Noname manuscript No.
(will be inserted by the editor)

Opportunistic sampling-based active visual SLAM for underwater inspection

Stephen M. Chaves · Ayoung Kim · Enric Galceran · Ryan M. Eustice

Received: January 1, 2017/ Accepted: date

Abstract This paper reports on an active SLAM framework for performing large-scale inspections with an underwater robot. We propose a path planning algorithm integrated with visual SLAM that plans loop-closure paths in order to decrease navigation uncertainty. While loop-closing revisit actions bound the robot's uncertainty, they also lead to redundant area coverage and increased path length. Our proposed opportunistic framework leverages sampling-based techniques and information filtering to plan revisit paths that are coverage efficient. We employ Gaussian process regression for modeling the prediction of camera registrations and use a two-step optimization procedure for selecting revisit actions. We show that the proposed method offers many benefits over existing solutions and good performance for bounding navigation uncertainty in long-term autonomous operations with hybrid simulation experiments and real-world field trials performed by an underwater inspection robot.

Keywords Active SLAM · Sampling-based planning · Gaussian processes · Underwater robotics

S. M. Chaves (✉)
University of Michigan, Ann Arbor, MI, USA
E-mail: schaves@umich.edu

A. Kim
Korea Advanced Institute of Science and Technology, Daejeon, Korea
E-mail: ayoungk@kaist.ac.kr

E. Galceran
ETH Zurich, Zurich, Switzerland
E-mail: enricg@ethz.ch

R. M. Eustice
University of Michigan, Ann Arbor, MI, USA
E-mail: eustice@umich.edu

1 Introduction

Mobile robots operating autonomously in real-world environments must fulfill three core competencies: localization, mapping, and planning. Solutions to these problems are important in order to perform tasks like exploration, search and rescue, inspection, reconnaissance, target-tracking, and others. The core competencies of localization and mapping have been well-studied in the past decades under the topic of simultaneous localization and mapping (SLAM) (Durrant-Whyte and Bailey 2006; Bailey and Durrant-Whyte 2006). However, research toward integrating planning and SLAM solutions to form an all-encompassing framework is still in its early years. SLAM solutions are typically agnostic to the path or control policy given to the robot. Likewise, planning algorithms assume that accurate localization is available and information about the environment is at least partially known. Of course, real-world robots operate at the intersection of these topics where assumptions about the available prior information are not always true, posing an interesting challenge for robust autonomy.

Autonomy in marine environments can be especially difficult. Underwater robots commonly operate in GPS-denied scenarios with limited communication, often over large unknown areas with sparsely-distributed features. In these situations, the path taken by the robot through the environment can drastically affect the performance of SLAM. Thus, real-world robotic systems demand a comprehensive, probabilistic framework for integrated localization, mapping, and planning.

In this paper, we focus on integrating path planning and visual SLAM into a coupled solution, an area of research termed *active* SLAM. Our objective is to provide a robust framework for long-term autonomous inspection in underwater environments. We consider a robot surveying an *a priori* unknown target area subject to a desired navigation

uncertainty threshold, defined as the maximum acceptable robot pose covariance. The robot explores the target environment following a nominal exploration policy—a strategy that ensures efficient area coverage but leads to an open-loop trajectory without loop-closures. Without closing loops in the SLAM formulation, the robot accumulates navigation error and its uncertainty grows unbounded. Hence, we propose an opportunistic active SLAM framework for guiding the robot to make loop-closure revisit actions throughout the mission. Specifically, our proposed method

1. employs Gaussian process (GP) regression for probabilistic modeling of the predicted visual loop-closure utility of unmapped areas,
2. combines sampling-based planning with information filtering to efficiently search hundreds of revisit paths through the environment for their expected utility, and
3. uses a two-step optimization process for selecting revisit actions that yields an opportunistic active SLAM framework.

An illustration of the proposed algorithm is given in Fig. 1. Our method is opportunistic in that it trades off convenience versus necessity; it does not wait until the uncertainty threshold is reached when selecting revisit actions to close loops in the SLAM formulation.

The remainder of the paper is organized as follows: related work is presented in §2, followed by the problem statement in §3. The proposed method for predicting visual saliency is detailed in §4. Revisit path evaluation is described in §5 before explaining how the proposed algorithm searches for candidate paths in §6. Finally, we validate the proposed method using a hybrid simulation in §7 and real-world field trials with an underwater inspection robot in §8. Concluding remarks are given in §9. An earlier version of this work was presented by Chaves et al (2014). Extending the previous work, we present (*i*) a more detailed description and an updated implementation of the framework and (*ii*) results from real-world field trials of the system that validate our approach. The field trials were performed in a controlled laboratory environment at the University of Michigan's Marine Hydrodynamics Laboratory (MHL) and while inspecting the hull of the *USCGC Escanaba* in Boston, MA.

2 Related Work

Research in the area of active SLAM stems from the seminal works of Bajcsy (1988), Whaite and Ferrie (1997), and Feder et al (1999) and generally focuses on finding control actions to reduce the uncertainty or maximize the information of the SLAM posterior. Sim and Roy (2005) did not include a measure of path length in the objective function, only optimizing the trace of the SLAM covariance matrix. Others included path length by focusing on reducing uncertainty while performing efficient exploration (Gonzalez-Banos and Latombe 2002; Bourgault et al 2002; Stachniss et al 2005). Our formulation is similar in that we minimize an objective that considers both path length and robot navigation uncertainty, but we additionally consider the stochasticity of expected camera measurements when evaluating the utility of a loop-closure path. Valencia et al (2011, 2012, 2013) presented an active SLAM framework that is close to our own in that they use information filtering to evaluate the uncertainty of revisit actions on a pose-graph; however, their formulation only enumerates a small number of possible options when deciding the best action. In contrast, we leverage sampling-based planning to quickly explore the configuration space of the robot with hundreds of candidate revisit paths. In the context of bathymetric mapping, Galceran et al (2013) proposed a survey planning method that minimizes an autonomous underwater vehicle (AUV)'s pose uncertainty during the mission by revisiting salient locations in the environment, while accounting for coverage efficiency and meeting the application constraint of performing the survey in parallel tracks. However, their method is targeted at the particulars of bathymetric surveys and is not integrated with a SLAM framework. We are also unable to modify the execution order of parallel tracks in the nominal exploration policy due to operational constraints with our robot.

Sampling-based planning originates from traditional motion planners like the Rapidly-exploring Random Tree (RRT) (LaValle and Kuffner 2001) and Probabilistic Roadmap (PRM) (Kavraki et al 1996), but was only recently extended to include a notion of uncertainty during the planning process (Prentice and Roy 2009). Karaman and Frazzoli (2011) proposed the Rapidly-exploring Random Graph (RRG) and RRT*—adaptations of the RRT that incrementally improve the shortest-distance path toward the goal with guarantees of optimality in the limit. Bry and Roy (2011) extended the RRG further to include stochasticity of measurements and robot dynamics, a formulation generalized by Hollinger and Sukhatme (2014) to accomplish information gathering. We take a similar approach and adopt the use of the RRG, but present a proposed algorithm integrated with SLAM.

Recently, a substantial body of promising work has used optimal control techniques and trajectory smoothing for handling robot uncertainty during point-to-point planning queries (van den Berg et al 2012; Indelman et al 2015; Patil et al 2014). These methods result in locally-optimal solutions but have yet to be extended to account for area coverage and more challenging real-world robotic systems.

In order to predict the likelihood of making camera measurements over unmapped areas, we use GP regression and visual saliency to model the loop-closure utility of the environment (Rasmussen and Williams 2006). Recently, GPs

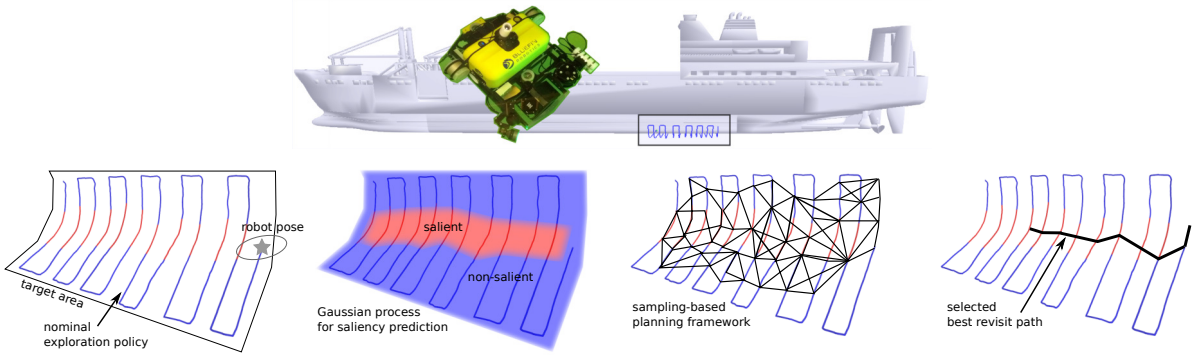


Fig. 1 Overview of the proposed active SLAM framework, demonstrated for ship hull inspection. Given a target area and nominal exploration policy, the robot explores the environment subject to an acceptable navigation uncertainty. We use GP regression to predict the visual saliency of the environment and a sampling-based planning algorithm to find and evaluate loop-closure revisit paths in order to drive down the robot's uncertainty.

have been widely used in learning spatial distributions and predicting sensor data. O'Callaghan and Ramos (2012) accurately predict an occupancy map using a GP with a trained neural network kernel, despite dealing with noisy and sparse sensor information. Within marine environments, Barkby et al (2011) used a GP to predict bathymetric data in unmapped patches in order to perform SLAM without any actual sensor overlap. They achieve tractable prediction with the GP over a large dataset by using a sparse covariance function (Melkumyan and Ramos 2009).

Much of the work in this paper builds upon the perception-driven navigation (PDN) method for active visual SLAM presented by Kim and Eustice (2015). The PDN approach considers the same problem formulation, however it plans revisit paths by clustering visually salient nodes in the SLAM pose-graph to form waypoints, planning a saliency-weighted direct path to each waypoint, and computing a reward for revisiting along each path. In contrast, the proposed algorithm reported here improves upon this method by removing the need for clustering while evaluating many more paths using sampling-based planning techniques. In addition, instead of the simple interpolation scheme used by PDN, we predict the visual saliency along paths with a more principled approach based upon GP regression. Our algorithm is also opportunistic, unlike PDN, in that it does not wait until the uncertainty threshold is reached to execute a beneficial revisit action.

3 Problem Formulation

We consider a robot performing pose-graph SLAM to survey an *a priori* unknown environment, subject to a desired uncertainty threshold. We are interested in performing large-scale underwater inspections such that the inspection quality is coupled to the value of the uncertainty threshold. Our primary application is ship hull inspection with a Hovering Autonomous Underwater Vehicle (HAUV) (Hover et al 2012), where the overall goal is to produce a metrically-accurate map of the underwater portion of a ship. Since the HAUV moves slowly and has control authority along all three principal axes and in heading, we assume holonomic kinematic constraints and no adverse effects from abrupt changes in direction. The robot is equipped with proprioceptive sensors to perform dead-reckoning and exteroceptive sensors for perception, which define the robot's ability to survey the environment and gather loop-closure measurements. The HAUV uses a Doppler velocity log (DVL) and depth sensor for dead-reckoning and is equipped with an underwater camera and imaging sonar for capturing images. These sensors are positioned on an actuated tray on the robot that rotates to always point normal to the inspection surface. Ranges from the DVL allow the robot to maintain a uniform standoff over the ship hull. While we showcase surveys in this paper for dense sonar coverage, we use camera images to close loops between poses in the SLAM formulation by making registrations between overlapping images (via a homography mapping or the Essential matrix). For ease of incorporation into our existing SLAM system, we adopt the incremental smoothing and mapping (iSAM) algorithm by Kaess et al (2008, 2013) as the SLAM back-end for our pose-graph. iSAM performs nonlinear inference over the factor graph containing constraints from odometry and camera image registrations (Hover et al 2012). The resulting estimate of the pose-graph can be represented as a multivariate Gaussian with mean vector of 6-degree of freedom (DOF) poses, $\mu_{\texttt{slam}}$, and posterior information matrix, $\Lambda_{\texttt{slam}}$.

