Advanced Perception, Navigation and Planning for Autonomous In-Water Ship Hull Inspection

Franz S. Hover, Ryan M. Eustice, Ayoung Kim, Brendan Englot,
Hordur Johannsson, Michael Kaess, and John J. Leonard

Abstract

Inspection of ship hulls and marine structures using autonomous underwater vehicles has emerged as a unique and challenging application of robotics. The problem poses rich questions in physical design and operation, perception and navigation, and planning, driven by difficulties arising from the acoustic environment, poor water quality and the highly complex structures to be inspected. In this paper, we develop and apply algorithms for the central navigation and planning problems on ship hulls. These divide into two classes, suitable for the open, forward parts of a typical monohull, and for the complex areas around the shafting, propellers and rudders. On the open hull, we have integrated acoustic and visual mapping processes to achieve closed-loop control relative to features such as weld-lines and biofouling. In the complex area, we implemented new large-scale planning routines so as to achieve full imaging coverage of all the structures, at a high resolution. We demonstrate our approaches in recent operations on naval ships.

1 Introduction

Security of ship hulls and marine structures is a major concern to the naval community, operators of commercial vessels, and port authorities. Among objects that have been or could be attached to these structures are small mines of order ten centimeters scale; clearly the ability to detect such items quickly and cheaply is desirable. As one pressing example, hull inspections in a foreign port are sometimes triggered by the observance of air bubbles near the ship, which could signal an adversarial diver. Adversaries can be detected using passive and active sonar, as well as other means, but monitoring the space in this way alone does not guarantee that the hull is clear of threats.

Divers are of course the conventional method for in-water inspection of stationary structures; highly-skilled units in the U.S. Navy employ sonar cameras in poor water conditions, along with various navigation aids. However, divers are not always available on short notice. Among other approaches for inspection, marine mammals (Olds, 2003) such as sealions and dolphins easily exceed the speed and maneuverability of human divers. But the animals can be unreliable in finding or reporting small anomalies, and are expensive to maintain especially on an active vessel. A calibrated imaging system, attached to a fixed structure through which the vessel passes, could provide the necessary resolution via a combination of vision, laser and sonar imaging. We do not know of any fundamental issues that would deter such a development, but a system like this has not been commercialized at scale.

On the other hand, unmanned underwater vehicles have become extraordinarily capable today, following more than two decades of intense development by both the offshore oil and gas community and the U.S. Navy. Whereas remotely operated vehicles (ROVs) employ a tether for continuous power and high-capacity communications, autonomous underwater vehicles (AUVs) move freely and therefore usually require on-board algorithms for navigation, perception, and decision-making. A survey of autonomous vehicle technology as of 2000 was provided in (Yuh, 2000), and there are many additional publications on specific applications and vehicle systems as used in industry and the military. Oceanographers have also contributed to and used underwater robots for specialized tasks (for example, see (Ballard et al., 2002; Yoerger et al., 2007)), as well as untethered gliders in open water (Sherman et al., 2001; Webb et al., 2001). Among devices that have been developed for ship inspection and non-destructive testing of hulls, *Lamp Ray* (Harris and Slate, 1999; D'Amaddio et al., 2001) was an early hull-crawling ROV; there are also vehicles employing magnetic tracks (Carvalho

Draft manuscript, April 25, 2012.

Portions of this work have appeared previously in (Johannsson et al., 2010) and (Englot and Hover, 2011).

F. Hover, B. Englot, H. Johannsson, M. Kaess and J. Leonard are with the Massachusetts Institute of Technology (MIT), Cambridge, Massachusetts 02139, USA {hover, benglot, hordurj, kaess, jleonard}@mit.edu.

R. Eustice and A. Kim are with the University of Michigan, Ann Arbor, Michigan 48109, USA {eustice, ayoungk}@umich.edu.

et al., 2003; Menegaldo et al., 2008), related to others designed for tank and wall inspection. Off the hull, the AUV *Cetus II* was operated by Lockheed-Martin (Trimble and Belcher, 2002) and later by the Space and Naval Warfare Systems Command (SPAWAR). Navigation has been a primary operational theme in almost all of these systems, and a great majority of underwater vehicles—with perhaps the exception of gliders—have employed long-baseline, short-baseline or ultra-short-baseline acoustics as a non-drifting reference.

In 2002, MIT Sea Grant and Bluefin Robotics Corp. began a collaborative effort to develop an AUV system for precision (i.e., hover-capable) ship hull inspection, under Office of Naval Research (ONR) funding. The initial vehicle, first operated at-sea in 2005, established a new paradigm for navigation by using a Doppler velocity log (DVL) to lock onto the ship hull, maintaining distance and orientation, and with the same sensor to compute dead-reckoned distances in hull surface coordinates (Vaganay et al., 2005). Over the ensuing years, the Hovering Autonomous Underwater Vehicle, (HAUV, pronounced "H-A-U-V") has operated on more than thirty vessels, with substantial improvements in operations and algorithms; in March of 2011 Bluefin received a contract from the U.S. Navy for a fleet of production vehicles (Weiss, 2011). DVL-based dead-reckoning on the hull has remained an extremely robust and effective means for covering large open areas.

This paper presents our recent accomplishments in the application of advanced robotics capabilities to the HAUV and its hull inspection mission, building on the core technologies above. The first focus is feature-based navigation, in which we take advantage of the fact that the environment itself can be an effective reference: distinct physical objects that are detected and recognized by the vehicle enable simultaneous localization and mapping, or SLAM (Bailey and Durrant-Whyte, 2006; Durrant-Whyte and Bailey, 2006). On a ship hull, we regularly observe bolts, protrusions, holes, weld lines and biofouling, both above and below water. Feature-based navigation can seamlessly complement a dead-reckoning sensor stream, to create a high-quality stable and drift-free position estimate.

A second major thrust is path planning. The running gear in particular presents a complicated non-convex, three-dimensional structure, whose overall scale is much larger than a typical AUV and its field of view. Yet the structure can also have tight areas, such as between the shafting and the hull, where the vehicle physically cannot go, and where the viewpoint has to be carefully planned. The sensor trajectory has to guarantee complete coverage of the structure, and should achieve it with a short dive time. In both navigation and planning, we are sometimes completely dependent on acoustic imaging due to poor water quality, while at other times a visual camera can be extremely effective.

We begin in Section 2 with a brief description of the physical vehicle and its major components, as well as the overall concept of operations. In Sections 3, 4, and 5, we lay out the problem of SLAM navigation and control on the open hull. We apply sonar- and vision-based SLAM processes (Johannsson et al., 2010; Kim and Eustice, 2009), and combine them via incremental smoothing and mapping (iSAM) (Kaess et al., 2008; Kaess and Dellaert, 2009), to create a single comprehensive map. This enables drift-free control of the vehicle in real-time, relative to an environment it has never seen before. Section 6 describes the culmination of this work in experiments on the open areas of two ship hulls. In Section 7, we develop and demonstrate a methodology for creating watertight mesh models from sonar data taken at a safe distance from the hull. A watertight mesh is a prerequisite for any inspection to be performed at closer scale, because for this the vehicle usually has to move "up and into" the gear. The design of efficient and collision-free trajectories for detailed inspection, based on a mesh model and sampling-based planning, is explored in Section 8. Because this paper draws on a number of disciplines, additional background material is given in some sections.

Although our present work focuses on naval vessel inspection for mine-like objects, we believe that maturing of the HAUV commercial platform in combination with advanced navigation and planning algorithms could contribute more broadly to port and harbor security, and to ship husbandry tasks that include assessments of structural damage, debris, paint, corrosion, cathodic protection, and biofouling.

2 The Hovering Autonomous Underwater Vehicle

We summarize the essential aspects of the vehicle in its current generation "HULS" (Hull Unmanned Underwater Vehicle Localization Systems), as details on older versions can be found in prior publications (Vaganay et al., 2005, 2006; Hover et al., 2007); HULS itself is documented in (Vaganay et al., 2009). As shown in Fig. 1, the vehicle is flat with a square planform. The major components include flotation, a main electronics housing, a pressure-tolerant battery, thrusters, and navigation and payload sensors; these are specified in Table 1.

For basic dead-reckoned (DR) navigation, the vehicle relies on the depth sensor, attitude from the inertial measurement unit (IMU), and the DVL. Although the HG1700 IMU does have a magnetic compass, we do not use it in close proximity to steel structures. The DVL is oriented in one of two main configurations:

1. DVL normal to and locked onto the hull at a range

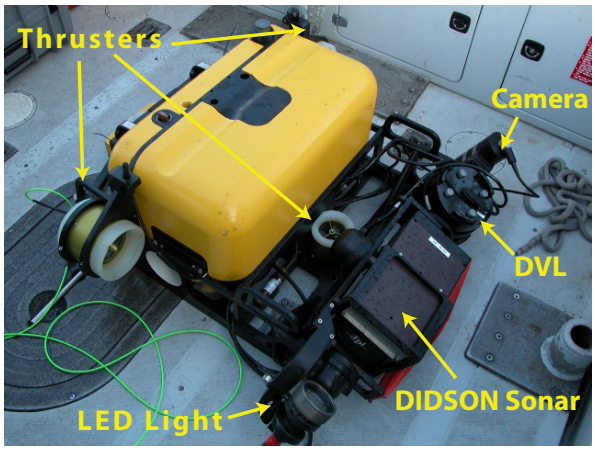


Fig. 1: The Bluefin-MIT Hovering Autonomous Underwater Vehicle (HAUV).

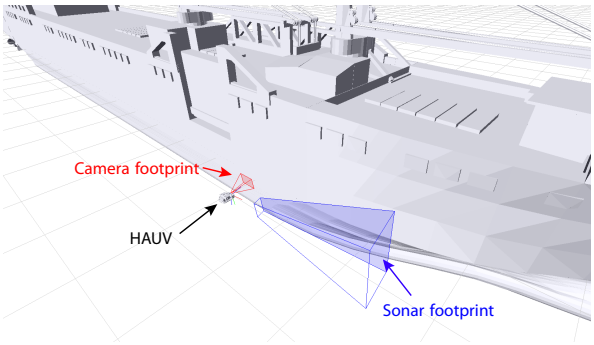


Fig. 2: Depiction of the sensor field of view for the imaging sonar and monocular camera during open-area, hull-locked inspection. Note that the two sensors concurrently image different portions of the hull. The footprint of the DVL's four beams is approximately the same as that shown for the camera.

Table 1: Specifications of major HAUV components.

Dimensions	1 m × 1 m × 0.45 m (L × W × H)
Dry Weight	79 kg
Battery	1.5 kWh lithium-ion
Thrusters	6, rotor-wound
IMU Sensor	Honeywell HG1700
Depth Sensor	Keller pressure
Imaging Sonar	Sound Metrics 1.8 MHz DIDSON
Doppler Velocity	RDI 1200 kHz Workhorse; also provides four range beams
Camera	1380 × 1024 pixel, 12-bit CCD
Lighting	520 nm (green) LED
Processor	650 MHz PC104
Optional Tether	150 m long, 5 mm dia. (fiber-optic)

of 1–2 m; horizontal and vertical strips following the hull are the most common trajectories.

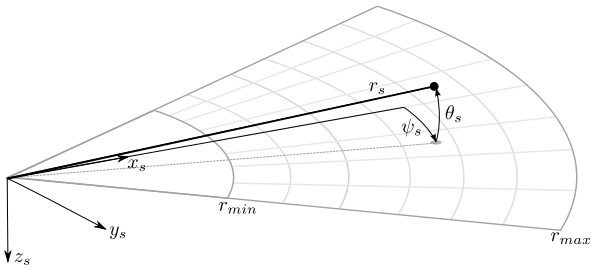
2. DVL pointing down and locked onto the seafloor; arbitrary trajectories in three-space are possible.

In HULS, for the first configuration the DIDSON imaging sonar (described more fully below and in a later section) and the DVL are mounted on a single servo-controlled pitching tray. The DIDSON is additionally mounted on a yaw servo. This allows for the DVL to stay normal to the hull, the condition of best performance. Assuming that the hull is locally smooth, then the DIDSON imaging volume intersects the hull symmetrically, and its grazing angle is controlled through the yaw servo; see Fig. 2. For bottom-lock navigation with HULS, we physically mount the DVL to the bottom of the pitch tray, and fix the tray at ninety degrees up. Then the yaw servo can point the DIDSON fan at any pitch angle from horizontal to ninety degrees up.

