, J.G. Bellingham, H. Thomas, C.H. Thompson, and E. Rienecker. A peak-capture algorithm used on an autonomous underwater vehicle in the 2010 gulf of mexico oil spill response scientific survey. *J. of Field Robotics*, 28(4):484–496, 2011.