The inspection problem is then formulated with the following assumptions:

1. The boundaries of the target coverage area are given.
2. An open-loop nominal exploration policy provides dense coverage of the target area in efficient time (lawnmower tracklines for sonar coverage with the HAUV).

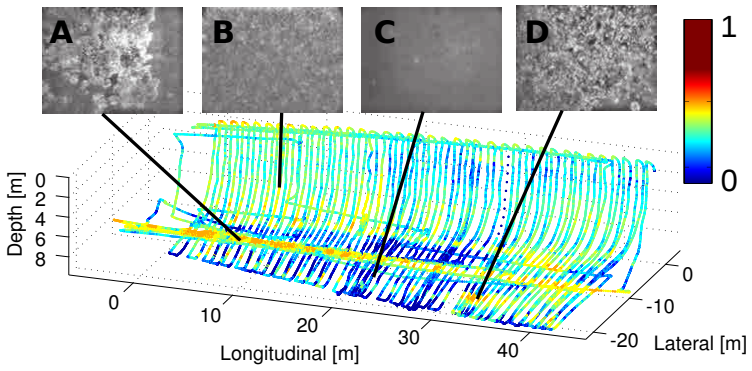


Fig. 2 Representative sample images captured on the below-water portion of the *SS Curtiss*' ship hull. The underwater feature distribution is highly diverse, ranging from feature-less (B,C) to feature-rich (A,D) images. The robot poses along the trajectory are color-coded by their local saliency score, S_L.

3. The user defines a desired navigation uncertainty threshold as a function of the robot pose covariance.
4. The environment is free of obstacles and the inspection surface is locally planar. The effects of water current and turbidity are assumed to be negligible.
5. No other prior information is provided. Planning is performed online in conjunction with SLAM.

Considering these assumptions, the goal of our proposed active SLAM algorithm is to find and execute loop-closure revisit actions throughout the mission in order to bound navigation uncertainty and maintain efficient area coverage. Without this type of active SLAM system, the inspection task suffers from growing navigation uncertainty over long durations. In these cases of long open-loop behavior, modeling inaccuracies in the SLAM system cause the robot to risk losing localization, leading to data association difficulties when proposing loop-closures. With a lost robot, even appearance-based approaches for finding loop-closures consistently fail in the underwater environment (Ozog et al 2015). Thus, we adopt an incremental approach where multiple loop-closure actions are necessary and we can guarantee an acceptable inspection quality by bounding the navigation uncertainty throughout the mission. We must integrate planning into the SLAM system to allow for long-term autonomy in this marine application.

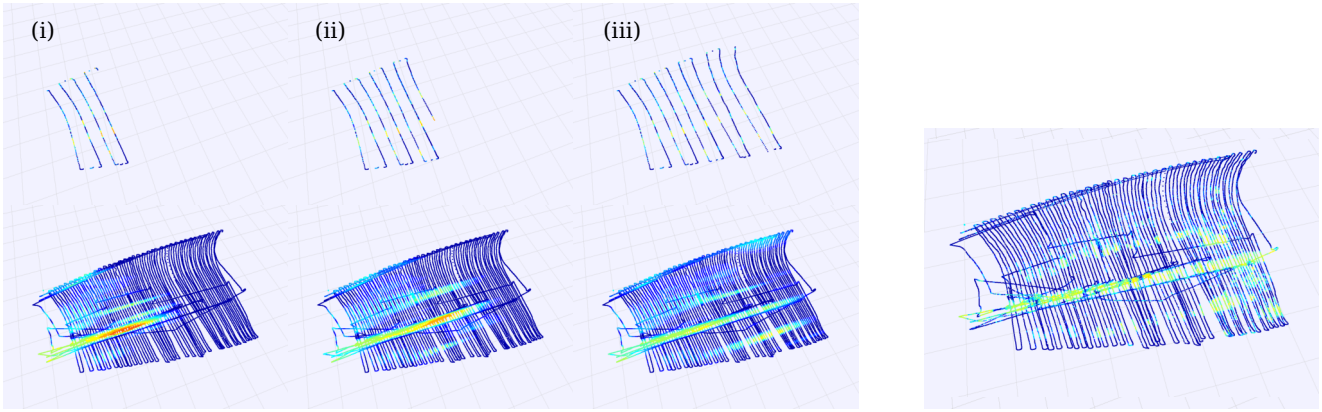
The next two sections explain mathematical details pertaining to visual saliency prediction and revisit path evaluation, respectively. Then, we describe algorithmically how the proposed method finds candidate revisit paths in §6.

4 Visual Saliency Prediction

The utility of a revisit path in our active SLAM formulation is defined by its ability to reduce navigation uncertainty, and thus heavily depends on the expected loop-closures. To gather loop-closures in our visual SLAM formulation, the robot must execute revisit paths that re-observe previously-seen portions of the environment. However, feature-richness in underwater environments is highly sporadic and images acquired by the robot typically have a variable degree of registrability. As shown in the sample images in Fig. 2, good underwater visual features tend to be sparsely distributed, such that much of the environment has little utility for closing loops, or even none at all.

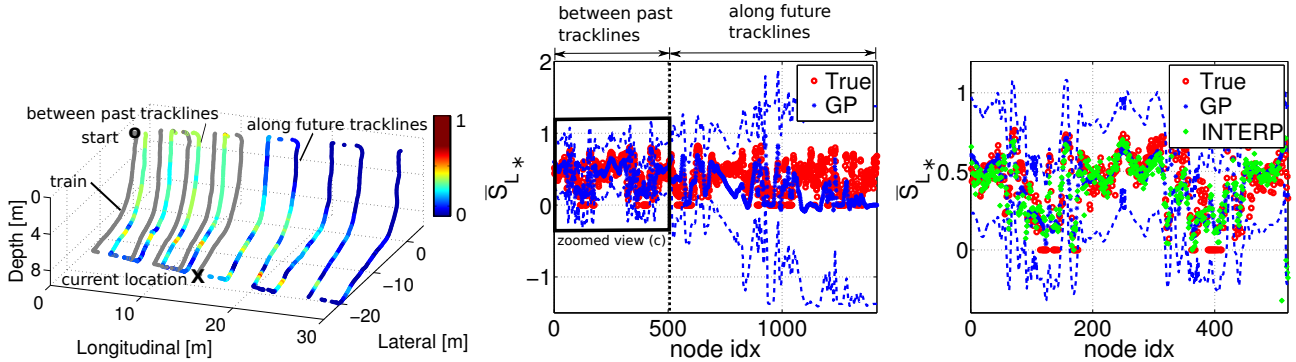
To aid in estimating the likelihood of successful registrations, we adopt the visual saliency metrics defined by Kim and Eustice (2013), including *local saliency* (S_L). The local saliency score is computed using the entropy of a bag-of-words (BoW) histogram from a vocabulary built online and takes on values ranging from 0 to 1 (non-salient to very salient). This score describes the texture-richness of an image and directly correlates to its registrability. The saliency score can only be calculated for images actually captured by the robot, and must be predicted for virtual poses along not-yet-traveled revisit paths. Thus, estimating successful registrations hinges on accurate saliency prediction.

Previously, PDN used a simple interpolation scheme for predicting saliency scores throughout an unmapped area. However, this method has no notion of whether the predicted score is accurate, and failure to accurately predict can lead to planned revisit actions that greatly overestimate their actual utility. Instead, here, we handle saliency prediction using GP regression—a more principled approach that accounts for the spatial distribution of visual features and yields distributions of the saliency values to include in the planning formulation. The variance of the prediction provided by the GP serves as a measure of accuracy for the predicted saliency score, and is accounted for during path evaluation (described later). More information on GP regression for learning can be found in Rasmussen and Williams (2006).



(a) Predicted saliency maps throughout a mission from GP regression (b) Actual saliency map

Fig. 3 (a) Predicted saliency maps from GP regression at different points during an example mission: (i) after 25% coverage with the nominal mission policy, (ii) 50% coverage, and (iii) 75% coverage. The training data is shown along the top row, with predicted saliency maps along the bottom row. (b) The actual saliency map of the environment, for comparison.



(a) Sampled trajectory (b) Variance (c) Zoomed view

Fig. 4 (a) The predicted saliency map from the GP along a sampled trajectory. The 'X' indicates the current robot location while gray dots represent the historical poses of the robot for training the GP. The predicted saliency scores are color-coded with respect to their saliency level. (b) Predicted mean saliency values with variance, overlaid on the true saliency scores. We intentionally excluded interpolated results in order to clearly present how the predicted covariance envelope captures the true value. (c) A zoomed view for predictions between past tracklines. Two dotted blue lines indicate the 3-σ envelope of the prediction. Red circles are the true saliency scores and green diamonds are the interpolated prediction used by PDN.

4.1 Saliency Prediction using Gaussian Process Regression

The objective of using GP regression is to probabilistically handle expected camera registrations over unmapped areas. As the robot follows the nominal exploration policy, we train the GP using the captured images in association with their 3D capture locations. The training data $\mathscr{D} = \{\mathrm{X}, \mathrm{y}\}$ is composed of the 3D positions of the robot at the n capture times, $\mathrm{X} = \{X_i\}_{i=1}^n = \{x_i, y_i, z_i\}_{i=1}^n$, and the local saliency scores of the n images, $\mathrm{y} = \{S_{L_i}\}_{i=1}^n$. Similar to Barkby et al (2011), we use a simple stationary covariance function suitable for large scale data (Melkumyan and Ramos 2009):

$$k(X,X';l,\sigma_0) = \begin{cases} \sigma_0 \left(\frac{2+\cos(2\pi d/l)}{3}(1-\frac{d}{l}) + \frac{\sin(2\pi d/l)}{2\pi} \right), & \text{if } d < l \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

This covariance function achieves sparsity and scalability by truncating values for training pairs that exceed the weighted distance $d = |X - X'|_W = [(X-X')^\top W (X-X')]^{1/2}$. For our application, we align the coordinate frame with the survey area and choose $W = \mathrm{diag}(1/\Delta_d^2, 1/\Delta_h^2, 1)$. Here, Δ_d is the distance between two along-track SLAM poses associated with the nominal exploration policy and Δ_h is the distance between two cross-track SLAM poses.

The GP prediction performance over unmapped areas is shown in Fig. 3. The top row of Fig. 3(a) depicts the pose-graph of the robot at various points throughout a mission while following the nominal exploration policy. The bottom row of Fig. 3(a) shows dense coverage of saliency predictions from the GP both in the future—throughout the remaining survey area, and in the past—in unmapped areas between previously-traveled tracklines. Fig. 4 shows the mean

and variance from GP prediction along a sampled trajectory, considering the robot already surveyed five full tracklines. Fig. 4(b) shows that the variance increases as the predictions transition to farther-out areas not yet reached by the robot. Between past tracklines, the saliency prediction closely matches the actual value with small variance, shown closer in Fig. 4(c). This zoomed view also compares the GP prediction to the linear interpolation scheme from PDN in predicting mean values; however, the true strength of using the GP is in the measure of variance it also provides.

Together, the predicted mean and variance characterize a Gaussian probability density function (pdf) of the saliency score of a not-yet-seen image at the queried pose. Now that we have a probabilistic method for modeling the saliency throughout the environment, we focus on evaluating the utility of a candidate revisit path.