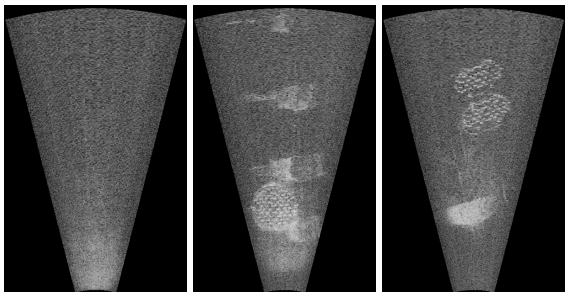
The DIDSON and the monocular camera system (Fig. 2) are the HAUV's two primary sensors for perception, and both are integrated into our real-time SLAM framework. The DIDSON has a 29-degree width, comprised of 96 separate beams (Belcher et al., 2001, 2002). We use it extensively in both its "imaging" (Fig. 3(b)) and "profiling" (Fig. 3(c)) modes, which are really descriptions of the vertical aperture: 28 degrees in the former, and about one degree in the latter. Functionally, the imaging mode is akin to side-scan sonar where protrusions from a flat surface, viewed at an appropriate grazing angle, are easily picked out by the human eye. Profiling mode provides a much narrower scan with no ambiguity, and thus can be used to create point clouds in three-space. We typically run the DIDSON at 5 fps.

The monocular camera system complements the DIDSON, and the HAUV supports two different configurations for it: an "underwater" mode (Fig. 4(a)) and a "periscope" mode (Fig. 4(b)). In underwater mode, the camera pitches with the DVL to keep an approximately nadir view to the hull—this results in continuous image coverage regardless of hull curvature. In periscope mode, the camera is mounted on top of the HAUV at a fixed angle of sixty degrees up, so that the camera protrudes above the water when the vehicle is near the surface. This provides above-water hull features that are useful for navigation, even when water turbidity conditions are very poor. In both configurations, we typically run the camera at 2–3 fps.

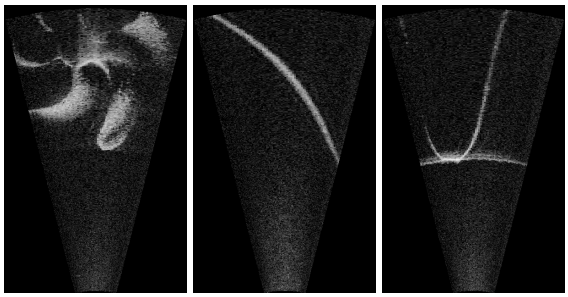
The vehicle's main processor integrates the DVL, IMU, and depth sensor, and provides low-level flight control. The payload sensors and our real-time mapping and control algorithms communicate with it through a backseat control interface. These functions can be carried out by a second computer onboard, or, as in our development, on a separate computer connected to the vehicle through a fiber optic tether.



(a) Sonar geometry.



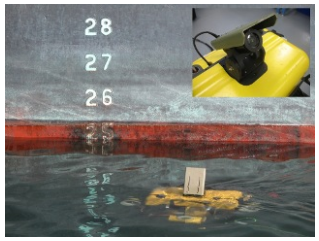
(b) Imaging mode (28° in θ_s, 1.9–6.4 m range).



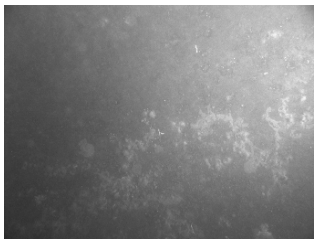
(c) Profiling mode (1° in θ_s, 1.9–10.9 m range).

Fig. 3: DIDSON sonar geometry and sample images for two different configurations. Imaging mode shows a clean hull (left) and several views of intakes and other structures. Profiling mode shows a propeller in cross-section (left), and two hull bottom profiles.

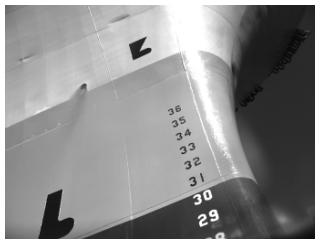
(a) Underwater mode.



(b) Periscope mode.



(c) Underwater imagery.



(d) Periscope imagery.

Fig. 4: Two different camera configurations for the HAUV: "underwater" mode and "periscope" mode. Sample imagery for each configuration is shown.

3 Integrated SLAM Navigation and Control

One of the main challenges of fielding a free-swimming hull inspection vehicle is navigation over a period of hours. Typical solutions rely upon either odometry-derived dead-reckoning (wheel encoders or DVL, augmented with IMU), some form of acoustic ranging for absolute positioning (such as long-baseline (LBL) or ultra-short-baseline (USBL)), or a combination thereof. Working examples of representative systems are documented in the literature; for example, see (Trimble and Belcher, 2002) and (Vaganay et al., 2005). The main difficulties of traditional navigation approaches are that they either suffer from unbounded drift (i.e., DR), or they require external infrastructure that needs to be set up and calibrated (e.g., LBL and USBL). Both of these scenarios tend to vitiate the "turn-key" automation capability that is desirable in hull inspection.

Nontraditional approaches to hull-relative navigation seek to alleviate these issues. Negahdaripour and Firoozfam (2006) developed underwater stereo-vision to navigate an ROV near a hull; they used mosaic-based registration methods and showed early results for pool and dock trials. Walter et al. (2008) used an imaging sonar for feature-based SLAM navigation on a barge, showing offline results using manually-established feature correspondence. Most recently, Ridao et al. (2010) have reported on an AUV for autonomous dam inspection; their navigation solution uses USBL and DVL data *in-situ* during the mapping phase, followed by an offline image bundle adjustment phase to produce a globally-optimal photomosaic and vehicle trajectory.

Our path has been to take an integrated approach toward the real-time navigation and control problem. Specifically, we have developed a system that allows for the HAUV to build a map *in situ* of an *a priori* unknown hull, and to simultaneously use this map for navigation correction and waypoint control of the vehicle. Our system uses hull-relative DVL odometry, DIDSON imaging sonar and monocular camera constraints, and a modern pose-graph SLAM optimization framework to produce an accurate and self-consistent 3D (i.e., six degree of freedom) trajectory estimate of the vehicle.

Our SLAM navigation and control architecture (Fig. 5) is organized into distinct modules: a perceptual "front-end" (`camera-client` and `sonar-client`), a single shared "back-end" for inference (`seserver`), and a high-level controller for waypoint navigation (`HAUV-client`). Here, we document the most salient aspects of our integrated real-time system and approach.

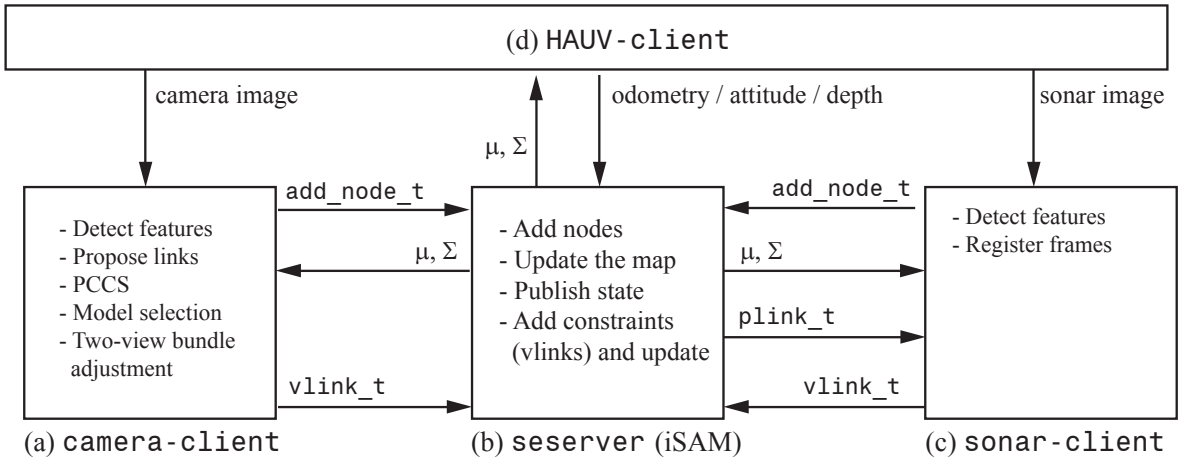


Fig. 5: Publish/subscribe shared estimation architecture using iSAM. The shared estimation server, `seserver`, listens for add node message requests, `add_node_t`, from each sensor client. For each node, sensor client processes, `camera-client` and `sonar-client`, are responsible for generating message requests for proposed, `plink_t`, and verified, `vlink_t`, edge links to be added to the graph by `seserver` as constraints. Finally, the `HAUV-client` process is responsible for high-level vehicle waypoint control using the `seserver` state estimate.

3.1 State Representation

An estimate of the vehicle pose, consisting of position and attitude, is needed for navigating along the ship hull during inspection. The vehicle position in 3D is specified by Cartesian coordinates x, y, z, with the x-y plane horizontal and z downward. The attitude of the vehicle is specified by the standard Euler angles ϕ, θ, ψ that reference roll, pitch, and heading, respectively. We call the position vector of the vehicle $\mathbf{x} = [x, y, z, \phi, \theta, \psi]^\top$. The origin of the coordinate system can be chosen arbitrarily; here we use the pose at which DVL lock on the hull is first achieved.

3.2 Efficient State Estimation

We adopt a pose-graph formulation of the SLAM problem to obtain a least-squares estimate of the vehicle trajectory, based on all available measurements. The pose-graph formulation keeps the complete trajectory, allowing loop-closure constraints to be added based on geometric measurements obtained from sonar and camera data, and correcting navigation drift that would accumulate over time. For efficient online solving, we use an open source implementation of the iSAM algorithm (Kaess et al., 2010).

Following the formulation in Kaess et al. (2012), we use a factor graph to represent the estimation problem (Fig. 6). A factor graph is a bipartite graph $G = (\mathcal{F}, \mathcal{Q}, \mathcal{E})$ with two node types: factor nodes $f_i \in \mathcal{F}$ (each representing a constraint) and variable nodes $q_j \in \mathcal{Q}$ (each representing a quantity to be estimated). Edges $e_{ij} \in \mathcal{E}$ encode the sparse structure, where a single edge e_{ij} exists if and only if the factor node f_i depends on variable node q_j. A factor graph G defines the factorization of a function $f(\mathcal{Q})$ as

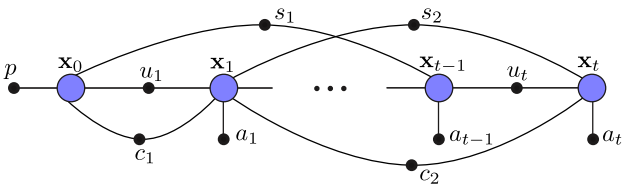


Fig. 6: Factor graph formulation of the integrated navigation problem, where variable nodes are shown as large circles, and factor nodes (measurements) as small solid circles. The factors shown are relative-pose measurements u and absolute pose measurements a, a prior p, sonar constraints s and camera constraints c.

$$f(\mathcal{Q}) = \prod_i f_i(\mathcal{Q}_i), \tag{1}$$

where $\mathcal{Q}_i$ is the set of variables that factor i connects to. Our goal is to find the variable assignment $\mathcal{Q}^*$ that maximizes (1)

$$\mathcal{Q}^* = \arg\max_{\mathcal{Q}} f(\mathcal{Q}). \tag{2}$$

When assuming Gaussian measurement models

$$f_i(\mathcal{Q}_i) \propto \exp\left(-\frac{1}{2}\left\|h_i(\mathcal{Q}_i) - \zeta_i\right\|^2_{\Sigma_i}\right), \tag{3}$$

as is standard in the SLAM literature, the factored objective function we want to minimize (2) corresponds to the nonlinear least-squares criterion

$$\arg\min_{\mathcal{Q}} \left(-\log f(\mathcal{Q})\right) = \arg\min_{\mathcal{Q}} \frac{1}{2}\sum_i \left\|h_i(\mathcal{Q}_i) - \zeta_i\right\|^2_{\Sigma_i}. \tag{4}$$

Here $h_i(\mathcal{Q}_i)$ is a measurement function and ζ_i a measurement, and the notation

$$\|\mathbf{e}\|_{\Sigma}^2 := \mathbf{e}^\top \Sigma^{-1} \mathbf{e} = \left\| \Sigma^{-\frac{1}{2}} \mathbf{e} \right\|^2 \tag{5}$$

defines the vector $\mathbf{e}$'s squared Mahalanobis distance with covariance matrix Σ.