5 Revisit Path Evaluation

For every candidate revisit path, $\mathscr{P}$, we use a linearized snapshot of the SLAM system to construct a delayed-state extended information filter (EIF) (Eustice et al 2006a) that tracks the posterior distribution of the path with information $\Lambda_{\mathtt{path}}$. The posterior information is the sum of the prior SLAM information, $\Lambda_{\mathtt{slam}}$, and sources of delta information corresponding to the the expected odometry and camera registrations available along the round-trip path. In this way, a candidate revisit path is distributed as a multivariate Gaussian parameterized by a mean vector of both real (SLAM) and virtual (revisit path) poses, $\mu_{\mathtt{path}} = \mu_{\mathtt{slam}} \cup \{\mathrm{x}_{v_0}, \cdots, \mathrm{x}_{v_{p-1}}\}$, and the posterior information, given by

$$\Lambda_{\mathtt{path}} = \Lambda_{\mathtt{slam}} + \Lambda_{\mathtt{odo}} + \Lambda_{\mathtt{cam}}. \tag{2}$$

A toy example of how the information is summed along a path is given in Fig. 5. Our method assumes maximum-likelihood measurements that occur at the mean of the distribution, allowing for the incorporation of expected measurements within the EIF without relinearization.

An odometry measurement, $\mathrm{x}_{v_i,v_{i+1}}$, is the relative-pose increment between sequential virtual poses in the path, $(\mathrm{x}_{v_i}, \mathrm{x}_{v_{i+1}}) \in \mathscr{P}$ (Smith et al 1990):

$$\mathrm{x}_{v_{i+1}} = \mathrm{x}_{v_i} \oplus \mathrm{x}_{v_i,v_{i+1}}. \tag{3}$$

Adding the expected odometry measurements along the path results in block-tridiagonal delta information given by

$$\Lambda_{\mathtt{odo}} = \sum_{i=0}^{p-1} \mathrm{H}^{\top}_{\mathtt{odo}_{i,i+1}} \mathrm{Q}^{-1}_{\mathtt{odo}_{i,i+1}} \mathrm{H}_{\mathtt{odo}_{i,i+1}}, \tag{4}$$

where $\mathrm{H}_{\mathtt{odo}_{i,i+1}}$ is the sparse Jacobian and $\mathrm{Q}_{\mathtt{odo}_{i,i+1}}$ is the noise of the odometry model.

The delta information from expected camera registrations along a path has a similar expression, but requires more care. We adopt the "link proposal" method from Kim and Eustice (2013) for proposing camera registration hypotheses between a virtual pose on a revisit path and n_{plink} existing target poses in the SLAM pose-graph that may contain spatially-overlapping image views. For each link hypothesis, we can compute its probability of successful registration, P_L, using the GP prediction and the knowledge of past registration attempts. We aggregate historical camera registration data to produce an empirical probability of successful registration as a function of target pose saliency and virtual pose saliency for overlapping pairs, $g(S_{L_t}, S_{L_v})$, shown in Fig. 6 and described further by Kim and Eustice (2013). The GP prediction returns a distribution of virtual pose saliency scores at the queried pose with pdf $f(S_{L_v})$, which we transform into the censored pdf $f'(S_{L_v})$ (Hand 2008). The censored pdf ensures that the predicted saliency is correctly represented within the acceptable range of values from 0 to 1. Then, the expected probability of the link is computed as

$$\begin{aligned} P_L &= \mathbb{E}_{f'}\left[g(S_{L_t}, S_{L_v})\right] \\ &= \int_0^1 g(S_{L_t}, S_{L_v}) f'(S_{L_v}) dS_{L_v}. \end{aligned} \tag{5}$$

The delta information corresponding to camera registrations expected along a revisit path is then calculated by

$$\Lambda_{\mathtt{cam}} = \sum_{i=0}^{p-1} \sum_{t \in \mathscr{L}_i} P_L \cdot \mathrm{H}^{\top}_{\mathtt{cam}_{t,i}} \mathrm{R}^{-1} \mathrm{H}_{\mathtt{cam}_{t,i}}, \tag{6}$$

where $\mathrm{H}_{\mathtt{cam}_{t,i}}$ is the camera measurement Jacobian (Kim and Eustice 2013), R is the camera measurement noise (assumed constant for convenience), and $\mathscr{L}_i$ is the index set of camera registrations associated with virtual pose x_{v_i}. The information of an expected camera registration is scaled by its probability of success as a method for handling the stochasticity of achieving the measurement in the prediction. This information scaling was first practiced by Kim and Eustice (2015), but Indelman et al (2015) formalized the intuition by showing that scaling the information results from using Expectation-Maximization as a way of estimating a latent variable describing whether or not the measurement is acquired. In the harsh marine environment where camera registrations are difficult to make, this information scaling is essential to the prediction accuracy of the proposed method.

Given the posterior information, $\Lambda_{\mathtt{path}}$, the cost function we use to determine the utility of a revisit path reflects the planning tradeoff between navigation uncertainty and area coverage. The planning objective is to minimize the cost of a revisit path, computed by

$$\mathscr{C}(\mathscr{P}) = \alpha \cdot \frac{\mathscr{U}(\mathscr{P})}{\mathscr{U}_{\mathtt{upper}}} + (1-\alpha) \cdot \frac{d(\mathscr{P})}{d_{\mathtt{upper}}}, \tag{7}$$

where $\mathscr{U}(\mathscr{P}) = s(\Sigma_{nn})$ is a function of the terminating pose covariance of the path and used as a measure of navigation

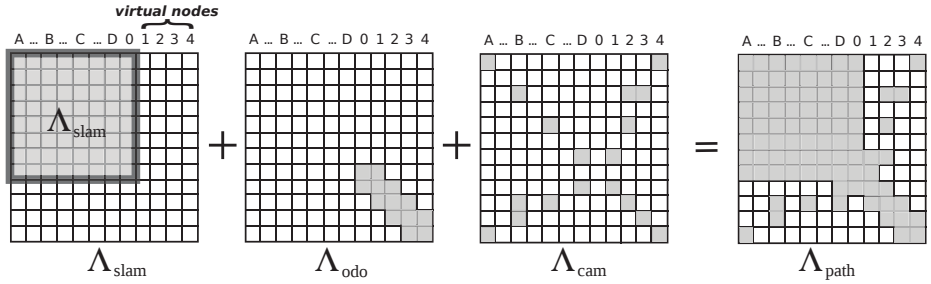
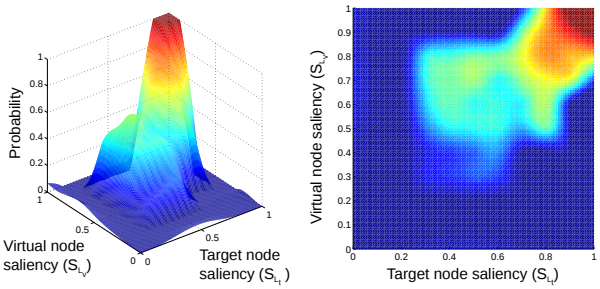


Fig. 5 The path posterior information is found by adding sources of delta information from odometry measurements and camera registrations to the prior SLAM information. The odometry measurements have a block-tridiagonal structure and camera registrations result in loop-closing constraints to previous poses in the pose-graph.



(a) Link probability function, $g(S_{L_t}, S_{L_v})$ (b) Top view

Fig. 6 The empirical probability of the camera registration to be successful, used in (5). The probability is a function of target node saliency S_{L_t} and virtual node saliency S_{L_v}.

uncertainty. We express the covariance in the frame of the initial robot pose, which is aligned with the frame of the global origin. Either the A-optimal trace or the D-optimal determinant can be used for $s(\cdot)$ (Sim and Roy 2005). We use $s(\Sigma_{nn}) = |\Sigma_{nn}|^{\frac{1}{6}}$ throughout the experiments in this paper to yield a quantity with units of m-rad. The uncertainty upper bound ($\mathscr{U}_{\texttt{upper}}$) is specified by the user as the desired threshold to be maintained during the exploration of new areas throughout the mission. Note that our method does allow for the threshold to be exceeded during diversions from exploration along revisit paths, provided the robot returns to exploring with the nominal policy with bounded uncertainty. $d(\mathscr{P})$ is the revisit (redundant coverage) distance of the path and $d_{\texttt{upper}}$ is the maximum revisit distance allowed by the user. The tuning parameter $\alpha \in [0,1]$ controls the balance between uncertainty and revisit distance.

Per Eustice et al (2006b), we recover the terminating covariance of a revisit path, Σ_{nn}, by

$$\Lambda_{\texttt{path}}\Sigma_{*n} = \mathrm{I}_{*n}, \tag{8}$$

where Σ_{*n} and I_{*n} are the n^{th} block-columns of the covariance matrix and block identity matrix, respectively. Since the information matrix is exactly sparse for our visual SLAM formulation, this calculation can be performed efficiently using sparse Cholesky factorization.

6 Planning Algorithm Description

We now discuss algorithmically how the proposed method searches for candidate revisit paths, prunes paths that are not promising, and selects which path to execute. We leverage a sampling-based approach to quickly explore the configuration space of the robot while evaluating candidate paths using the formulation from §5. The planning algorithm is described in detail below and outlined in Algorithm 1.

6.1 Graph Construction

The underlying structure for the path planner is the RRG (Karaman and Frazzoli 2011; Bry and Roy 2011; Hollinger and Sukhatme 2014), which incrementally builds a roadmap of vertices and edges describing connectivity through the configuration space of the robot. Contained at each vertex in the RRG is a list of partial candidate revisit paths (hereafter called just *candidates*, denoted by $\mathscr{P}_i$) that each describe a unique trajectory over edges in the RRG to arrive at the vertex. Every candidate at every vertex is tracked by the planner and represented by its mean, $\mu_{\texttt{path}}$, and associated information, $\Lambda_{\texttt{path}}$, from (2). The virtual poses in the mean vector, $\{\mathrm{x}_{v_0}, \cdots, \mathrm{x}_{v_{p-1}}\}$, arise from traveling along edges, and the information matrix $\Lambda_{\texttt{path}}$ is the sum of the SLAM information, $\Lambda_{\texttt{slam}}$, and the delta information gathered along the way. Fig. 7 displays an example RRG sampled on a typical SLAM pose-graph built by the HAUV.

A benefit of using the information form to track candidates is that each of the sources of delta information from (2) can be divided into multiple components and attributed to the edges from which they originate. In this way, the total delta information for a candidate is simply the sum of the delta information contributed by each edge that it travels. Hence, (2) can be rewritten as

$$\Lambda_{\texttt{path}} = \Lambda_{\texttt{slam}} + \Lambda^1_{\texttt{edge}} + \ldots + \Lambda^k_{\texttt{edge}}, \tag{9}$$

where

$$\Lambda^i_{\texttt{edge}} = \Lambda^i_{\texttt{odo}} + \Lambda^i_{\texttt{cam}}, \tag{10}$$

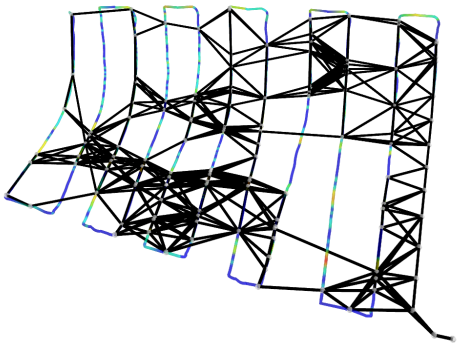


Fig. 7 An example RRG sampled over a typical SLAM pose-graph from a ship hull inspection. Black lines represent edges between vertices in the RRG.

and $\Lambda^i_{\texttt{edge}}$ represents the delta information matrix encoded by the odometry and camera registration factors arising from the edge. A key insight is that the delta information $\Lambda^i_{\texttt{edge}}$ only needs to be computed once during construction, and is additively applied to a candidate's information when the edge is traversed. This benefit of the information form was alluded to by Valencia et al (2013) and is analogous to the one-step transfer function used by the Belief Roadmap (BRM) (Prentice and Roy 2009). Edge construction is accomplished within the `Connect()` function of the proposed algorithm.

6.2 Path Propagation and Pruning

As the RRG is built, the algorithm grows a tree of candidates over the roadmap, where candidates can be thought of as leaves of the tree. The root of the tree is initialized at the most recent SLAM pose with initial information $\Lambda_{\texttt{slam}}$. New leaves are generated by creating a new vertex from a sampled pose in the configuration space (v_{new}), connecting nearby vertices (V_{near}) to the new vertex with new edges, and propagating the candidates from the nearby vertices over the new edges and recursively throughout the rest of the graph. Propagating a candidate over an edge (and hence creating a new leaf) amounts to branching the candidate (parent leaf) from the edge source vertex, creating a new candidate (child leaf) at the destination vertex, and calculating the child information by adding the delta information from the edge to the parent leaf information. Without pruning leaves, the tree encodes every possible path through the graph to reach any vertex from the root.

However, the number of candidates tracked by the planner can quickly become too large for computational feasibility. It is logical to prune leaves of the tree that are not useful given the objective. A conservative pruning strategy would eliminate only suboptimal leaves from the tree (Bry and Roy 2011; Hollinger and Sukhatme 2014), but we are willing to employ a more aggressive heuristic that sacrifices optimality for a large increase in speed, as done by Hollinger and Sukhatme (2014) for submodular objective functions. Thus, we maintain a partial ordering of candidates at each vertex according to the distance and uncertainty metrics of (7):

$$\mathscr{P}_a > \mathscr{P}_b \Leftrightarrow d(\mathscr{P}_a) < d(\mathscr{P}_b) \wedge \mathscr{U}(\mathscr{P}_a) < \mathscr{U}(\mathscr{P}_b) + \varepsilon, \tag{11}$$

where ε is a small factor to aid in pruning when the candidates are quite similar (Bry and Roy 2011). When (11) is true, candidate $\mathscr{P}_b$ has both a longer revisit distance and a higher uncertainty than $\mathscr{P}_a$, so $\mathscr{P}_b$ can be pruned from the vertex, as well as its children.

Algorithm 1 Opportunistic Path Planning Algorithm

Input: current pose x_0, SLAM pose-graph, exploration policy
Output: best path $\mathscr{P}*$
Initialize: vertices $V = \{v_0\}$, edges $E = \{\}$, queue $Q = \{\}$

for x_l in look-ahead poses **do**
  $v_{new} = \texttt{ExtendToNearest}(\mathrm{x}_l)$
  while Q is not empty **do**
    `ProcessQueue()`
  end while
end for
`ComputeMaxCost()`
while computation time remains **do**
  $\mathrm{x}_{sample} = \texttt{SamplePose}()$
  $v_{new} = \texttt{ExtendToNearest}(\mathrm{x}_{sample})$
  $V_{near} = \texttt{FindNearVertices}(v_{new})$
  for all v_k in V_{near} **do**
    $E = E \cup \{\texttt{Connect}(v_k, v_{new}), \texttt{Connect}(v_{new}, v_k)\}$
    $Q = Q \cup \{\text{all } \mathscr{P} \text{ at } v_k\}$
  end for
  while Q is not empty **do**
    `ProcessQueue()`
  end while
  `UpdateBestPath()`
end while

Function: $\texttt{ExtendToNearest}(\mathrm{x}_i)$
  $v_{nearest} = \texttt{FindNearestVertex}(\mathrm{x}_i)$
  $v_{new} = \texttt{SteerToward}(\mathrm{x}_i, v_{nearest})$
  $V = V \cup v_{new}$
  $E = E \cup \{\texttt{Connect}(v_{nearest}, v_{new}), \texttt{Connect}(v_{new}, v_{nearest})\}$
  $Q = Q \cup \{\text{all } \mathscr{P} \text{ at } v_{nearest}\}$
  return v_{new}

Function: `ProcessQueue()`
  $\mathscr{P}_{parent} = \texttt{Pop}(Q)$
  for all $v_{neighbor}$ of $v(\mathscr{P}_{parent})$ **do**
    $\mathscr{P}_{child} = \texttt{PropagatePath}(e_{neighbor}, \mathscr{P}_{parent})$
    if not $\texttt{PrunePath}(v_{neighbor}, \mathscr{P}_{child})$ **then**
      $\texttt{AddPathToVertex}(v_{neighbor}, \mathscr{P}_{child})$
      $Q = Q \cup \mathscr{P}_{child}$
    end if
  end for

6.3 Opportunistic Look-Ahead Planning

Our method provides online decision-making for the robot by integrating the planning process with the visual SLAM formulation. Selecting a revisit action is not triggered by breaching the uncertainty threshold as in PDN (Kim and Eustice 2015). Rather, the planner runs in conjunction with SLAM and we design our framework to be opportunistic in nature, such that the selection of a revisit action at any point during the mission is based on a tradeoff between convenience and necessity. To this end, we develop a two-step optimization to decide whether to continue the exploration policy or divert to make loop-closures along a revisit path:

1. we predict the uncertainty along a *look-ahead horizon* to determine the maximum cost of a revisit path the planner should accept, and
2. we search for the candidate path with minimum cost less than this upper bound.

The look-ahead horizon is composed of the next m robot poses predicted from following the nominal exploration policy. We add the look-ahead poses as vertices in the RRG and designate each as a valid goal point, such that the planner searches for candidates that divert from, and return to, the nominal policy at any point along the horizon. (We used $m = 100$ in the hybrid simulation experiments, corresponding to a 20 m horizon. During field trials, the horizon corresponded to the length of an entire trackline.) Incorporation of the horizon gives the planner a sense of the expected measurements and predicted uncertainty, $\mathscr{U}_{\texttt{exp}}$, for exploring with the nominal policy. We query the planner for $\mathscr{U}_{\texttt{exp}}$ in order to compute the maximum acceptable cost for a candidate that diverts within the look-ahead horizon, given by

$$\mathscr{C}_{max} = \beta^{\left(\frac{\mathscr{U}_{\texttt{exp}}}{\mathscr{U}_{\texttt{upper}}} - 1\right)}, \tag{12}$$

where β is selected by the user. (We used $\beta = 1000$ in hybrid simulation and $\beta = 20$ in field trials.) This exponential function yields a higher allowable cost as the uncertainty $\mathscr{U}_{\texttt{exp}}$ increases (see Fig. 8). When the exploration uncertainty is low, the planner is restrictive when choosing paths, only selecting diversions if they are convenient and very beneficial. When the exploration uncertainty is high, the planner is willing to accept more costly revisit actions out of necessity. Instituting this upper bound on path cost is the first step in the optimization, accomplished within the function `ComputeMaxCost`().

From here, the algorithm proceeds with RRG construction and candidate path propagation to find candidates that may divert from exploration at any point within the look-ahead horizon. The second step in the optimization is the search for the candidate with minimum cost below $\mathscr{C}_{max}$, performed by the `UpdateBestPath`() function. If no revisit paths are found with cost below $\mathscr{C}_{max}$, the nominal exploration action of traveling along the look-ahead steps is selected.

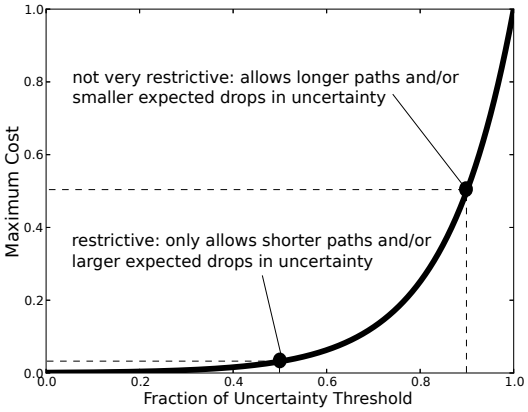


Fig. 8 The curve from (12) with $\beta = 1000$, used to determine the maximum allowable cost of a candidate revisit path during the planning process.

Whenever a plan is executed and finished, either exploring or revisiting, the algorithm begins again with a new plan initialization and look-ahead horizon.

6.4 Parameter Discussion

The proposed method includes a number of tunable parameters to provide the user with finer control and aid in transferring the system to other applications and domains. We discuss some of the more important parameters here.

The uncertainty threshold $\mathscr{U}_{\texttt{upper}}$ is a function of the robot pose covariance. In this paper, it is set to a value that illustrates sufficient performance of the system over the duration of our experiments. In the future, however, we intend for this threshold to be set according to a measure of the resulting SLAM map quality. Note, though, that setting the threshold too low can lead to losing the opportunistic nature of the algorithm since there may be no freedom in selecting continued exploration with each planning event.

The path cost tuning parameter α controls the preference for choosing paths within each planning event. A value of α near 1 means that paths with the most beneficial drops in uncertainty are preferred over paths with shorter revisit lengths. An α near 0 means that shorter diversions are preferred even if the uncertainty benefit is minimal. If the utility of the environment allows, setting α near 1 generally causes the robot to execute fewer, larger loop-closure actions that drop the uncertainty well below the threshold. Setting α near 0 causes the robot to gather many small loop-closures to minimally satisfy the uncertainty constraint.

The parameter β defines the degree to which the algorithm is opportunistic. Setting $\beta = 1$ provides a linear rela-

tionship to govern acceptable loop-closure actions with respect to the robot's exploration uncertainty and the uncertainty threshold. The opportunistic nature of the algorithm decreases as the value of β is increased, such that a very high β causes the robot to select revisit actions only when the uncertainty threshold is reached. For our experiments, we found that $\beta \in [10, 10000]$ reflected sensible opportunistic behavior. Refer to Fig. 8 for more information on β.

6.5 Comments on Implementation

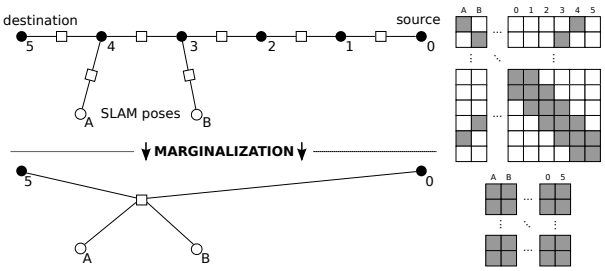
The computational complexity of the proposed algorithm can be roughly attributed to three sources: construction of the RRG, propagation of candidates throughout the RRG, and the evaluation of the cost of candidates for pruning and best path selection. The complexity of the construction phase is well-documented (Karaman and Frazzoli 2011), although we include the additional computation required to predict the measurements available along each edge. Similarly, Hollinger and Sukhatme (2014) present the complexity of propagating candidates throughout the RRG, which is exponential in the worst case but tractable for aggressive pruning strategies like the one we propose. Evaluation of the cost of a candidate using the Cholesky decomposition is generally $O(n^3)$, where n is the dimension of the information matrix, but lower with methods for sparse systems. A helpful feature of the incremental sampling-based nature of the algorithm is that it finds solutions quickly but searches to improve the best path as computation time remains.

Below we describe some implementation techniques to help reduce overall computation time, in addition to common methods like biasing the sampling toward salient regions or adjusting parameters related to RRG connectivity and pruning aggressiveness.

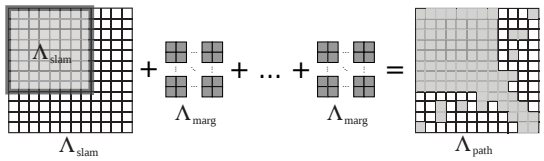
6.5.1 Edge Marginalization

When an edge is traversed during propagation, the size of the candidate's information matrix grows proportional to the number of virtual poses added by the edge. For example, an edge adding five 6-DOF virtual poses to a candidate path adds five odometry factors to the pose-graph and grows the candidate information matrix by 30 rows and columns. To alleviate this growth and leverage the information form parameterization, we can use marginalization to significantly reduce the dimensionality of the delta information of an edge during its construction.