In practice one always considers a linearized version of problem (4) in the context of a nonlinear optimization method. We use Gauss-Newton iterations to solve a succession of linear approximations to (4) in order to approach the minimum. At each iteration of the nonlinear solver, we linearize around a point $\mathcal{Q}$ to get a new, linear least-squares problem in $\mathbf{\Delta}$

$$\arg\min_{\mathbf{\Delta}} (-\log f(\mathbf{\Delta})) = \arg\min_{\mathbf{\Delta}} \|A\mathbf{\Delta} - \mathbf{b}\|^2, \tag{6}$$

where $A \in \mathbb{R}^{m \times n}$ is the measurement Jacobian consisting of m measurement rows, $\mathbf{\Delta}$ is an n-dimensional tangent vector, and $\mathbf{b}$ is a combination of measurement and residual. Note that the covariances Σ_i have been absorbed into the corresponding block rows of A, making use of (5).

The minimum of the linear system $A\mathbf{\Delta} - \mathbf{b}$ is obtained by QR factorization, yielding $R\mathbf{\Delta} = \mathbf{b}'$, which is then solved by back-substitution. To avoid repeated solving of the same system, iSAM uses Givens rotations (Kaess et al., 2008) to update the existing matrix factorization when new measurements arrive. Periodic batch factorization steps allow for re-linearization as well as variable reordering, which is essential to retaining the sparsity and therefore the efficiency of the algorithm.

Once $\mathbf{\Delta}$ is found, the new estimate is given by $\mathcal{Q} \oplus \mathbf{\Delta}$, which is then used as the linearization point in the next iteration of the nonlinear optimization. The operator $\oplus$ represents simple addition for position, and an exponential map update for 3D rotations (Kaess et al., 2012), internally represented by over-parametrized quaternions.

3.3 Shared State Estimation

We have developed a shared estimation engine, called `seserver` (Fig. 5(b)), based on the iSAM state estimation algorithm above. Sonar and camera clients each have their own criteria to add a node to the graph (e.g., feature richness or distance to the existing nodes). Non-temporal (i.e., loop-closure) constraints added from the sonar and camera are essential in bounding the SLAM navigation uncertainty.

The client has two choices for creating registrations: the server proposes possible registration candidates (as for `sonar-client` depicted in Fig. 5(c)), or the client can request state information from the server to guide the search for registration candidates (as for `camera-client` depicted in Fig. 5(a)). In the latter case, the server publishes mean and covariance (μ, Σ) back to the client. The corresponding entries of the covariance matrix are recovered by an efficient algorithm (Kaess and Dellaert, 2009).

Nodes of the factor graph (Fig. 6) correspond to sensor "keyframes," and represent historical poses from the vehicle trajectory. The nodes are constrained via four different types of sensor constraints, and a special prior. The prior p removes the gauge freedom of the system by fixing the first pose, arbitrarily chosen to be at the origin. The IMU integrates DVL measurements to give relative constraints u in position and attitude. Absolute measurements of pitch and roll are available from the projection of gravity onto the IMU's accelerometers; note that a magnetic compass is not viable operating in close proximity to a ferrous ship hull. With an absolute depth measurement from the pressure sensor, we thus obtain 3-degree of freedom (DOF) absolute constraints a. Sonar constraints s provide 3-DOF information, and camera constraints c provide 5-DOF information; these are described in more detail in Sections 4 and 5.

3.4 Integrated Navigation

We have implemented integrated waypoint control (depicted functionally by the `HAUV-client` module in Fig. 5(d)) to allow revisiting places of interest along the hull. The vehicle can be sent back to inspect the object from a different perspective or in more detail, for example by going closer or moving slower than during the original pass.

The key challenge to waypoint navigation is that the vehicle interface uses hull-relative coordinates for navigation (i.e., the DVL is locked to the hull), while the SLAM estimate is in Cartesian coordinates. Hull-relative coordinates are derived from the DVL measurement and specify the vertical and horizontal distance along the hull, relative to an arbitrary starting point (Vaganay et al., 2005). They are inconsistent as a global coordinate system, however, because the ship hull surface is a two-dimensional manifold: the same point on the hull generally has different hull-relative coordinates if reached by different trajectories.

Our solution is to control the vehicle with estimated Cartesian information, projected into hull-relative coordinates by assuming a locally flat hull. Navigation to a waypoint requires retrieving the current Cartesian estimate $\hat{\boldsymbol{\ell}}_t$ for the given waypoint at timestamp t. With the current vehicle location given by $\boldsymbol{\ell} = [x, y, z]^\top$, the Cartesian vector of desired motion is given by $\delta\boldsymbol{\ell} = \hat{\boldsymbol{\ell}}_t - \boldsymbol{\ell} = [\delta_x, \delta_y, \delta_z]^\top$. This move is projected into hull-relative horizontal and vertical coordinates (δ_h, δ_v) using the vehicle's current heading ψ. For the horizontal move, we have

$$\delta_h = \delta_x \sin\psi + \delta_y \cos\psi. \tag{7}$$

For the vertical component, we project the target vector onto the plane given by the vehicle's forward and downward axes according to its current Cartesian estimate

$$\delta_v = \sigma\sqrt{\delta_z^2 + (\delta_x \cos\psi + \delta_y \sin\psi)^2}, \tag{8}$$

where $\sigma \in \{-1, +1\}$ has to be chosen suitably, depending on whether the vehicle is on the side of the hull (vertical surface) or the bottom (horizontal surface), and the direction of the move. As we will show, these local mappings are satisfactory for controlling and moving the HAUV accurately over the hull.

4 Sonar Registration and Constraints

On the open parts of the hull, registering sonar frames while assuming a locally flat ship hull provides 3-DOF relative-pose constraints. On the event of a sonar link proposal (i.e., a `plink_t` in Fig. 5) from `seserver`, the `sonar-client` attempts to register a pair of sonar images and, if successful, estimates and publishes the verified link (i.e., a `vlink_t`). In this section we describe the sonar imaging geometry, and our feature extraction and registration procedure (Johannsson et al., 2010).

4.1 Imaging Sonar Geometry

Following the formulation in Negahdaripour et al. (2009), we define the geometry of the imaging sonar and derive a model that describes how the image is formed. To generate an image, the sonar emits from an angular array a set of narrow-beam sound waves, and then listens to the returns, sampling the acoustic energy returned from the different directions. The instrument provides time of flight and intensity for each azimuth angle, and combining the returns from all the elements provides an image of the scattering surfaces in front of the sonar. Imaging sonar samples are shown in Fig. 3(b), with 96 beams over a 29-degree horizontal field of view, and a vertical beam width of 28 degrees using a spreader lens. Note that for a given point in the image the actual object can lie anywhere on an arc at a fixed range, spanning the vertical beam width.

Mathematically the imaging process can be described as follows. We define the coordinate system for the sonar as shown in Fig. 3(a). Let us consider a point $\mathbf{p} = [x_s\ y_s\ z_s]^\top$ in the sensor coordinate frame, and let $\mathbf{s} = [r_s\ \theta_s\ \psi_s]^\top$ be the same point in spherical coordinates, as indicated in the figure. We can relate the spherical and Cartesian coordinates with

$$\mathbf{p} = \begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix} = \begin{bmatrix} r_s \cos\theta_s \cos\psi_s \\ r_s \cos\theta_s \sin\psi_s \\ -r_s \sin\theta_s \end{bmatrix}, \tag{9}$$

$$\mathbf{s} = \begin{bmatrix} r_s \\ \theta_s \\ \psi_s \end{bmatrix} = \begin{bmatrix} \sqrt{x_s^2 + y_s^2 + z_s^2} \\ \arctan 2\left(-z_s, \sqrt{x_s^2 + y_s^2}\right) \\ \arctan 2(y_s, x_s) \end{bmatrix}. \tag{10}$$

The sensor, however, does not provide θ_s, so we measure point $\mathbf{p}$ as $I(\mathbf{p}) = [r_s\ \psi_s]^\top$, and the Cartesian projection from this data is

$$\hat{I}(\mathbf{p}) = \begin{bmatrix} r_s \cos\psi_s \\ r_s \sin\psi_s \end{bmatrix}. \tag{11}$$

For a small vertical beam width, this can be viewed as an approximation to an orthographic projection.

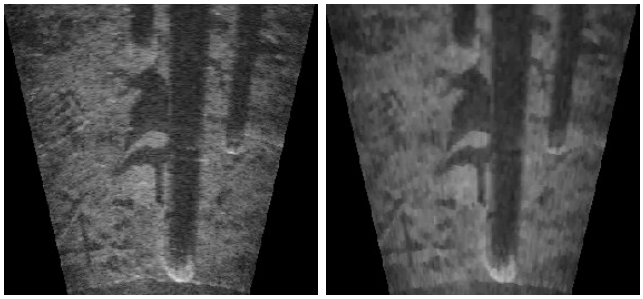
4.2 Feature Extraction

The imaging sonar returns intensity values at a number of ranges along each azimuth beam. An example sonar image with some features on a flat surface is shown in Fig. 7(a). Protrusions are typified by a strong return followed by a shadow, while a hole or depression on the surface shows as a shadow, often without an associated bright return. An object viewed from two substantially different angles may not be identifiable as the same object; this is a well-known attribute of sonar imaging. The return is also affected by material properties, strength of the transmission, receiver sensitivity, distance to the target and grazing angle, among other factors.

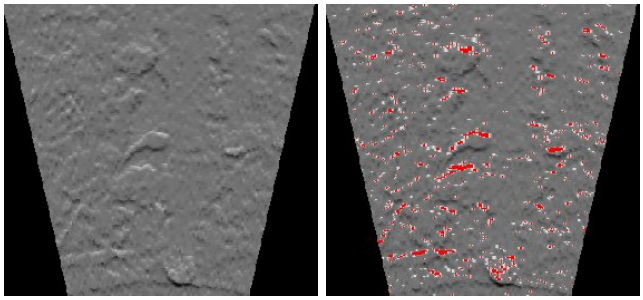
Stable features are most identifiable from sharp transitions in image intensity. Based on this fact, the main steps of the algorithm are:

1. Smooth the image.
2. Calculate its gradient.
3. Threshold a top fraction as features.
4. Cluster points and eliminate the small clusters.

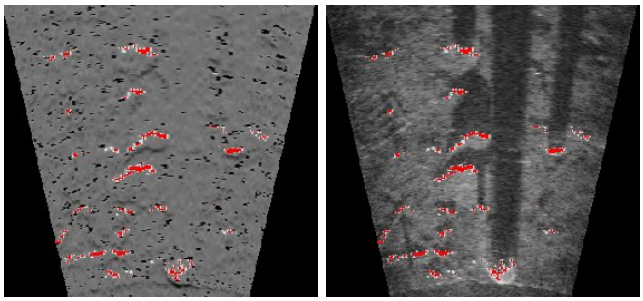
We smooth using a median filter, which significantly reduces noise while still preserving edges, as shown in Fig. 7(b). Next, the gradient at each range (pixel) is calculated as the difference between the local value and the mean of the n_p previous values, taken outward along the beam (Fig. 7(c)). The number of previous values n_p affects the type and size of objects that will be detected. Then points with negative gradients exceeding a given threshold are selected as candidates (Fig. 7(d)). The threshold is chosen adaptively, such that a fixed fraction of the features are retained. Note that strong positive gradients are ignored because these correspond to the ends of shadows that, unlike the negative gradients,



(a) Initial sonar image. (b) Smoothed image.

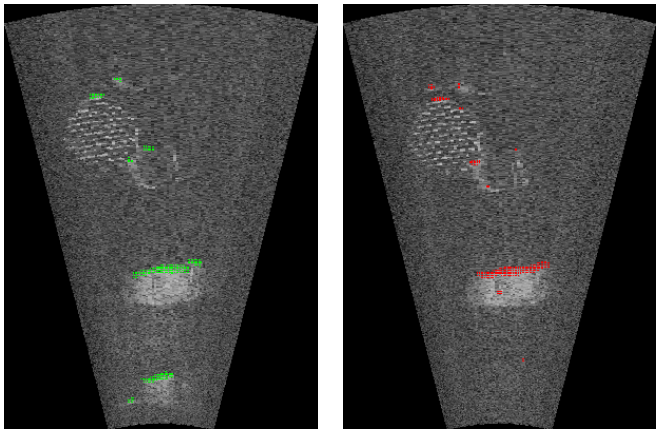


(c) Gradient of image. (d) Thresholded gradient.

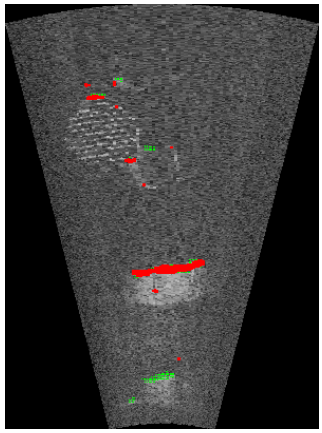


(e) Clustering of detections. (f) Extracted features.

Fig. 7: Intermediate steps of the sonar feature extraction process. The extracted features are shown in red.



(a) Frame A. (b) Frame B.



(c) Registration of A and B.

Fig. 8: Two scans before and after registration. Red and green points show features from the model and current scan, respectively.

depend on the sensor distance to the hull. Next, spurious features are eliminated by clustering (Fig. 7(e)). The remaining extracted features as shown in Fig. 7(f) typically contain on the order of one thousand points.

The Cartesian error associated with a successful registration is governed primarily by the tall vertical aperture of the sensor in imaging mode. With a typical inclination angle of twelve degrees between the sensor's x_s-y_s plane and the flat surface, and a perpendicular distance of one to two meters, we observe errors up to 10–15 cm, but typically they are below 5 cm.

4.3 Registration

We align two overlapping sonar images by registration as shown in Fig. 8, using the normal distribution transform (NDT) algorithm (Biber and Strasser, 2003). The NDT assigns a scan's feature points to cells of a regular grid spanning the covered area. For each cell we calculate the mean and variance of its assigned points. This is done for four overlapping grids, where each grid is shifted by half a cell width along each axis; using multiple shifted grids alleviates the effect of discontinuities resulting from the discretization of space. Two of the NDT's benefits are that it provides a compact representation of the scan, and that no exact correspondences between points are needed for alignment. This second fact is useful for us because movement of the HAUV causes variations in the returns from surfaces, leading some points to drop in and out of the extracted feature set.

The NDT of a scan serves as our model for registration. Given a new scan, a score is calculated for each point by evaluating the Gaussian of the NDT cell that receives the point. This provides a measure of the likelihood that a given point is observed based on our model. We define a cost function as the sum of the negative scores of all the points in the current view, and minimizing this cost with respect to the change in (x_s, y_s)

position and heading of the sonar provides the transformation between scans. Because the main goal of the registration method is to close loops after drift has accumulated, we do not use the `seserver` estimate to initialize the search. Instead, we repeat the registration starting from several fixed initial values in an attempt to find the global minimum. To avoid incorrect matches, acceptance is based on a conservative threshold of a normalized score, and also requires a minimum number of points to be matched. A successful registration gets published to `seserver` as a constraint after adding the relative transformation between sonar and vehicle frames.

Our method is inherently sensitive to viewpoint because it extracts a dense representation of the sonar features, in contrast to point landmarks. This dense representation is desirable because it is far more discriminative in large image sets. To reduce sensitivity, multiple overlapping frames are stored in the map so that when the vehicle revisits we can expect at least one of the frames to be close to the current view. Typically, the registrations are within 0.5 m translation and 10° rotation; our approach has worked well in practice.

5 Camera Registration and Constraints

Independent of sonar constraints, the camera provides 5-DOF bearing-only constraints between `seserver` camera nodes that have sufficient image overlap for registration. We assume that the camera is calibrated, such that pairwise image registration yields relative-pose modulo scale observations, as depicted in Fig. 9. Following the formulation in Eustice et al. (2008), these constraints are modeled as an observation of the relative azimuth α_{ji} and elevation angle β_{ji} of the baseline direction of motion, and the relative Euler orientation $[\phi_{ji}\ \theta_{ji}\ \psi_{ji}]^\top$ between two camera nodes i and j in the graph:

$$\zeta_{ji} = h_{ji}(\mathbf{x}_j, \mathbf{x}_i) = [\alpha_{ji}\ \beta_{ji}\ \phi_{ji}\ \theta_{ji}\ \psi_{ji}]^\top. \qquad (12)$$

Image keyframes are added to `seserver`'s pose graph using a minimum-distance-traveled metric, which in our case is approximately 50% image overlap. We employ a standard pairwise feature-based image registration framework (Fig. 10) to register candidate image pairs, though adapted to the peculiarities of underwater imaging, as outlined next.

5.1 Link Hypothesis

Link hypothesis deals with the proposal of candidate image pairs for which to attempt registration with the current view. An exhaustive attempt over all image pairs is not practical, and therefore we first prune unrealistic candidate node pairs by considering their projected image overlap, using the most recent estimate

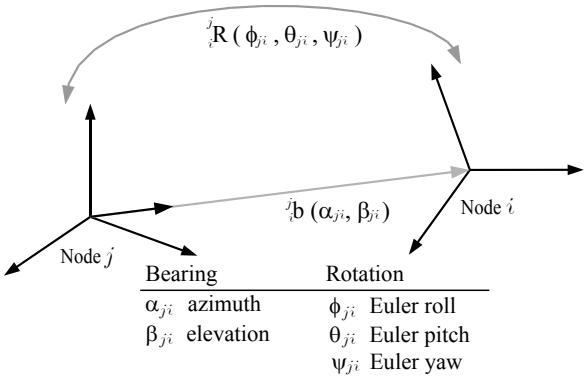


Fig. 9: The pairwise 5-DOF camera measurement (i.e., relative-pose modulo scale).

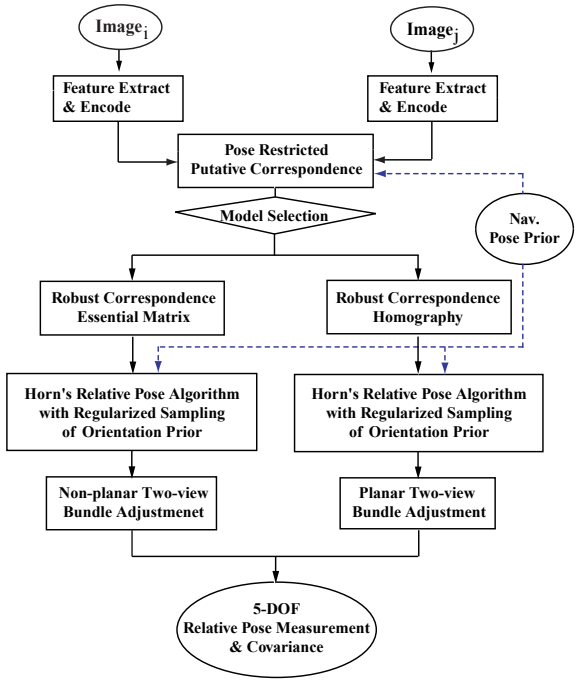
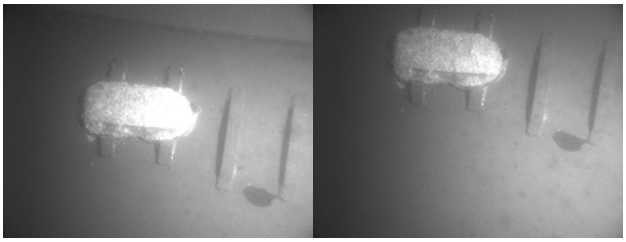


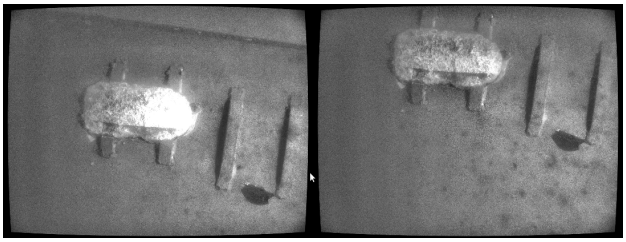
Fig. 10: Block-diagram for the camera registration engine.

from `seserver`. Since the HAUV approximately maintains a fixed standoff distance to the hull (using DVL ranges), our link hypothesis strategy uses a greatly simplified radial model for image overlap (i.e., analogous to a circular field-of-view assumption) to project image footprints onto the hull. This model also enforces a view angle constraint based upon the camera's principal axis.

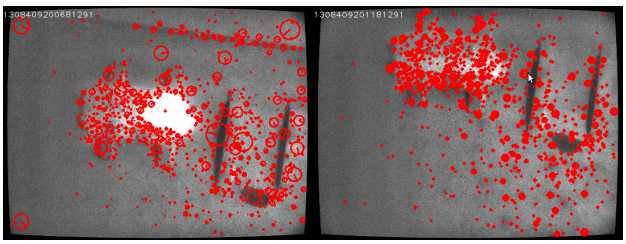
Under this scheme, we set a minimum and maximum percent image overlap to obtain plausible bounds on inter-camera-node distance. We compute a first-order probability associated with whether or not a candidate node pair satisfies these constraints (Eustice et al., 2008); the k most likely overlapping candidate image nodes ($k = 5$ in our application) are then attempted for registration with the current view.



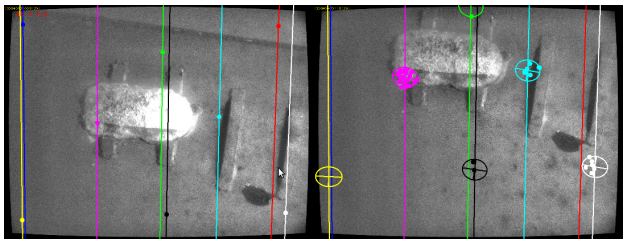
(a) Raw imagery for two keyframes i and j.



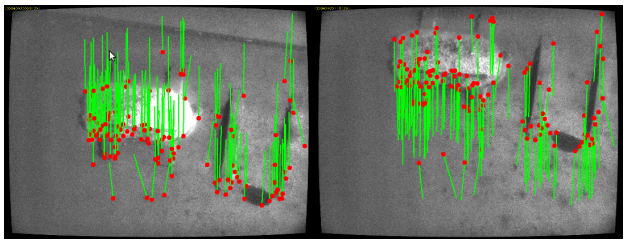
(b) Radially undistorted and histogram equalized imagery.



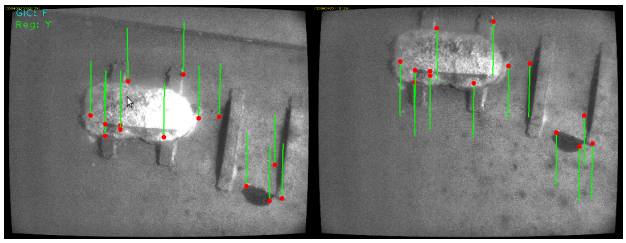
(c) Extracted SIFT features.



(d) Pose constrained correspondence search (PCCS).



(e) Putative correspondences resulting from PCCS.



(f) Resulting inliers from geometric model selection framework.

Fig. 11: Intermediate steps of the real-time visual SLAM processing pipeline depicted for sample pairwise imagery. Raw images (a) are radially undistorted and contrast-limited adaptive histogram equalized to enhance visibility (b); these are then used to extract SIFT features (c) (§5.2). To find the correspondences between images, we exploit our navigation prior and project pose uncertainty into image space to obtain first-order ellipsoidal search bounds for feature matching (d) (§5.3.1). Putative correspondences (e) are found by searching within these ellipses, which are then refined to a set of inliers (f) via an automatic geometric model selection framework (§5.3.2), which in this example selects an essential matrix registration model. In (e) and (f), red dots represent image feature locations where the other end of the green line indicates the corresponding pixel point in the paired image.

5.2 Feature Extraction

Raw images (Fig. 11(a)) are first radially undistorted and contrast-limited adaptive histogram equalized (Zuiderveld, 1994), to enhance visibility (Fig. 11(b)). For feature extraction, we use the scale invariant feature transform (SIFT) (Lowe, 2004) to extract and encode 128-vector features from the imagery (Fig. 11(c)). To enable real-time performance at 2–3 fps, we use a graphics processing unit (GPU)-based SIFT implementation (Wu, 2007).

5.3 Registration

5.3.1 Pose Constrained Correspondence Search

For each candidate image pair, we use a pose-constrained correspondence search (PCCS) (Eustice et al., 2008) to guide the putative matching (Fig. 11(d)–(e)). PCCS allows us to spatially restrict the search region in an image when establishing putative correspondences, thereby reducing the required visual uniqueness of a feature. In other words, PCCS allows us to confidently identify correspondences that would not be possible using global appearance-based information only—since visual feature uniqueness no longer needs to be globally identifiable over the whole image. Rather, it only needs to be locally identifiable within the geometrically-constrained region. These ellipsoidal regions are aligned with the epipolar geometry, and in cases where we have a strong prior on the relative vehicle motion (for example, sequential imagery with odometry or when the `seserver` estimate is well-constrained), then this probabilistic search region provides a tight bound for putative matching; see Fig. 11(d).

5.3.2 Geometric Model Selection

The resulting putative correspondences are refined through a robust-estimation model-selection frame-

work (Kim and Eustice, 2009). Because we operate the camera for navigation drift correction in open areas of the hull, one would expect locally planar structure to appear with a fair amount of regularity, and assume that a homography registration model is adequate. However, this assumption is not everywhere true, as some portions of the hull are highly three-dimensional (e.g., bilge keel, shaft, screw). To accommodate this structure variability, we employ a geometric model-selection framework to automatically choose the proper registration model, either homography or essential matrix, when robustly determining the inlier set (Fig. 11(f)) to pass as input to the two-view bundle adjustment step that follows.