Consider the example edge shown in Fig. 9(a); rather than explicitly representing all the factors along the edge, we condense their combined delta information ($\Lambda^i_{\texttt{edge}}$) into a single n-ary factor by marginalizing out the intermediate poses in the edge (i.e., edge poses that are not the source or destination). The single factor is represented by



(a) Edge construction with marginalization technique



(b) Calculating path posterior information using marginalized edges

Fig. 9 (a) The factor graph and delta information matrix arising from an edge before (top, $\Lambda_{\texttt{edge}}$) and after marginalization (bottom, $\Lambda_{\texttt{marg}}$) during construction. (b) The algorithm builds the posterior information incrementally from the sources of delta information arising from each edge in the path. The marginalization technique reduces the complexity of path evaluation within the algorithm.

the marginalized delta information, $\Lambda^i_{\texttt{marg}}$, which replaces $\Lambda^i_{\texttt{edge}}$ but induces the exact same information into the factor graph. The marginalized delta information is found in the usual way via the Schur complement,

$$\Lambda^i_{\texttt{marg}} = \Lambda^i_{\texttt{aa}} - \Lambda^i_{\texttt{ab}} {\Lambda^i_{\texttt{bb}}}^{-1} \Lambda^i_{\texttt{ba}}, \tag{13}$$

where a represents the source and destination poses (as well as states from target poses corresponding to expected camera registrations) and b represents intermediate poses on the edge. This process is visualized in Fig. 9(a). Then, for implementation, (9) takes the final form,

$$\Lambda_{\texttt{path}} = \Lambda_{\texttt{slam}} + \Lambda^1_{\texttt{marg}} + \ldots + \Lambda^k_{\texttt{marg}}. \tag{14}$$

As a result of this marginalization, a candidate's mean and information are augmented by only *one* new virtual pose upon traversing an edge, no matter how many virtual poses it originally contained.

6.5.2 Parallel Evaluation

In our C++ implementation of the algorithm, propagation and evaluation of a candidate path over an edge averages 33.2 ms for representative missions like those in §7 and §8. Roughly 80% of this time is due to calculating the terminating covariance via Cholesky factorization of $\Lambda_{\texttt{path}}$. We can achieve significant speed-up by parallelizing evaluations of candidates within `ProcessQueue()` by performing the necessary Cholesky factorizations across multiple threads. For

the *USCGC Escanaba* field trials presented in §8.2 that used this parallel implementation, the equivalent timing for propagation and evaluation of a candidate path was reduced to an average of 9.0 ms.

7 Hybrid Simulation Results

We tested the proposed active SLAM framework both in a hybrid simulation environment, presented here, and in real-world field trials with the HAUV, presented in the next section. The hybrid simulation incorporates real-world data collected in field trials but simulates the path traveled by the robot. In particular, the hybrid simulation presented here uses data collected during a ship hull inspection of the *SS Curtiss* by the HAUV in February 2011. More information about this experimental setup and dataset can be found in the work by Kim and Eustice (2015).

7.1 Synthetic Saliency Environment

The first simulation features real odometry and depth measurements collected by the robot but uses synthetic imagery, such that salient and non-salient portions of the environment are assigned by the user. We compare the proposed active SLAM method against three other methods: an open-loop survey with no revisit actions, a preplanned deterministic strategy with revisit actions along every trackline, and PDN (Kim and Eustice 2015). The deterministic case is simulated twice—over an optimistic salient region of the environment and over a pessimistic non-salient region. In all scenarios, the robot follows the same nominal exploration policy and the compared planning methods determine when to make diversions and which revisit paths to take.

Results from the synthetic saliency simulation are shown in Fig. 10. While the path length of the deterministic method is inherently long, its performance at bounding the uncertainty of the robot is completely dependent upon whether the preplanned revisit paths travel a salient region. As such, we see its two extremes of the spectrum: a salient case that serves as a good baseline for "best-possible" uncertainty performance, and a non-salient case that underperforms even the open-loop survey. Considering the problem formulation of operating in an *a priori* unknown environment, the selection of salient preplanned revisits and the performance of this method are left completely up to chance.

PDN identifies salient areas of the environment online and thus performs well regarding path length and uncertainty. However, it is a naïve framework that only enumerates a few candidate revisit paths and simply waits until the uncertainty threshold is breached before gathering loop-closures. In contrast, the opportunistic nature and sampling-based approach of the proposed method evaluates hundreds of candidate paths for revisiting at any point during the mission. As a result, the proposed method outperforms the deterministic and PDN methods in terms of path length and results in uncertainty levels similar to the "best-possible" deterministic case. Summarized statistics for the hybrid simulations (and all field trials) are presented in Table 1.

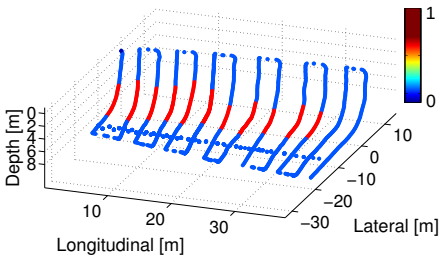
Table 1 Summarized Results

METHOD	PATH LENGTH [m]	AVG. UNCERTAINTY [% of $\mathscr{U}_{\mathtt{upper}}$]
HYBRID SIMULATION—SYNTHETIC SALIENCY		
Open-loop	386.5	121.8
DET (non-sal)	885.2	151.7
DET (sal)	876.2	33.4
PDN	583.3	71.3
Proposed	524.5	42.5
HYBRID SIMULATION—REAL IMAGERY		
Open-loop	386.5	121.8
DET	865.8	64.3
PDN	527.2	67.2
Proposed	557.3	53.7
MHL FIELD TRIAL 1		
Open-loop	124.1	111.8
Proposed	233.1	61.0
MHL FIELD TRIAL 2		
Open-loop	77.2	99.3
DET	271.3	199.3
Proposed	124.5	60.4
USCGC ESCANABA FIELD TRIAL 1		
Open-loop	109.8	97.2
DET	203.9	55.3
Proposed	214.3	56.2
USCGC ESCANABA FIELD TRIAL 2		
Open-loop	94.4	90.0
DET	163.0	67.3
Proposed	169.8	63.8

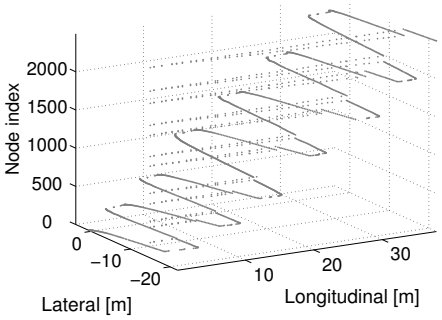
7.2 Real Image Data

Here we present results using the hybrid simulation with the real imagery collected during the *SS Curtiss* inspection. Images from the dataset densely cover the entire target environment, as shown in Fig. 2. We use the same nominal exploration policy as the previous simulation but instead use the real camera images recorded by the robot to calculate saliency scores and attempt loop-closure registrations. The proposed algorithm is again compared to the open-loop, deterministic, and PDN methods. Like the previous simulation, the deterministic strategy revisits along every trackline to a preplanned pose. Results are presented in Fig. 11.

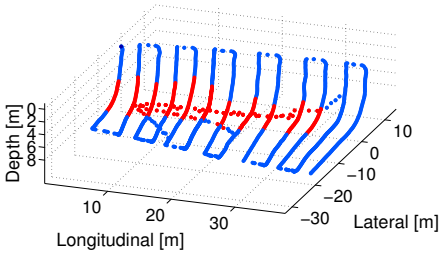
This time, the deterministic revisits travel portions of the environment that yield some camera registrations, but none that significantly drive down the uncertainty. PDN slightly outperforms the proposed method in overall path length. (Notice, though, that PDN is very close to triggering the execution of a fourth revisit action at the end of the mission.)



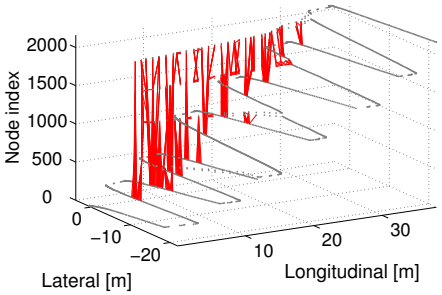
(a) Trajectory—DET (non-sal)



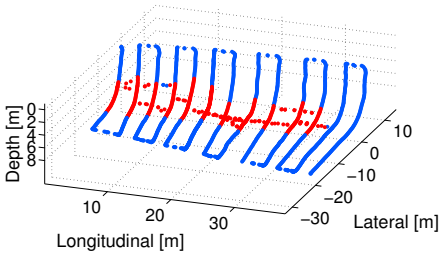
(b) Time Elevation



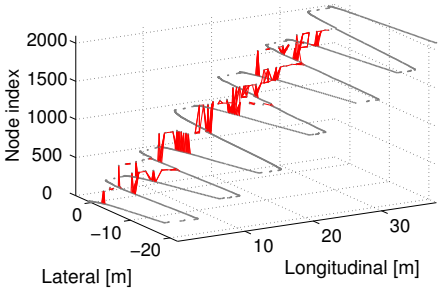
(c) Trajectory—PDN



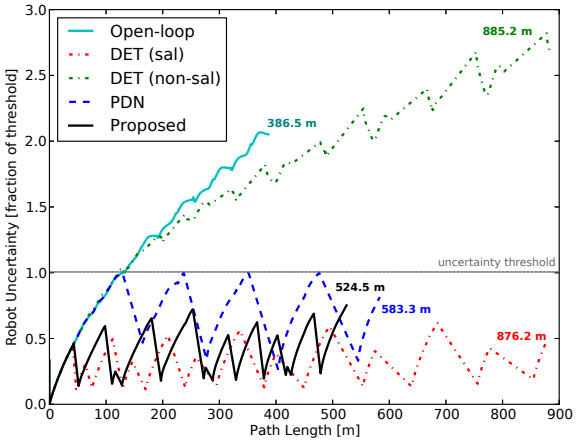
(d) Time Elevation



(e) Trajectory—Proposed

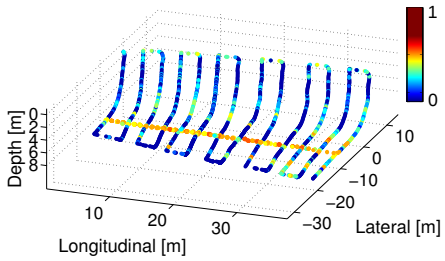


(f) Time Elevation

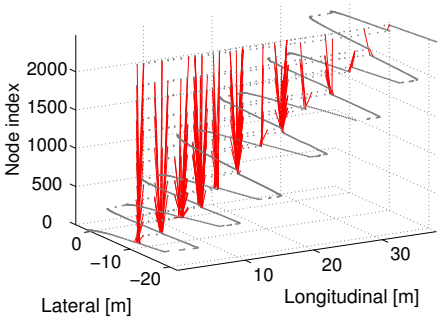


(g) Uncertainty vs. Path Length

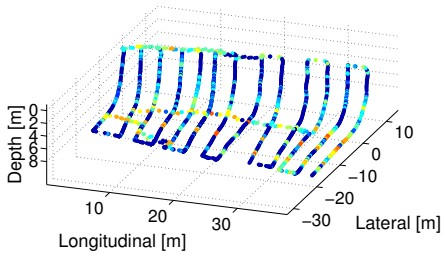
Fig. 10 Hybrid simulation results with the synthetic saliency environment. The non-salient deterministic strategy (a) (b) results in no loop-closures and underperforms even the open-loop survey. Both PDN (c) (d) and the proposed method (e) (f) constrain the navigation uncertainty with loop-closures throughout the mission. The uncertainty versus path length plots are shown in (g). SLAM poses in the trajectory plots are color-coded by their visual saliency, from red (salient) to blue (non-salient). Visual loop-closures are represented by red links on the time elevation plots, where the z-axis is scaled by time. Larger links close greater loops with respect to time elapsed.



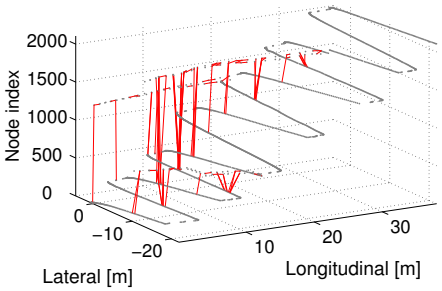
(a) Trajectory—DET



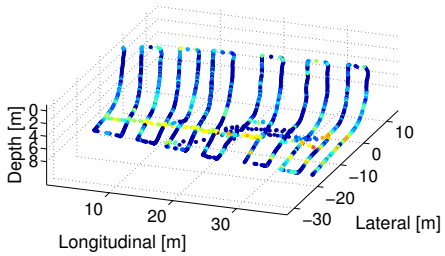
(b) Time Elevation



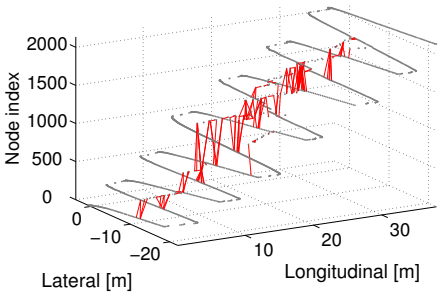
(c) Trajectory—PDN



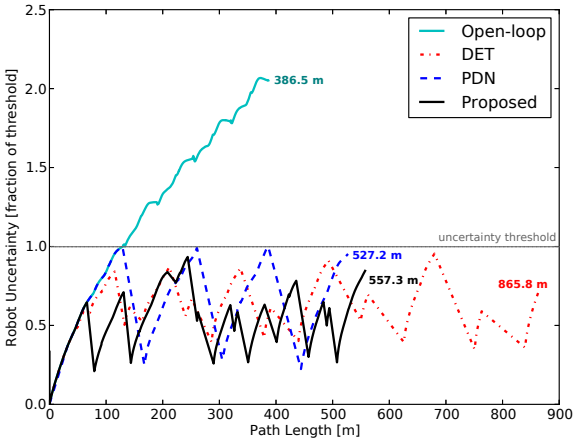
(d) Time Elevation



(e) Trajectory—Proposed

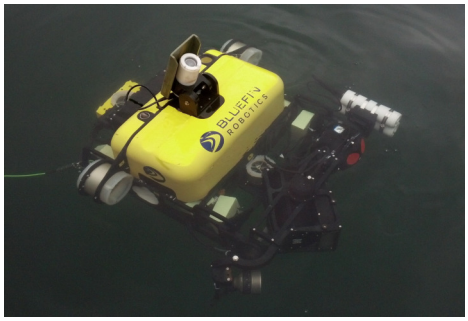


(f) Time Elevation



(g) Uncertainty vs. Path Length

Fig. 11 Hybrid simulation results using real imagery collected by the HAUV while surveying the *SS Curtiss*. The deterministic strategy (a) (b) results in an unnecessarily long path length. PDN (c) (d) results in the shortest path length but the proposed method (e) (f) yields the lowest average uncertainty for the mission. The uncertainty versus path length plots are shown in (g). Visual loop-closures are represented by red links on the time elevation plots, where the z-axis is scaled by time. Larger links close greater loops with respect to time elapsed.



(a) HAUV

(b) Physical modeling basin at MHL

(c) *USCGC Escanaba*

Fig. 12 (a) The HAUV, developed by Bluefin Robotics, in the water before field testing. (b) The HAUV performing a seafloor survey in the physical modeling basin at the University of Michigan's Marine Hydrodynamics Laboratory. (c) The *USCGC Escanaba*, used for hull-relative inspection field trials. Image from U.S. Coast Guard at coastguard.dodlive.mil.

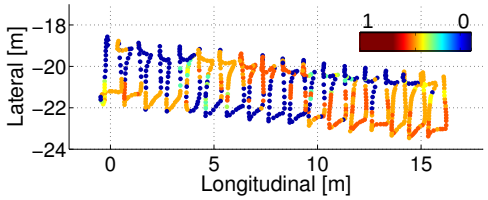
Table 2 Parameters for Field Trials

	MHL	ESCANABA 1	ESCANABA 2
α	0.9	0.9	0.75
β	20	20	20
$\mathscr{U}_{\text{upper}}$ [m-rad]	4×10^{-5}	6×10^{-5}	6×10^{-5}
d_{upper} [m]	40	100	100

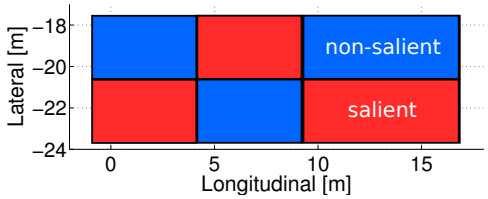
Still, the proposed method results in an average uncertainty along the exploration trajectory that is 20% lower than the PDN result with less than 6% increase in path length.

8 Real-World Field Trials

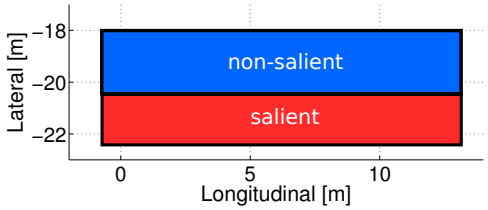
In addition to the hybrid simulations, we tested the proposed framework in real-world field trials with the HAUV. We applied the framework to two different types of inspections with the robot: seafloor surveys of a long, narrow basin, and hull-relative surveys of a US Coast Guard cutter. Details of these field trials are presented below and results are summarized in Table 1. Table 2 reports the algorithm parameters used in field testing. All parameters were selected to illustrate sufficient operation of the proposed method in the target environment.



(a) Trajectory—Open-loop



(b) Saliency for First Trials



(c) Saliency for Second Trials

Fig. 13 The experimental setup for the field trials at MHL. (a) The open-loop trajectory resulting from following the nominal exploration policy. (b) The environment saliency for the first set of trials. (c) The environment saliency for the second set of trials.

8.1 MHL Seafloor Surveys

We used the HAUV and the proposed framework to survey the floor of the physical modeling basin at the Marine Hydrodynamics Laboratory (MHL) at the University of Michigan, seen in Fig. 12(b), in May 2014. The physical modeling basin is over 100 m long, 6 m wide, and 3 m deep. For seafloor surveys, the HAUV is configured with both the DVL and underwater camera facing downward and a nominal exploration policy similar to those for hull-relative surveys—to cover the target area in an open-loop fashion with back and forth tracklines. We performed these surveys with the HAUV operating just below the waterline and limited the target area to a (roughly) 15 m section of the basin.

Since the basin is a very salient environment, images acquired almost anywhere are useful for making camera registrations. To increase the difficulty of making loop-closure registrations and obtain scenarios that better illustrate the applicability of our approach, we blurred the imagery (with a Gaussian blur) in some portions of the basin to degrade its quality. Blurring occurred in specific areas of the basin according to an independent estimate of the robot pose from an

AprilTag fiducial marker (Olson 2011) observed through the HAUV's periscope camera. By controlling the image quality, we uniquely tailored the saliency of the environment for two sets of field trials.

For the first set of trials at MHL, we blurred the imagery in the environment in offsetting rectangular regions. This resulted in alternating salient and non-salient areas of the basin as seen in Fig. 13(b). We tested the proposed framework against the open-loop nominal policy in this set of trials, with results shown in the left column of Fig. 14. Some small cross-track loop-closures are made during the open-loop survey, but none that prevent the navigation uncertainty from growing unbounded. The planning algorithm within the proposed framework efficiently bounds the uncertainty by selecting revisit paths that weave through the salient patches of the environment and avoid non-salient patches. The proposed method does result in an overall path length nearly twice as long as the open-loop case, however.

The environment for the second set of field trials at MHL consisted of a single salient band running lengthwise along the edge of the basin, shown in Fig. 13(c). For this set of trials, we compared the proposed framework against the open-loop policy and a deterministic strategy with preplanned revisit actions along every other trackline. Once again, the preplanned actions travel a non-salient portion of the environment and the deterministic strategy performs poorly, with large uncertainty growth and extremely long path length. The time elevation graph of Fig. 14(b) shows zero large loop-closing camera registrations. Contrastingly, the resulting trajectory from the proposed framework shows loop-closures spread uniformly throughout the mission, as revisit actions are performed within the salient portion of the environment. Compared to the deterministic case, the proposed method surveyed the target area with a 54% shorter path length and 70% lower average navigation uncertainty. Even if the deterministic strategy happened to travel the salient portion of the basin, the path length would be equivalent to the deterministic result shown. Refer to Table 1 for a summary of results.

8.2 *USCGC Escanaba* Hull-Relative Surveys

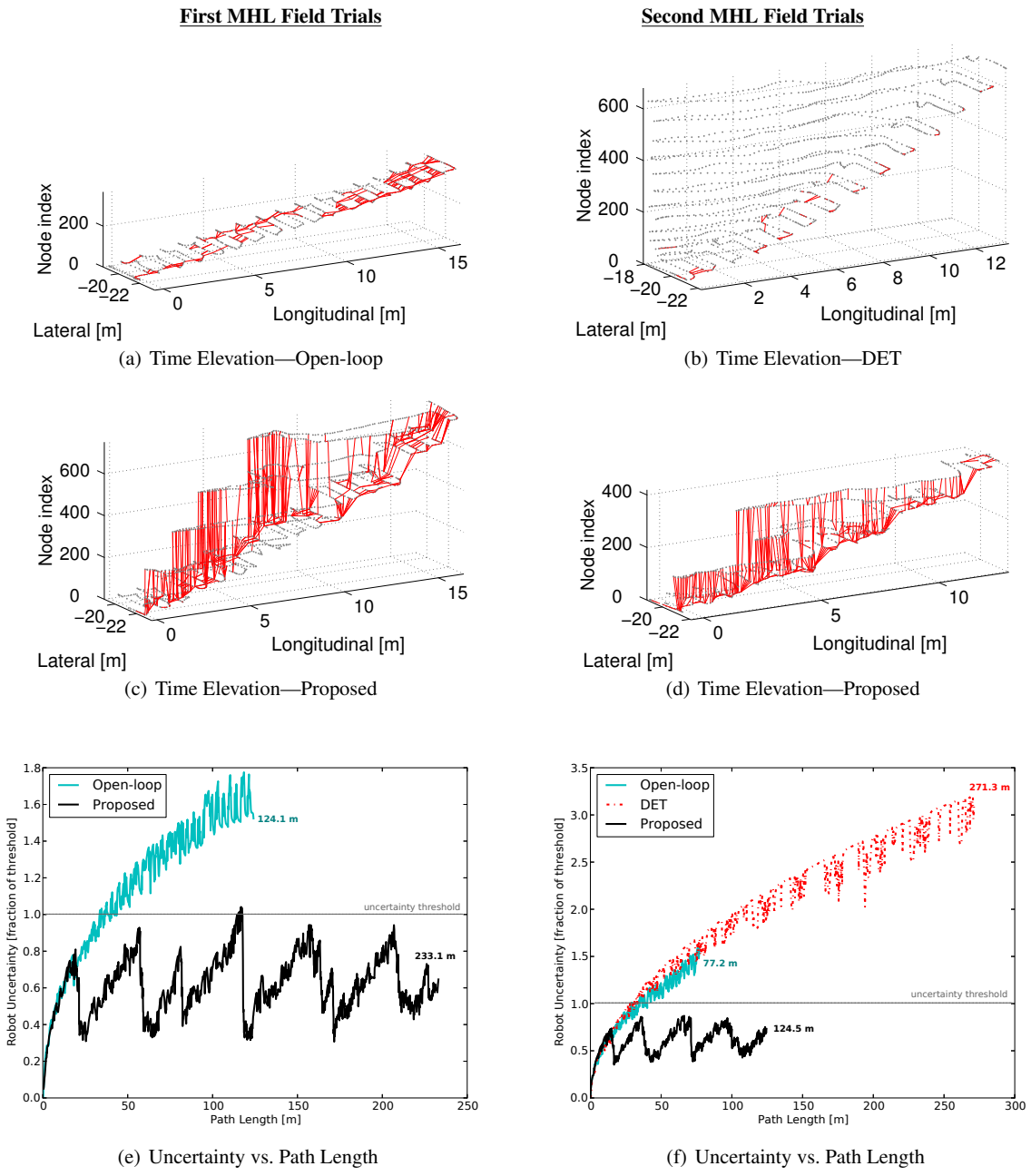
The proposed active SLAM framework was also tested while inspecting the hull of the *USCGC Escanaba* in Boston, MA in October 2014. The *USCGC Escanaba* (WMEC-907) is 82 m in length with a beam of 12 m and draft of 4.4 m. For hull-relative missions, the HAUV travels along the underwater portion of the ship hull and actuates the DVL and underwater camera such that they always point normal to the hull surface. Two sets of experiments were performed on a section of the hull situated between the stabilizer fin and bow of the ship. Each set of trials included surveys using the open-loop, deterministic, and proposed active SLAM strategies.