5.3.3 *Two-view Bundle Adjustment*

Once a proper model has been selected, we run a two-view bundle adjustment to estimate the optimal relative-pose constraint between the two camera nodes (Kim and Eustice, 2009). When the model selection criteria results in choosing the essential matrix, the objective function is chosen such that it minimizes the sum squared reprojection error in both images by optimizing a 5-DOF camera-relative pose and the triangulated 3D structure points associated with the inlier image correspondences. Alternatively, in cases where geometric model selection chooses the homography registration model, the objective function is chosen to minimize the sum squared reprojection error using a plane-induced homography model. The optimization is performed over the homography parameters and the optimal pixel correspondences that satisfy the homography mapping exactly. In either case, the two-view bundle adjustment results in a 5-DOF bearing-only camera measurement (Fig. 9), and a first-order estimate of its covariance, which is then published to `seserver` as a constraint.

6 Experimental Results for SLAM Navigation and Control

In this section, we highlight survey results for two different vessels (Fig. 12) on which we have demonstrated our real-time system. The first is the *SS Curtiss*, a 183 m-long single-screw roll-on/roll-off container ship currently stationed at the U.S. Naval Station in San Diego, California; experiments with the *SS Curtiss* were conducted in February 2011. The second vessel for which we show results is the U.S. Coast Guard Cutter *Seneca*, an 82 m medium-endurance cutter stationed in Boston, Massachusetts; experiments with the *USCGC Seneca* were conducted in April 2011.

(a) *SS Curtiss*.

(b) *USCGC Seneca*.

	SS Curtiss	***USCGC Seneca***
Length	183 m	82 m
Beam	27 m	12 m
Draft	9.1 m	4.4 m
Displacement	24,182 t	1,800 t
Propulsion	Single-Screw	Twin-Screw

(c) Vessel characteristics.

Fig. 12: Two representative vessels on which the HAUV has been deployed.

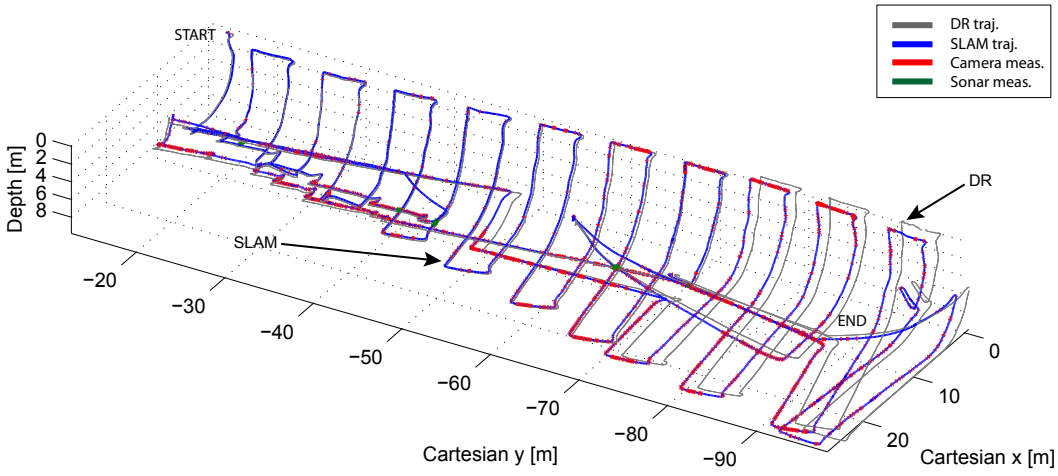
6.1 Real-time SLAM Navigation

Open-area surveys using concurrent sonar and camera for integrated SLAM navigation have been demonstrated on both the *SS Curtiss* and the *USCGC Seneca*. For the *SS Curtiss* inspection described in Fig. 13, tracklines consist of vertical lines up and down the hull from the waterline to the keel. These tracklines are spaced approximately four meters apart to provide redundant coverage with the sonar; this is the primary means of search and detection in the mission. Conversely, at a hull standoff distance of one meter, the camera footprint is only a meter or so in width, and hence its field of view does not overlap regularly with imagery from neighboring tracklines. This is acceptable because we are using the camera here as a navigation aid, not trying to achieve 100% coverage with it. In this particular survey we ran the camera in periscope mode, allowing for constraints to be derived both above- and below-water.

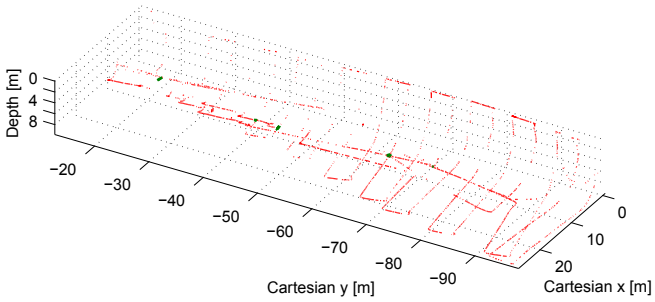
Both the camera and sonar imagery provide sequential constraints and loop closures when crossing areas that were previously visited, as in the horizontal slices seen in Fig. 13(a). The set of all camera- and sonar-derived SLAM constraints is depicted in Fig. 13(b) and (c).

6.2 Waypoint Navigation and Control

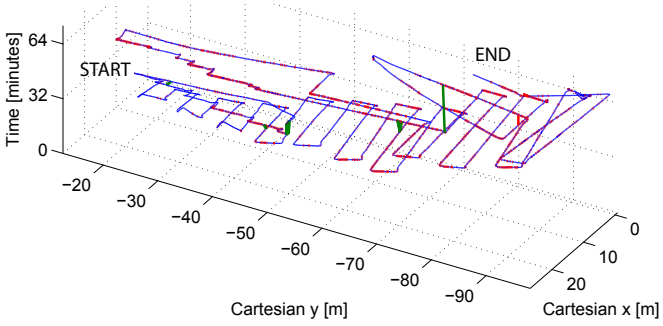
Similar real-time SLAM results are shown for an inspection mission on the *USCGC Seneca* in Fig. 14. The trackline and standoff spacing were approximately the same as that used on the *SS Curtiss* mission above, but the camera was configured in underwater mode, i.e., actuated with the DVL to look nadir to the hull. The drift between the gray trajectory (DR) and blue trajec-



(a) *SS Curtiss* SLAM result.



(b) Sonar and camera constraints.



(c) Time elevation graph of loop-closure constraints.

Fig. 13: SLAM navigation results on the *SS Curtiss*. (a) The SLAM-derived trajectory (blue) versus DVL dead-reckoned (gray). The survey started at the upper left corner of the plot and continued toward the right. Periodically, the vehicle traversed orthogonal to the nominal trackline pattern to revisit areas of the hull and obtain loop-closure constraints. (b) The SLAM estimate combines both camera (red) and sonar (green) constraints in the pose-graph. (c) A time-elevation depiction of the loop-closure constraints, where the vertical axis is time. This view clearly shows where large loop-closure events occur in both time and space.

tory (SLAM) captures the navigation error corrected by SLAM. Successful camera and sonar measurements are indicated in the time elevation plot of Fig. 14(b).

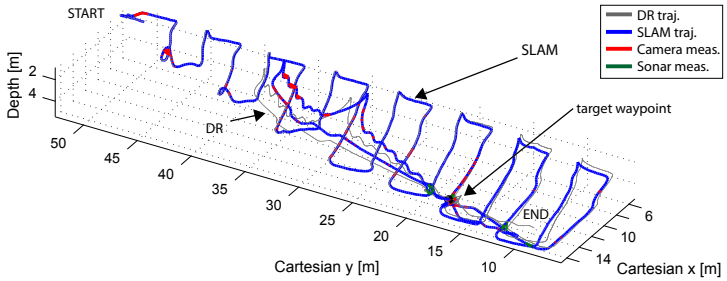
Demonstrating the accuracy and utility of our SLAM-derived state estimate during inspection of the *USCGC Seneca*, Fig. 14 also shows waypoint functionality. Waypoints in this experiment demarcated "human-recognizable" features on the hull in either the camera or sonar modality (e.g., port openings, inert mine targets), observed on the first pass; later in the mission, we commanded the vehicle to revisit them. Fig. 14(a) denotes one of these waypoints (i.e., the point labeled "target waypoint"), which we twice commanded the vehicle to return to. Sample camera (Fig. 14(c)) and sonar (Fig. 14(d)) images are shown for the vehicle when it revisits the site. If the navigation is correct, then the "live" sonar and camera views should match the corresponding keyframes stored in the `seserver` pose-graph, which they do.

A video of the real-time SLAM waypoint navigation performance is presented as Extension 1 in Appendix A.

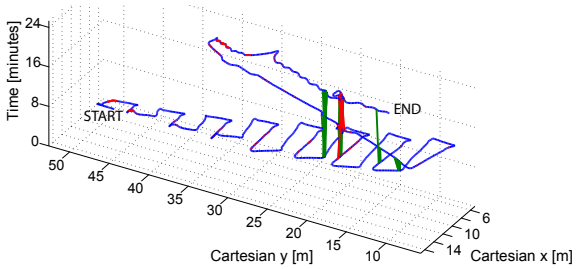
7 Modeling and Inspection of Complex Regions

We now move into the second major element of our algorithmic work in this paper. As described in the Introduction, when considering the ship as a whole we have to address the inspection of complex 3D structures, particularly at the stern where shafts, propellers, and rudders protrude from the hull. Long-term navigation drift is less of an issue here than on the open areas of the hull, because it is a smaller area. On the other hand, to assure full 100% coverage in this complex space requires careful path planning, subject to obstacles and to limited sensor views; this challenge is the subject of the present and following sections.

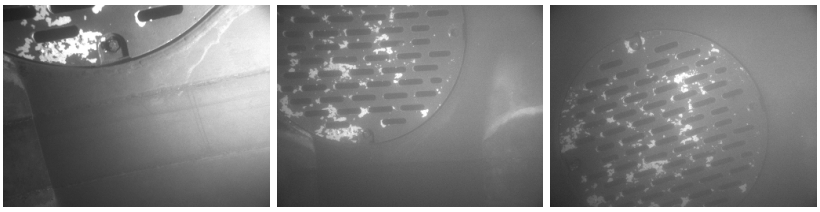
For all of our work involving large 3D structures, we deploy the DIDSON sonar in profiling mode, i.e., with



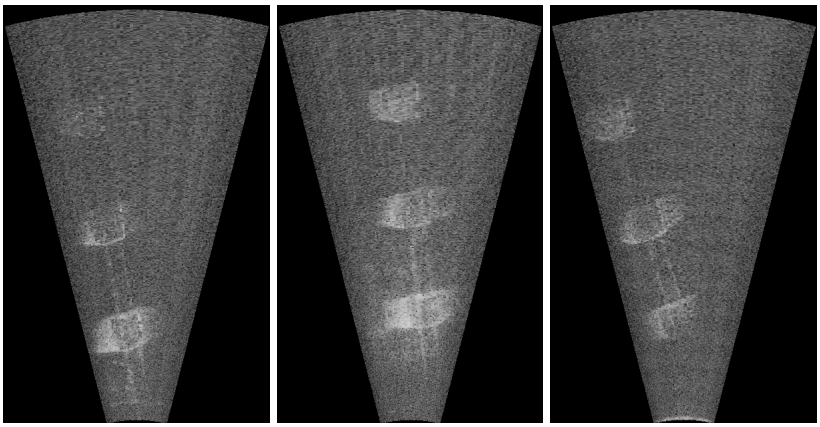
(a) *USCGC Seneca* SLAM result.



(b) Time elevation graph of loop-closure constraints.



(c) Camera waypoint imagery.



(d) Sonar waypoint imagery.

Fig. 14: Depiction of the HAUV waypoint navigation feature as used on a hull survey of the *USCGC Seneca* in April 2011. The operator manually selects points of interest (i.e., waypoints) along the hull via a visual dashboard display; the vehicle can re-visit these waypoints (and other points) using the SLAM state estimate for real-time control. (a) The real-time SLAM trajectory estimate (blue) versus the DVL dead-reckoned trajectory estimate (gray); SLAM camera constraints (red) and sonar constraints (green) are shown. (b) The SLAM pose-graph where time is the vertical axis; this rendering clearly shows the loop-closure constraints derived from camera and sonar. (c) and (d) show sample imagery from the target waypoint identified in (a) to highlight the accuracy of the real-time waypoint navigation feature. In both modalities, the leftmost figure is the keyframe associated with the node in the `seserver` pose-graph, while the middle and right images are "live" views obtained when the HAUV is commanded to re-visit the target waypoint a first and second time. The real-time SLAM estimate is self-consistent and accurate enough to control the vehicle back to within a few centimeters of any position along the hull. For reference, the camera field-of-view in (c) is approximately one meter.

a one-degree vertical aperture (θ_s). Also, the DVL is pointed at the seafloor rather than at the hull itself. At present we require a clean line of sight between the DVL and the seafloor, although the future development of profiling-mode SLAM could permit lapses in the DVL lock.