Results from the first set of trials on the *USCGC Escanaba* are given in Fig. 15. The preplanned deterministic strategy revisits a waypoint placed on the first trackline of the mission through a salient patch along the hull. From the time elevation graph in Fig. 15(d), it can be seen that the deterministic case relies on large loop-closures near the beginning of the mission to bound the navigation uncertainty. The proposed method yields a trajectory with loop-closures distributed throughout the salient portions of the environment. Both strategies, however, produce favorable end results—nearly identical average uncertainties well below the acceptable threshold and path lengths within 11 m of each other.

The second set of field trials on the *USCGC Escanaba* were performed at a time of day when variable sunlight penetrating the water affected the imagery collected by the HAUV, specifically relating to the salient patch located about 4 m deep. This variability is evidenced by the trajectories and time elevation plots shown in Fig. 16 compared with those in Fig. 15. Still, the deterministic and proposed methods perform favorably with respect to uncertainty and path length, shown in Fig. 16(j). In this set of trials, the proposed method selects revisit actions solely along the deepest portion of the environment, where the imagery is less affected by sunlight but still possesses good loop-closure utility. Here, the proposed method results in a slightly lower average uncertainty than the deterministic method.

8.2.1 Performance of Planning Events

Here, we further investigate the planning algorithm within the proposed framework. For the *USCGC Escanaba* field trials, we report detailed performance statistics for each planning event in Table 3. The second and third columns of this table report the number of candidate plans propagated (the number of calls to `PropagatePath()` in Algorithm 1) and the number of candidates processed, or popped off the queue (calls to `Pop`(Q) in Algorithm 1), respectively. Timing statistics are reported in the fourth and fifth columns. The fourth column displays the total time spent planning for an event (triggered to exit after 20 s, upon completion of the current iteration). In the fifth column, the timing results for best path selection show that the sampling-based algorithm often returns a viable solution quickly, and can use any remaining time to search for improvements. The average time to find the selected best path is 2.992 s for these events. Other statistics reported include the resulting action and the predicted uncertainty of the selected best path. Regardless of the resulting decision—following the nominal exploration policy or diverting from this policy to gather loop-closure registrations—the planning process relies upon



(a) Time Elevation—Open-loop

(b) Time Elevation—DET

(c) Time Elevation—Proposed

(d) Time Elevation—Proposed

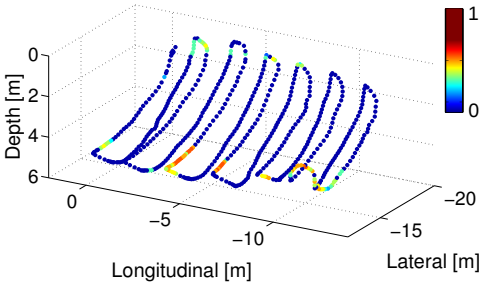
(e) Uncertainty vs. Path Length

(f) Uncertainty vs. Path Length

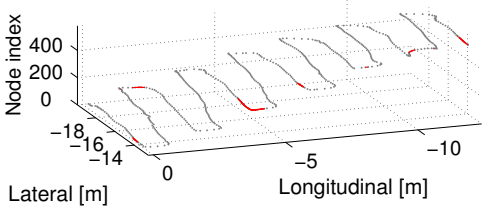
Fig. 14 Results from field trials performed in the MHL physical modeling basin at the University of Michigan. Results pertaining to the first set of trials are shown in the left column and the second set of trials in the right column. In the first set of trials, small cross-track camera registrations are made during the open-loop survey (a), but none contribute to a large reduction in uncertainty. The proposed method (c) produces revisit paths that weave through the salient patches in the environment. In the second set of trials, the deterministic strategy results in zero large loop-closures as the preplanned revisit paths travel the non-salient portion of the environment. The proposed framework (d) results in visual loop-closures spread throughout the mission. The uncertainty versus path length plots are shown in (e) and (f). Visual loop-closures are represented by red links on the time elevation plots, where the z-axis is scaled by time. Larger links close greater loops with respect to time elapsed.

accurate prediction of the information expected along candidate paths.

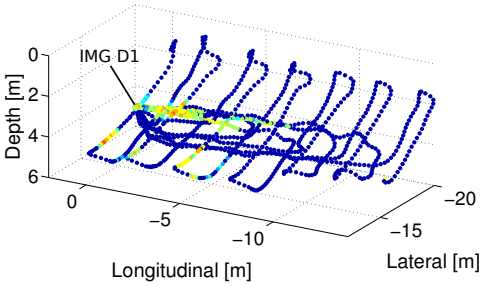
Fig. 17 displays the results of the predictions from each plan for the hull-relative field trials. The predicted path length and uncertainty of the selected best path from each planning event is overlaid on the actual data recorded. It can be seen that the planner accurately predicts the performance of the actual robot during the survey. These results support the formulation for evaluating path utility presented in §4 and §5. The statistics validate the method of using GP regression for saliency prediction and scaling camera information in (6) to accurately capture the stochasticity of



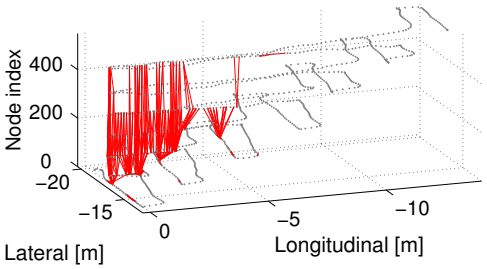
(a) Trajectory—Open-loop



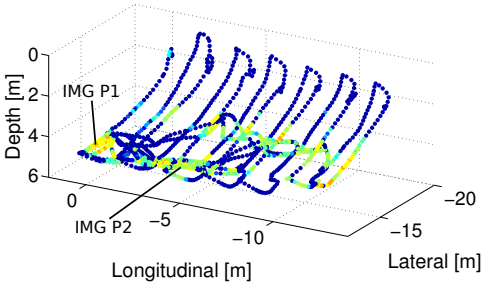
(b) Time Elevation



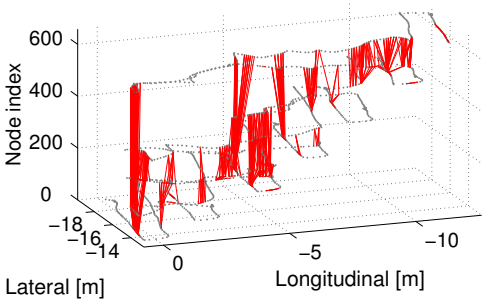
(c) Trajectory—DET



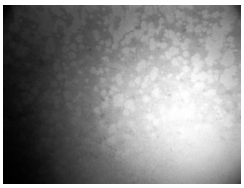
(d) Time Elevation



(e) Trajectory—Proposed

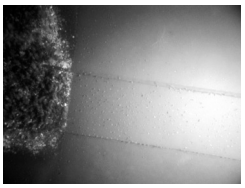


(f) Time Elevation

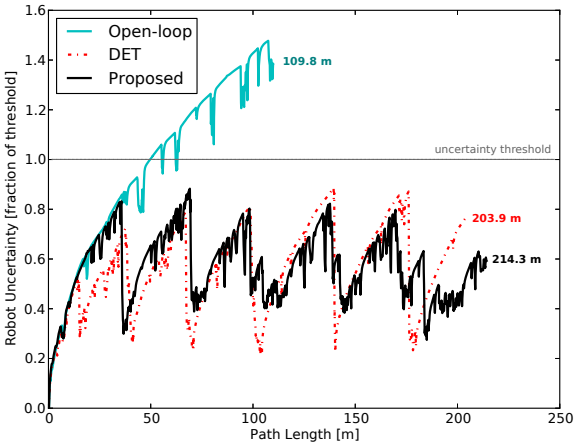


(g) IMG D1

(h) IMG P1

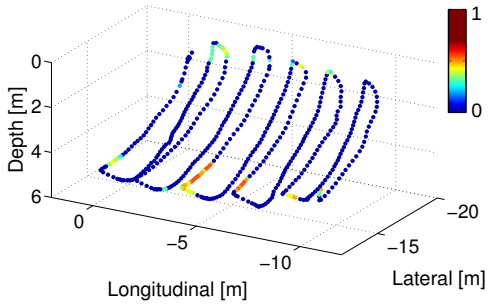


(i) IMG P2

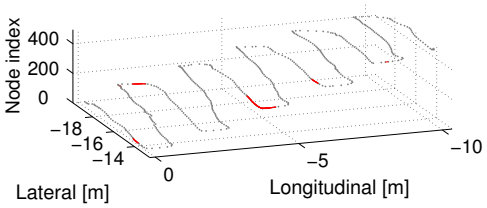


(j) Uncertainty vs. Path Length

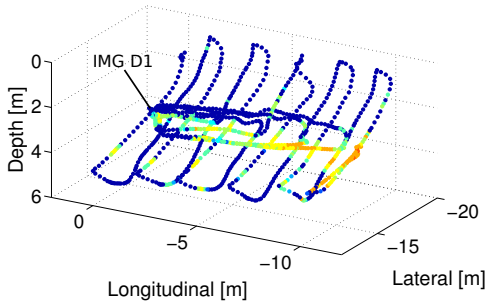
Fig. 15 Results from the first set of field trials performed while surveying the *USCGC Escanaba* in Boston, MA. The open-loop survey is shown in (a) and (b). The deterministic strategy (c) (d) relies on large loop-closures near the beginning of the mission to bound the uncertainty. The proposed method (e) (f) yields a trajectory with loop-closures throughout the salient portions of the environment. Sample images from the deterministic and proposed strategies are displayed in (g), (h), and (i). The uncertainty versus path length plots are shown in (j). Visual loop-closures are represented by red links on the time elevation plots, where the z-axis is scaled by time. Larger links close greater loops with respect to time elapsed.



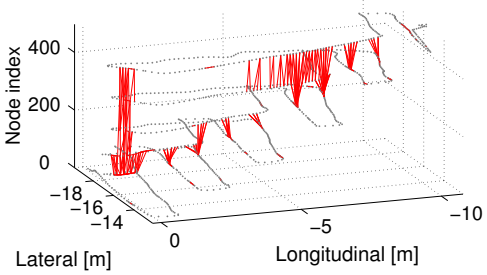
(a) Trajectory—Open-loop



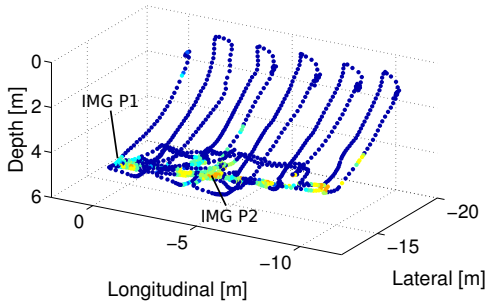
(b) Time Elevation



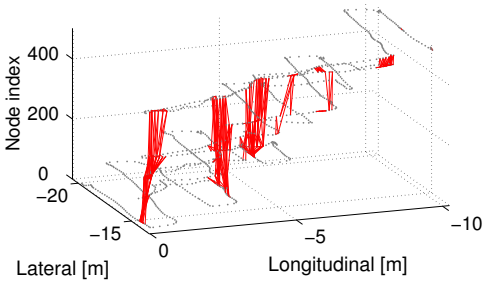
(c) Trajectory—DET



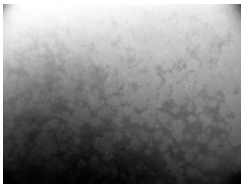
(d) Time Elevation



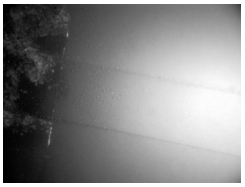
(e) Trajectory—Proposed



(f) Time Elevation

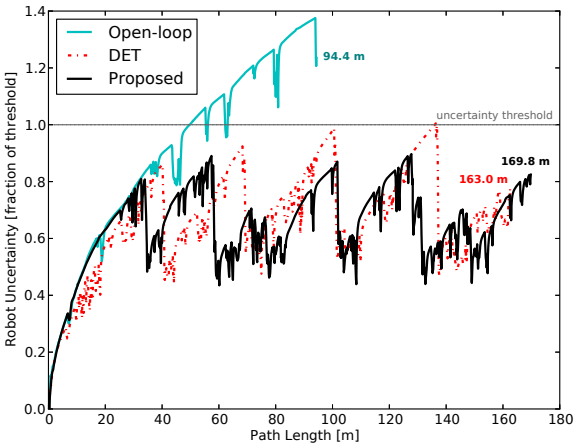


(g) IMG D1



(h) IMG P1

(i) IMG P2



(j) Uncertainty vs. Path Length

Fig. 16 Results from the second set of field trials performed while surveying the *USCGC Escanaba* in Boston, MA. The open-loop survey is shown in (a) and (b). The deterministic strategy (c) (d) still makes loop-closures despite imagery affected by variable sunlight during the mission. The proposed method (e) (f) selects revisit paths at the deepest part of the environment where sunlight has less effect on the imagery. Sample images from the deterministic and proposed strategies are displayed in (g), (h), and (i). The uncertainty versus path length plots are shown in (j). Visual loop-closures are represented by red links on the time elevation plots, where the z-axis is scaled by time. Larger links close greater loops with respect to time elapsed.

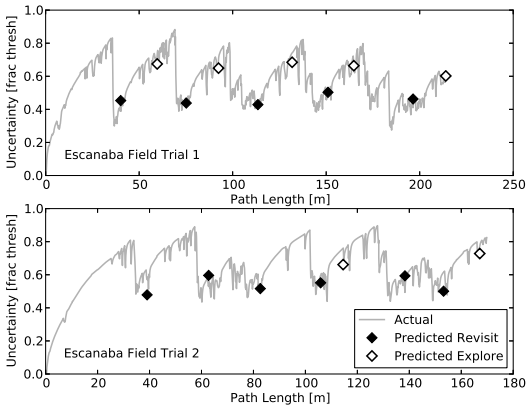


Fig. 17 The actual robot uncertainty and path length versus the robot uncertainty and path length as predicted by the active SLAM algorithm at the time of planning, for the field trials on the *USCGC Escanaba*. The predicted outcomes for each planning event align very well with the actual data collected by the robot. Black diamonds represent selected revisit actions and white diamonds represent selecting continued exploration.

achieving registrations (when the number of link proposals is large).

9 Conclusion

In this paper, we proposed a comprehensive active SLAM framework for underwater inspections. We presented a path planning algorithm integrated with visual SLAM in order to find and execute loop-closure revisit actions for a robot exploring an *a priori* unknown underwater environment subject to an acceptable uncertainty threshold. We combined a sampling-based planning approach for efficiently exploring the configuration space with an EIF framework for tracking the utility of candidate revisit paths. In addition, we included the use of GP regression for predicting environment saliency in unmapped areas and probabilistically handling expected camera registrations along a path. Finally, we developed an opportunistic approach for selecting the best revisit paths that allows the robot to autonomously execute useful loop-closure actions at any point during the mission, while still exploring the target area in efficient time. The proposed method was demonstrated using a hybrid simulation with both synthetic and real camera imagery, and using real-world field trials of a robot performing seafloor and hull-relative inpections. The results showed that, on the whole, our framework offers many benefits over other representative active SLAM methods: principled and accurate saliency prediction, the ability to search and evaluate hundreds of candidate paths, opportunistic path selection, and consistent good performance over our experimental trials. This type of active SLAM system is useful to accomplish robust, long-term autonomy in marine environments.

Acknowledgments

This work was supported by the Office of Naval Research under award N00014-12-1-0092, monitored by Dr. T. Swean and T. Kick. We would like to thank J. Vaganay from Bluefin Robotics and P. Ozog for their support during testing. S. Chaves was supported by The SMART Scholarship for Service Program by the Department of Defense.

References

Bailey T, Durrant-Whyte H (2006) Simultaneous localization and mapping (SLAM): Part II. Robotics Automation Magazine, IEEE 13(3):108–117

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

Barkby S, Williams SB, Pizarro O, Jakuba MV (2011) Bathymetric SLAM with no map overlap using Gaussian processes. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, pp 1242–1248

van den Berg J, Patil S, Alterovitz R (2012) Motion planning under uncertainty using iterative local optimization in belief space. International Journal of Robotics Research 31(11):1263–1278

Bourgault F, Makarenko AA, Williams SB, Grocholsky B, Durrant-Whyte HF (2002) Information based adaptive robotic exploration. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Lausanne, Switzerland, pp 540–545

Bry A, Roy N (2011) Rapidly-exploring random belief trees for motion planning under uncertainty. In: Proceedings of the IEEE International Conference on Robotics and Automation, Shanghai, China, pp 723–730

Chaves SM, Kim A, Eustice RM (2014) Opportunistic sampling-based planning for active visual SLAM. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, pp 3073–3080

Durrant-Whyte H, Bailey T (2006) Simultaneous localization and mapping: Part I. IEEE Robotics and Automation Magazine 13(2):99–110

Eustice RM, Singh H, Leonard JJ (2006a) Exactly sparse delayed-state filters for view-based SLAM. IEEE Transactions on Robotics 22(6):1100–1114

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

Feder HJS, Leonard JJ, Smith CM (1999) Adaptive mobile robot navigation and mapping. International Journal of Robotics Research 18(7):650–668

Galceran E, Nagappa S, Carreras M, Ridao P, Palomer A (2013) Uncertainty-driven survey path planning for bathymetric mapping. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, pp 6006–6012

Gonzalez-Banos HH, Latombe JC (2002) Navigation strategies for exploring indoor environments. International Journal of Robotics Research 21(10–11):829–848

Hand DJ (2008) Statistics: A Very Short Introduction. Oxford University Press, Oxford

Hollinger GA, Sukhatme GS (2014) Sampling-based robotic information gathering algorithms. International Journal of Robotics Research 33(9):1271–1287

Table 3 *USCGC Escanaba* Planning Events

PLAN NO.	NO. CANDIDATES		TIME ELAPSED [s]		ACTION SELECTED	PREDICTED UNCERTAINTY [% of $\mathscr{U}_{\mathtt{upper}}$]
	Propagated	Processed	Total	Best Path		
USCGC ESCANABA FIELD TRIAL 1						
1	2524	862	4.8	0.344	Revisit	45.3
2	3528	1211	10.3	0.144	Explore	67.5
3	3445	1247	13.2	0.449	Revisit	43.8
4	2856	1238	23.0	0.262	Explore	64.9
5	3452	1185	21.3	12.526	Revisit	42.9
6	2442	991	20.6	0.270	Explore	68.5
7	2100	944	20.1	12.909	Revisit	50.3
8	1724	726	23.3	0.302	Explore	66.4
9	1866	694	20.6	0.396	Revisit	46.3
10	976	459	20.5	0.576	Explore	60.2
USCGC ESCANABA FIELD TRIAL 2						
1	1602	661	2.7	1.206	Revisit	47.8
2	3074	945	8.9	6.960	Revisit	59.6
3	3442	1244	21.6	8.054	Revisit	51.6
4	2233	963	19.2	0.540	Revisit	55.1
5	1729	749	20.1	0.286	Explore	66.1
6	2377	973	23.2	4.884	Revisit	59.3
7	1031	503	20.4	3.414	Revisit	50.0
8	1361	695	20.3	0.337	Explore	72.8

Hover FS, Eustice RM, Kim A, Englot B, Johannsson H, Kaess M, Leonard JJ (2012) Advanced perception, navigation and planning for autonomous in-water ship hull inspection. International Journal of Robotics Research 31(12):1445–1464

Indelman V, Carlone L, Dellaert F (2015) Planning in the continuous domain: A generalized belief space approach for autonomous navigation in unknown environments. International Journal of Robotics Research 34(7):849–882

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

Kaess M, Johannsson H, Rosen D, Carlevaris-Bianco N, Leonard J (2013) Open source implementation of iSAM. http://people.csail.mit.edu/kaess/isam

Karaman S, Frazzoli E (2011) Sampling-based algorithms for optimal motion planning. International Journal of Robotics Research 30(7):846–894

Kavraki LE, Svestka P, Latombe JC, Overmars M (1996) Probabilistic roadmaps for path planning in high dimensional configuration spaces. IEEE Transactions on Robotics and Automation 12(4):566–580

Kim A, Eustice RM (2013) Real-time visual SLAM for autonomous underwater hull inspection using visual saliency. IEEE Transactions on Robotics 29(3):719–733

Kim A, Eustice RM (2015) Active visual SLAM for robotic area coverage: Theory and experiment. International Journal of Robotics Research 34(4–5):457–475

LaValle SM, Kuffner JJ (2001) Randomized kinodynamic planning. International Journal of Robotics Research 20(5):378–400

Melkumyan A, Ramos F (2009) A sparse covariance function for exact Gaussian process inference in large datasets. In: Proceedings of the International Joint Conference on Artifical Intelligence, Pasadena, CA, pp 1936–1942

O'Callaghan ST, Ramos FT (2012) Gaussian process occupancy maps. International Journal of Robotics Research 31(1):42–62

Olson E (2011) AprilTag: A robust and flexible visual fiducial system. In: Proceedings of the IEEE International Conference on Robotics and Automation, IEEE, Shanghai, China, pp 3400–3407

Ozog P, Carlevaris-Bianco N, Kim A, Eustice RM (2015) Long-term mapping techniques for ship hull inspection and surveillance using an autonomous underwater vehicle. Journal of Field Robotics, Special Issue on Safety, Security and Rescue Robotics In Press

Patil S, Kahn G, Laskey M, Schulman J, Goldberg K, Abbeel P (2014) Scaling up gaussian belief space planning through covariance-free trajectory optimization and automatic differentiation. In: Proceedings of the International Workshop on the Algorithmic Foundations of Robotics, Istanbul, Turkey

Prentice S, Roy N (2009) The belief roadmap: Efficient planning in belief space by factoring the covariance. International Journal of Robotics Research 28(11-12):1448–1465

Rasmussen CE, Williams CKI (2006) Gaussian Processes for Machine Learning. The MIT Press, Cambridge, MA

Sim R, Roy N (2005) Global A-optimal robot exploration in SLAM. In: Proceedings of the IEEE International Conference on Robotics and Automation, Barcelona, Spain, pp 661–666

Smith R, Self M, Cheeseman P (1990) Estimating uncertain spatial relationships in robotics. In: Cox I, Wilfong G (eds) Autonomous Robot Vehicles, Springer-Verlag, pp 167–193

Stachniss C, Grisetti G, Burgard W (2005) Information gain-based exploration using Rao-Blackwellized particle filters. In: Proceedings of the Robotics: Science & Systems Conference, Cambridge, MA, USA

Valencia R, Andrade-Cetto J, Porta J (2011) Path planning in belief space with pose SLAM. In: Proceedings of the IEEE International Conference on Robotics and Automation, Shanghai, China, pp 78–83

Valencia R, Miro J, Dissanayake G, Andrade-Cetto J (2012) Active pose SLAM. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura, Portugal, pp 1885–1891

Valencia R, Morta M, Andrade-Cetto J, Porta J (2013) Planning reliable paths with pose SLAM. IEEE Transactions on Robotics 29(4):1050–1059

Whaite P, Ferrie F (1997) Autonomous exploration: Driven by uncertainty. IEEE Transactions on Pattern Analysis and Machine Intelligence 19(3):193–205