7.1 Complex Region Inspection Strategy

The ability to discover and map a vessel's stern arrangements without the aid of a computer-aided-design (CAD) or other model has been valuable in our work so far with naval vessels. Many vessels are older and poorly documented, or have been modified to an extent that the available description is simply incorrect. The lack of accurate *a priori* models exists for commercial vessels as well. Thus, our methodology is intended to proceed from having no knowledge of the structure, to a survey made at large range and poor resolution, to another survey made at short range and high resolution. The coarse survey enables the fine survey trajectory to go "up and into" the running gear.

We execute the initial low-resolution *identification survey* at a safe distance from the stern, typically seven to ten meters; this is intended to identify the major ship structures, and enable the construction of a 3D model. Due to the challenges of filtering profiling-mode DIDSON data—including the removal of noise and second returns—we use manual processing to construct the mesh. Automating this task is an open problem.

Using the coarse 3D model, a path is then planned for a subsequent high-resolution *inspection survey*. The goal here is to identify small objects by obtaining short-range, high-resolution scans. Confined areas in the stern and the limited sensor field of view mean that the vehicle movements have to be carefully designed; we have developed an effective sampling-based planning strategy to this end. Once the inspection path is executed, the mesh modeling techniques of the identification survey can be applied again at higher resolution.

Our present work assumes the *a priori* ship model is sufficiently accurate to allow the identification of any and all mines planted on the structure if full coverage is achieved at requisite sonar resolution. This work has been extended to model uncertainties in the *a priori* ship mesh and to prioritize regions of high uncertainty in the selection of sonar scans (Hollinger et al., 2012). This and other inference methods may be suitable in cases where a reliable prior model cannot be obtained. We also note that although our stern inspection methodology may be applied to the flatter, forward sections of the hull, these areas are usually covered trivially by planar, back-and-forth sweep paths designed within the hull-relative coordinate frame.

Obtaining access to large ships for perfecting the whole procedure is not trivial, and we have decided to present in this paper an overview that includes planned paths, computed using sonar-derived models, that were not executed in-water. However, the identification-to-inspection survey pair was carried out successfully on the 82-meter-long *USCGC Seneca* in February 2012, and these results will appear in a later paper.

7.2 Construction of the Identification Mesh

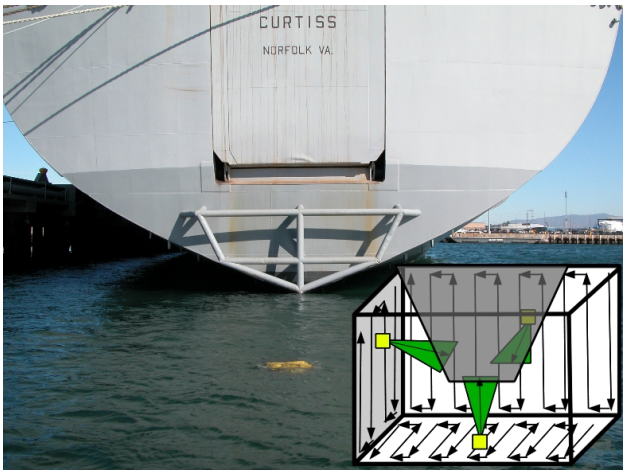
Within the community of laser-based range sensing, specialized algorithms have been designed to generate watertight, 3D mesh models from point clouds (Hoppe et al., 1992; Curless and Levoy, 1996). Laser-based range sensing, ubiquitous in ground, air, and space applications, however, yields substantially higher-resolution point clouds than does underwater acoustic range sensing (Kocak et al., 2008): typically sub-millimeter versus sub-decimeter resolution. Fortunately, a number of derivative tools have been developed for processing point clouds containing gaps, noise, and outliers (Weyrich et al., 2004; Huang et al., 2009), and these provide a direct avenue for us to pursue our identification survey mesh model.

Fig. 15 illustrates the execution and processing of an identification survey from start to finish. First, the HAUV traces out the walls of a safe bounding box providing collision-free observations of the stern; thousands of DIDSON frames are collected along with navigation estimates. Evident in the sonar frames shown is the range noise which makes our task difficult in comparison to laser-based modeling.

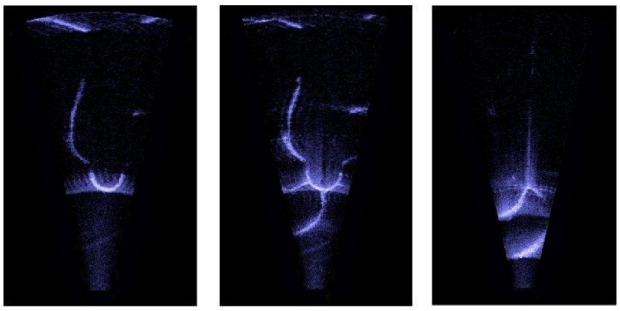
All of the tools used to transform a set of dense, raw-data point cloud slices into a 3D mesh reconstruction can be accessed within *MeshLab* (Cignoni et al., 2008). We first apply a simple outlier filter to the individual sonar frames collected. All points of intensity greater than a specified threshold are introduced into a slice, and then each is referenced using the HAUV's seafloor-relative navigation. Areas containing obvious noise and second returns are cropped out manually. The raw points are then sub-sampled using Poisson disk sampling (Cline et al., 2009), which draws random samples from the point cloud, separated by a specified minimum distance. The point cloud is typically reduced to about 10% of its original density, and then partitioned into separate component point clouds. The partitions are selected based on the likelihood that they will yield individually well-formed surface reconstructions. We note that objects such as rudders, shafts, and propellers are thin structures that usually need special attention such as duplication of a face. Normal vectors are then computed over the component point clouds, by applying principal component analysis to the point's k nearest neighbors; the normal's direction is selected to locally maximize the consistency of vector orientation (Hoppe et al., 1992). Both sub-sampling and estimation of normals are key steps in the processing sequence, and found in practice to significantly impact the accuracy of the mesh (Huang et al., 2009). Sub-sampling generates a low-density, evenly-distributed set of points, and normals aid in defining the curvature of the surface.

The Poisson surface reconstruction algorithm (Kazhdan et al., 2006) is next applied to the oriented point clouds. Octree depth is selected to capture the detail of the ship structures without including excess roughness or curvature due to noise in the data. The component surfaces are merged back together, and a final Poisson surface reconstruction is computed over the components. If the mesh is used as a basis for high-resolution inspection planning, then it may be further subdivided to ensure the triangulation suits the granularity of the inspection task. We iteratively apply the Loop subdivision algorithm (Loop, 1987) for this purpose, dividing each triangle larger than a specified size into four sub-triangles.

The resulting watertight 3D mesh in Fig. 15, comprised of several hundred thousand geometric primi-

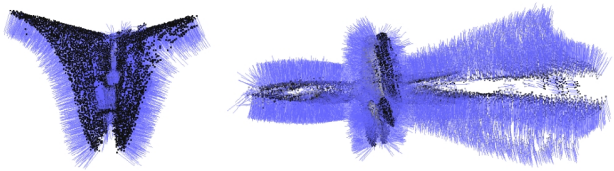


(a) Identification survey in progress on the *SS Curtiss*.

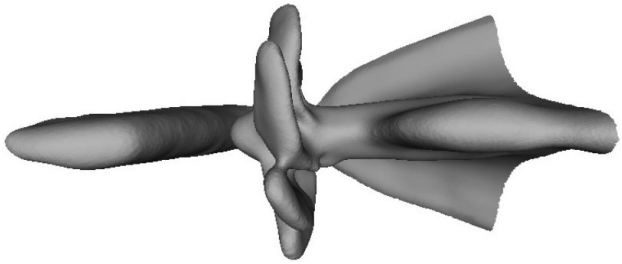


(b) Representative sonar frames from survey of *SS Curtiss* running gear, looking up at the shaft and propeller. The sensor range shown is 1.5-10.5 meters.

(c) Raw-data point clouds obtained from the starboard-side wall and bottom wall of the identification survey, respectively.



(d) Merged, subsampled data displayed with a vertex normal pointing outward from each individual point.



(e) A mesh model of *SS Curtiss* generated by applying the Poisson reconstruction algorithm to the point cloud of (d).

Fig. 15: An overview of the identification survey data and procedure.

tives, is the only data product of its kind that we are aware of, produced from underwater acoustic data.

8 Path Planning for High-Resolution Inspection

To plan the full-coverage inspection survey, we rely on a two-step process. First, a construction procedure obtains a feasible inspection tour; the quality of this initial solution can be improved as a function of available computation time, through building a *redundant roadmap* of designated granularity. Second, if time remains for further planning, an improvement procedure can be used to iteratively shorten the feasible tour. The full planning process is illustrated in Fig. 16. A brief overview of these algorithms is presented in the subsections below; for additional detail we refer the reader to a computational study of the redundant roadmap algorithm (Englot and Hover, 2011), and a computational study of the improvement procedure and analysis of the complete planning method (Englot and Hover, 2012).

8.1 Review of Coverage Path Planning

Coverage path planning applies to tasks such as sensing, cleaning, painting, and plowing, in which an agent sweeps an end effector through some portion of its workspace (Choset, 2001). Planning offers an advantage over greedy, next-best-view strategies when the area to be covered is expansive and known *a priori*.

One common modeling assumption is continuous sensing or deposition by the robot end effector as it executes the coverage path. Planning under this assumption is achieved in obstacle-filled 2D workspaces using cell decomposition methods (Choset, 2000; Huang, 2001), which allow areas of open floorspace to be swept with uninterrupted motions. In three dimensions, the coverage task typically requires a full sweep of the interior or exterior boundary of a structure embedded in the workspace. Back-and-forth sweeping across surface patches (Atkar et al., 2005) and circumferential looping of 2D cross-sections (Acar et al., 2002; Cheng et al., 2008) have been used to cover the full boundary of a closed 3D structure.

An alternative method is to formulate coverage as

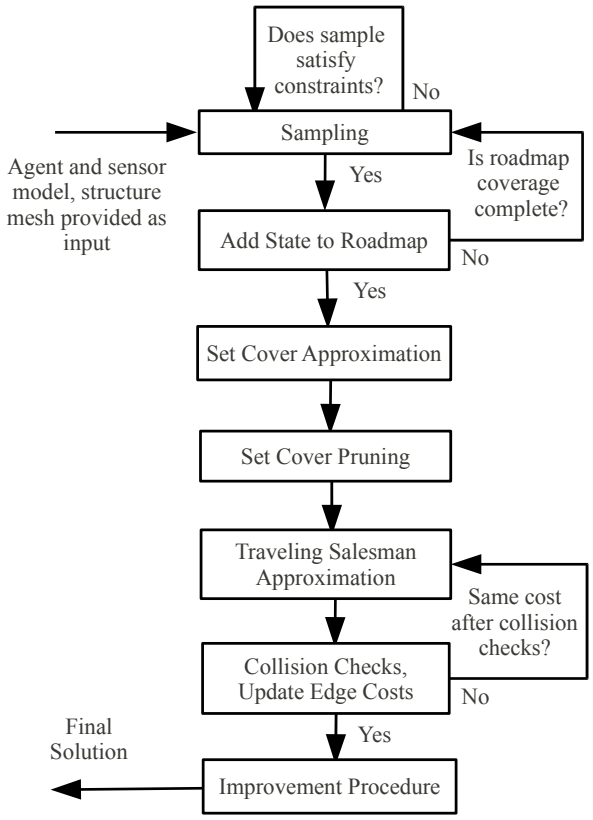


Fig. 16: A stateflow diagram illustrating the construction of an inspection tour using a redundant roadmap, and an improvement procedure in the last step.

a variant of the art gallery problem (Shermer, 1992), in which the structure boundary must be viewed by a minimum-cardinality set of stationary robot configurations. This is often applied to structures with complex, low-clearance, and highly-occluded boundaries. Globally-designed art gallery inspections of 2D and 3D structures have been achieved by randomized, sampling-based algorithms (Danner and Kavraki, 2000), as have 2.5D inspections in which a 3D structure is covered by one or more 2D cross-sections of the workspace (Gonzales-Baños and Latombe, 2001).

The above algorithms can also be categorized using a different characteristic, their modularity. In particular, most of the algorithms solve a single cross-section, cell, or partition without consideration of the neighboring areas. Our application poses a direct challenge to this paradigm at the ship's stern, where shafts, propellers, and rudders lie in close proximity to one another and to the hull. If a 2.5D approach were adopted for coverage planning, it would need to be augmented with special out-of-plane views, to grant visibility of confined areas that are occluded or inaccessible in-plane. If a path were designed for one component, such as a series of loops around a shaft, collision with neighboring structures would be likely.

Considering these factors, we take a global approach in which all 3D protruding structures are considered simultaneously. Rather than explicitly optimizing configurations over the thousands of collision and visibility constraints, our sampling-based planning approach finds feasible means for the HAUV to peer into the low-clearance areas from a distance.

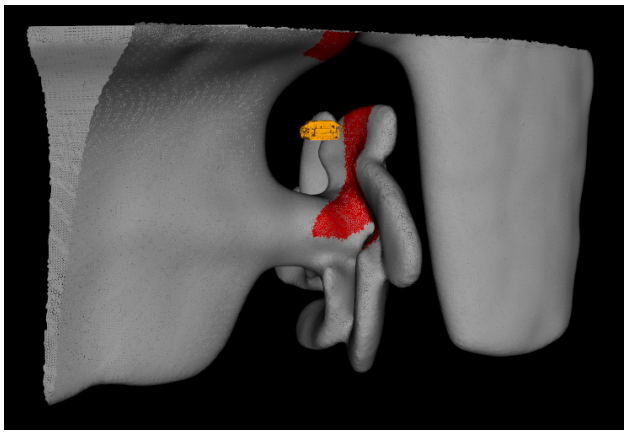
8.2 HAUV-Specific Approach

The format of the inspection tour is designed to be compatible with the HAUV's operational navigation and control capabilities. For instance, ideally we would plan such that sensor observations are collected continuously along every leg of the robot's path in the workspace (Englot and Hover, 2010). In reality, segments cannot be executed with high precision in the presence of hydrodynamic disturbances. But the HAUV can in fact stabilize with good precision at individual waypoints. Consequently our current inspection planning procedure adopts the art gallery approach and produces a set of four-dimensional waypoints that specify Cartesian $[x, y, z]^\top$ and yaw, with each waypoint contributing specific sensor observations.

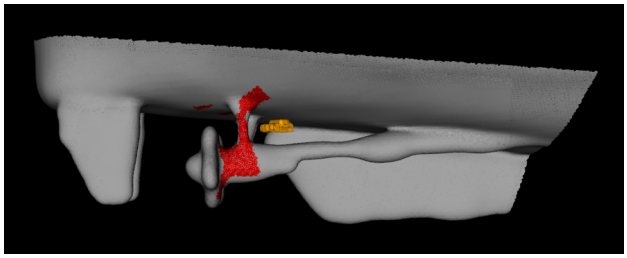
In the mission, it is assumed that at each waypoint, the vehicle stabilizes itself, rotates to the specified yaw angle, and then sweeps the DIDSON through the full arc of available pitch angles, to construct a volumetric range scan. This method of collecting scans isolates the movement of the sensor to a single degree of freedom, easing the burden of interpreting the data and using it for automated (or human-in-the-loop) target classification. The sonar pitch angle limits have varied on the HAUV from model to model, but we will assume that 180° of sonar pitch is available. We also assume that during the inspection survey, the DIDSON is tuned to a resolution that collects individual scans at one- to three-meters range. Examples of volumetric scans collected under these assumptions are illustrated in Fig. 17.

8.3 Construction of Coverage Roadmap and Initial Path

To construct an initial feasible path, robot configurations are sampled and a roadmap iteratively constructed to meet a user-specified redundancy in the sighting of individual primitives. Redundancy increases the density of the covering roadmap, and typical values are one to ten. A combinatorial optimization procedure is then applied to the roadmap, first by approximating the minimum-cardinality set cover (SC) using the greedy algorithm (Johnson, 1974), and pruning to the minimum feasible size. A traveling salesman problem (TSP) tour is then approximated over the set cover, by applying the Christofides approximation (Christofides, 1976) and



(a) *SS Curtiss* mesh, with 107,712 points and 214,419 triangular faces. The propeller is approximately 7 m in diameter.



(b) *USCGC Seneca* mesh, with 131,657 points and 262,173 triangular faces. Each propeller is approximately 2.5 m in diameter.

Fig. 17: The HAUV performing simulated inspection surveys on our mesh models. For the selected robot configurations, each red patch shows mesh points imaged at a desired sensor range between one and three meters, as the sonar sweeps through 180° in pitch.

the chained Lin-Kernighan improvement heuristic (Applegate et al., 2003). Point-to-point edges in the tour are checked for collisions, and the costs of edges in collision are replaced with the costs of point-to-point feasible paths planned using the bi-directional rapidly-exploring random tree (RRT) (Kuffner and LaValle, 2000). Computation of a TSP tour with lazy collision-checking is iteratively repeated until the tour cost stabilizes at a local minimum; this is adapted from a prior method used in multi-goal path planning (Saha et al., 2006).

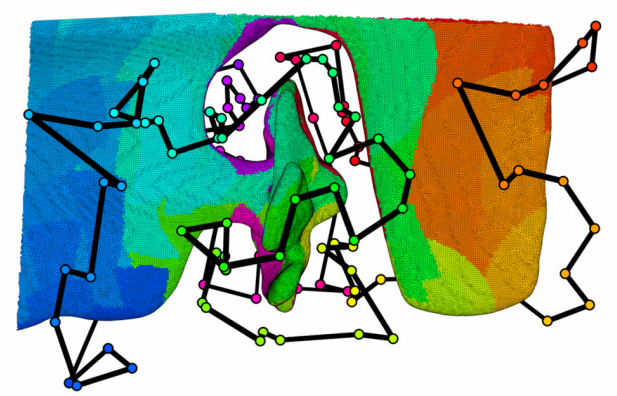
The feasible tour length produced by this approach decreases as the roadmap redundancy increases, but higher-redundancy roadmaps are expensive. More specifically, problems of the size and scale of a ship stern inspection involve a mesh model comprised of about 10^5 geometric primitives, inspected using about 10^2 robot configurations selected from a roadmap containing about 10^3 configurations. Under these conditions we have found ray shooting, the task of checking for line-of-sight between the robot sensor and a geometric primitive, to be the most significant computational burden; it is called millions of times in planning a stern inspection. Our computational studies have shown that for inspections of comparable quality, the redundant roadmap algorithm makes substantially fewer ray shooting queries than the alternative strategy of dual sampling (Gonzales-Baños and Latombe, 2001). This is a greedy approach that has a less demanding combinatorial optimization step, and hence may be more suitable in problems with high combinatorial complexity and minor geometric complexity. We have proven that both our method and the alternative watchman route algorithm (Danner and Kavraki, 2000), which uses dual sampling, are probabilistically complete with respect to the roadmap construction and multi-goal planning subroutines.

8.4 Sampling-Based Improvement Procedure

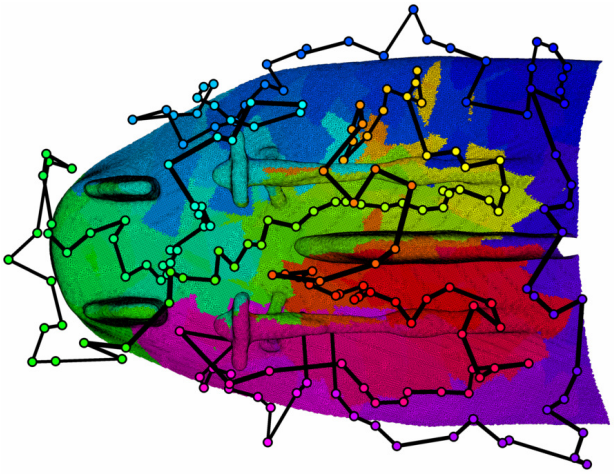
If time allows, a sampling-based improvement procedure can be implemented to further reduce the length of the tour. Rather than addressing the full combinatorial complexity of the combined SC-TSP planning problem, which requires the reordering of states, this procedure aims to physically move tour configurations (HAUV waypoints) into lower-cost locations, while preserving the order. A new configuration is considered feasible if it is collision-free and observes all the geometric primitives that were observed uniquely by its predecessor. A configuration is considered an improvement if it reduces the length of travel to the two adjacent configurations in the tour.

Lower-cost configurations are found using a generalization of the RRT* point-to-point path planning algorithm (Karaman and Frazzoli, 2011). A configuration in the tour is selected at random to be replaced, and two RRT* paths are simultaneously generated that start at the neighbors of this configuration in the tour and share a common goal region, comprised of all feasible replacement configurations. After drawing a designated number of samples, the feasible waypoint that minimizes tour length is selected as the replacement. This heuristic is the most complex adjustment to the tour that can be made without invoking an NP-hard reordering problem in every iteration of the improvement procedure. We have proven this algorithm to be probabilistically complete and asymptotically optimal with respect to the local problem of finding minimum-length paths to two neighbors.

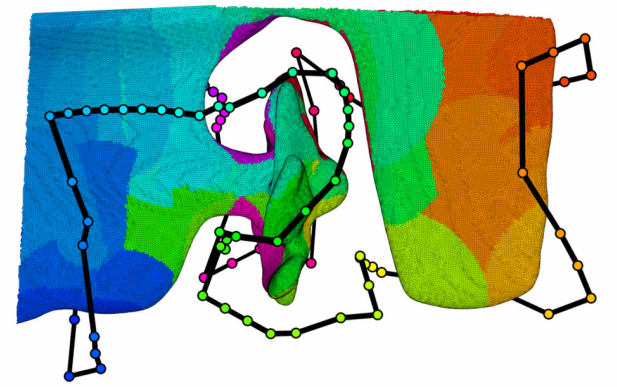
Representative inspection tours for both the *SS Curtiss* and *USCGC Seneca* are depicted in Fig. 18. The tours were planned using the redundant roadmap algorithm to find an initial, feasible path, followed by the sampling-based improvement procedure, over two hours of total allotted computation time. Our studies have shown that over an ensemble of trials of this length, using randomly-generated initial tours with a roadmap redundancy of ten, the improvement proce-



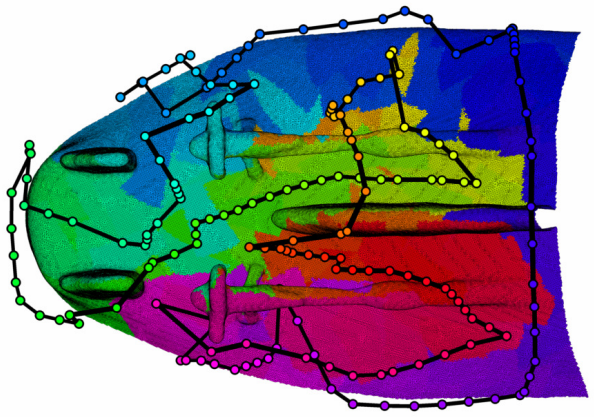
(a) Feasible tour for full coverage of *SS Curtiss* running gear; 176 m in length with 121 configurations.



(b) Feasible tour for full coverage of *USCGC Seneca* running gear; 246 m in length with 192 configurations.



(c) Shortening the tour of (a) using the improvement procedure; 102 m in length with 97 configurations.



(d) Shortening the tour of (b) using the improvement procedure; 157 m in length with 169 configurations.

Fig. 18: Representative full-coverage inspection paths before (top) and after (bottom) the improvement procedure. Waypoints along each tour are color-coded and correspond to the colored patches of sensor information projected onto each ship model. The changes in color occur gradually and follow the sequence of the inspection tour. The thickness of each line segment along the path corresponds to the relative depth of that segment from the viewer's perspective.

dure achieves an average length reduction of 35%. The initial tour computation never required more than one sixth of the total allotted time for the *Seneca*, or more than one thirtieth for the *Curtiss*, which has fewer confined and occluded areas.

9 Conclusion

This paper has described a methodology for performing ship hull inspection, based on the HAUV platform that is expected soon to be produced in quantity for the U.S. Navy. Our work makes contributions in applying feature-relative navigation and control, sonar- and vision-based SLAM, 3D mesh modeling, and coverage motion planning algorithms. On open hull areas, the vehicle navigates in the closed loop using a local hull frame, circumventing difficulties that would be encountered in a conventional approach based on predeployed transponder networks. Dead-reckoning errors are corrected in real-time using a full 6-DOF pose-graph SLAM algorithm that exploits concurrent sonar and camera observations of distinctive features on the hull. We have also developed new techniques in modeling and sampling-based motion planning to achieve full coverage of a complete ship at a resolution adequate to see small objects of interest—on the order of ten centimeters. Our emphasis in planning has been on the

running gear, where simple linear surveys are clearly inadequate. Our experimental results demonstrate the ability to operate effectively on all parts of a vessel.

A great many directions for future development and investigation have emerged from this work. An immediate extension is integrated navigation for the complex areas. Reducing drift through an integrated approach would increase the quality of the data product, and allow smaller objects to be found. Another valuable capability would be multi-session SLAM, which moves beyond constructing maps *in situ* and toward localizing to previously built SLAM maps, continuing to refine them over the periods of days, weeks, and years. Such long-term, high-definition maps then lead to very efficient change detection. We also expect that texture-mapped, large-area surface reconstructions will be of high interest to users. The goal here is to construct accurate hull models using both sonar and camera data, to provide the end-user with a detailed, photo-realistic CAD model of the hull. A task underlying all of these opportunities is the effective processing of noisy acoustic data.

Considering application areas, we believe that the ability to autonomously inspect and interact with marine structures is a strong growth area. Aside from naval ships and harbors, many commercial port authorities, ship operators, and other parties would benefit from improved awareness and security of expensive and life-critical submerged structures. These include for example liquid natural gas (LNG) carriers, tour ship berthing areas, container ports, and hydroelectric plants. Ship husbandry addresses the regular maintenance of a ship, and here, too, autonomous agents can eliminate the need for divers, to assess biofouling and debris. The offshore oil and gas industry is also taking an increased interest in autonomous agents for regular inspection of underwater structures (as mandated by law), and for manipulation. Among high-priority manipulation tasks offshore are the opening and closing of valves on subsea assemblies, placing and retrieving sensor packages, and the cutting and removal of decommissioned rigs. In each of these broad domains, it seems clear that advanced autonomous agents can bring substantial safety improvements and cost savings.

Acknowledgments

This work is supported by the Office of Naval Research under grants N00014-06-10043 and N00014-07-1-0791, monitored by Dr. T.F. Swean, M. Zalesak and V. Steward. We would like to thank J. Vaganay, K. Shurn and M. Elkins of Bluefin Robotics for their exceptional support of our experiments. We also gratefully acknowledge many helpful comments provided by the Reviewers.

References

Acar, E., Choset, H., Rizzi, A., Atkar, P., and Hull, D. (2002). Morse decompositions for coverage tasks. *Int. J. Robot. Res.*, 21(4):331–344.

Applegate, D., Cook, W., and Rohe, A. (2003). Chained Lin-Kernighan for large traveling salesman problems. *INFORMS J. Computing*, 15(1):82–92.

Atkar, P., Greenfield, A., Conner, D., Choset, H., and Rizzi, A. (2005). Uniform coverage of automotive surface patches. *Int. J. Robot. Res.*, 24(11):883–898.

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

Ballard, R., Stager, L., Master, D., Yoerger, D., Mindell, D., Whitcomb, L., Singh, H., and Piechota, D. (2002). Iron age shipwrecks in deep water off Ashkelon, Israel. *Amer. J. Archaeol.*, 106(2):151–168.

Belcher, E., Hanot, W., and Burch, J. (2002). Dual-frequency identification sonar (DIDSON). In *Proc. Intl. Symp. Underwater Technology*, pages 1559–1565, Tokyo, Japan.

Belcher, E., Matsuyama, B., and Trimble, G. (2001). Object identification with acoustic lenses. In *Proc. IEEE/MTS OCEANS Conf. Exhib.*, pages 6–11.

Biber, P. and Strasser, W. (2003). The normal distributions transform: a new approach to laser scan matching. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, volume 3, pages 2743–2748.

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

Cheng, P., Keller, J., and Kumar, V. (2008). Time-optimal UAV trajectory planning for 3d urban structure coverage. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, pages 2750–2757, Nice, France.

Choset, H. (2000). Coverage of known spaces: The boustrophedon cellular decomposition. *Autonomous Robots*, 9(3):247–253.

Choset, H. (2001). Coverage for robotics — a survey of recent results. *Ann. Math. and Artifical Intelligence*, 31:113–26.

Christofides, N. (1976). Worst-case analysis of a new heuristic for the traveling salesman problem. Technical Report CS-93-13, Carnegie Mellon University.

Cignoni, P., Corsini, M., and Ranzuglia, G. (2008). MeshLab: An open-source 3D mesh processing system. *ERCIM News*, 73:45–46.

Cline, D., Jeschke, S., White, K., Razdan, A., and Wonka, P. (2009). Dart throwing on surfaces. *Computer Graphics Forum*, 28(4):1217–1226.

Curless, B. and Levoy, M. (1996). A volumetric method for building complex models from range images. In *Proc. Annual Conf. Comput. Graphics and Interactive Techniques*, pages 303–312, New Orleans, LA.

D'Amaddio, E., Harris, S., Bergeron, E., and Slate, E. (2001).

Method and apparatus for inspecting a submerged structure. U.S. Patent 6,317,387.

Danner, T. and Kavraki, L. (2000). Randomized planning for short inspection paths. In *Proc. IEEE Int. Conf. Robot. and Automation*, volume 2, pages 971–976, San Francisco, CA.

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

Englot, B. and Hover, F. (2010). Inspection planning for sensor coverage of 3D marine structures. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, pages 4412–4417, Taipei, Taiwan.

Englot, B. and Hover, F. (2011). Planning complex inspection tasks using redundant roadmaps. In *Proc. Int. Symp. Robot. Res.*, Flagstaff, AZ.

Englot, B. and Hover, F. (2012). Sampling-based coverage path planning for inspection of complex structures. In *Proc. Int. Conf. Automated Planning and Scheduling*, Sao Paulo, Brazil. Accepted, To Appear.

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

Gonzales-Baños, H. and Latombe, J.-C. (2001). A randomized art gallery algorithm for sensor placement. In *Proc. ACM Symp. on Computational Geometry*, pages 232–240, Medford, MA.

Harris, S. and Slate, E. (1999). Lamp Ray: Ship hull assessment for value, safety, and readiness. In *Proc. IEEE/MTS OCEANS Conf. Exhib.*, volume 1, pages 493–500, Seattle, WA.

Hollinger, G., Englot, B., Hover, F., Mitra, U., and Sukhatme, G. (2012). Uncertainty-driven view planning for underwater inspection. In *Proc. IEEE Int. Conf. Robot. and Automation*, Minneapolis, MN. Accepted, To Appear.

Hoppe, H., DeRose, T., DuChamp, T., McDonald, J., and Stuetzle, W. (1992). Surface reconstruction from unorganized points. In *Proc. Annual Conf. Computer Graphics and Interactive Techniques*, pages 71–78, Chicago, IL.

Hover, F., Vaganay, J., Elkins, M., Willcox, S., Polidoro, V., Morash, J., Damus, R., and Desset, S. (2007). A vehicle system for autonomous relative survey of in-water ships. *Marine Tech. Soc. J.*, 41(2):44–55.

Huang, H., Li, D., Zhang, H., Ascher, U., and Cohen-Or, D. (2009). Consolidation of unorganized point clouds for surface reconstruction. *ACM Trans. Graph.*, 28(5):176:1–176:7.

Huang, W. (2001). Optimal line-sweep-based decompositions for coverage algorithms. In *Proc. IEEE Int. Conf. Robot. and Automation*, volume 1, pages 27–32, Seoul, South Korea.

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

Johnson, D. (1974). Approximation algorithms for combinatorial problems. *J. Comput. System Sci.*, 9:256–278.

Kaess, M. and Dellaert, F. (2009). Covariance recovery from a square root information matrix for data association. *Robot. and Autonmous Syst.*, 57:1198–1210.

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

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

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

Karaman, S. and Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning. *Int. J. Robot. Res.*, 30(7):846–894.

Kazhdan, M., Bolitho, M., and Hoppe, H. (2006). Poisson surface reconstruction. In *Proc. Eurographics Symp. on Geometry*, pages 61–70, Cagliari, Italy.

Kim, A. and Eustice, R. M. (2009). Pose-graph visual SLAM with geometric model selection for autonomous underwater ship hull inspection. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, pages 1559–1565, St. Louis, MO.

Kocak, D., Dalgleish, F., Caimi, F., and Schechner, Y. (2008). A focus on recent developments and trends in underwater imaging. *Marine Tech. Soc. J.*, 42(1):52–67.

Kuffner, J. and LaValle, S. (2000). Rrt-connect: An efficient approach to single-query path planning. In *Proc. IEEE Int. Conf. Robot. and Automation*, volume 2, pages 995–1001, San Francisco, CA.

Loop, C. (1987). Smooth subdivision surfaces based on triangles. Master's thesis, University of Utah.

Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.*, 60(2):91–110.

Menegaldo, L., Santos, M., Ferreira, G., Siqueira, R., and Moscato, L. (2008). SIRUS: A mobile robot for floating production storage and offloading (FPSO) ship hull inspection. In *Proc. Int. Workshop on Advanced Motion Control*, pages 27–32, Trento, Italy.

Negahdaripour, S. and Firoozfam, P. (2006). An ROV stereovision system for ship hull inspection. *IEEE J. Ocean. Eng.*, 31(3):551–546.

Negahdaripour, S., Sekkati, H., and Pirsiavash, H. (2009). Opti-acoustic stereo imaging: On system calibration and 3-D target reconstruction. *IEEE Trans. Image Process.*, 18(6):1203–1214.

Olds, R. B. (2003). Marine mammals systems in support of force protection. In *SSC San Diego Biennial Review 2003*, chapter 3: Intelligence, Surveillance, and Reconnaissance, pages 131–135. Space and Naval Warfare Systems Center, San Diego, San Diego, CA.

Ridao, P., Carreras, M., Ribas, D., and Garcia, R. (2010). Visual inspection of hydroelectric dams using an autonomous underwater vehicle. *J. Field Robot.*, 27(6):759–778.

Saha, M., Roughgarden, T., Latombe, J.-C., and Sanchez-Ante, G. (2006). Planning tours of robotic arms among partitioned goals. *Int. J. Robot. Res.*, 25(3):207–223.

Sherman, J., Davis, R., Owens, W., and Valdes, J. (2001). The autonomous underwater glider 'Spray'. *IEEE J. Ocean. Eng.*, 26(4):437–446.

Shermer, T. (1992). Recent results in art galleries. *Proc. IEEE*, 80(9):1384–1399.

Trimble, G. and Belcher, E. (2002). Ship berthing and hull inspection using the Cetus-II AUV and MIRIS high-resolution sonar. In *Proc. IEEE/MTS OCEANS Conf. Exhib.*, volume 2, pages 1172–1175, Biloxi, MS.

Vaganay, J., Elkins, M., Esposito, D., O'Halloran, W., Hover, F., and Kokko, M. (2006). Ship hull inspection with the HAUV: US Navy and NATO demonstrations results. In *Proc. IEEE/MTS OCEANS Conf. Exhib.*, pages 1–6, Boston, MA.

Vaganay, J., Elkins, M., Willcox, S., Hover, F., Damus, R., Desset, S., Morash, J., and Polidoro, V. (2005). Ship hull inspection by hull-relative navigation and control. In *Proc. IEEE/MTS OCEANS Conf. Exhib.*, volume 1, pages 761–766, Washington, D.C.

Vaganay, J., Gurfinkel, L., Elkins, M., Jankins, D., and Shurn, K. (2009). Hovering autonomous underwater vehicle — system design improvements and performance evaluation results. In *Proc. Int. Symp. Unmanned Untethered Subm. Tech.*, Durham, NH.

Walter, M., Hover, F., and Leonard, J. J. (2008). SLAM for ship hull inspection using exactly sparse extended information filters. In *Proc. IEEE Int. Conf. Robot. and Automation*, pages 1463–1470, Pasadena, CA.

Webb, D., Simonetti, P., and Jones, C. (2001). SLOCUM: An underwater glider propelled by environmental energy. *IEEE J. Ocean. Eng.*, 26(4):447–452.

Weiss, L. (2011). Autonomous robots in the fog of war. *IEEE Spectrum*, 48(8):30–36.

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

Wu, C. (2007). SiftGPU: A GPU implementation of scale invariant feature transform (SIFT). `http://cs.unc.edu/~ccwu/siftgpu`.

Yoerger, D., Jakuba, M., Bradley, A., and Bingham, B. (2007). Techniques for deep sea near bottom survey using an autonomous underwater vehicle. *Springer Tracts in Advanced Robotics*, 28:416–429.

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

Zuiderveld, K. (1994). Contrast limited adaptive histogram equalization. In Heckbert, P., editor, *Graphics Gems IV*, volume IV, pages 474–485. Academic Press, Boston.

A Index to Multimedia Extensions

The multimedia extensions to this article are at: http://www.ijrr.org

Table of Multimedia Extensions

Extension	Type	Description
1	Video	Mapping and waypoint navigation on the *USCGC Seneca*.