Active Visual SLAM for Robotic Area Coverage: Theory and Experiment

Ayoung Kim and Ryan M. Eustice

Abstract

This paper reports on an integrated navigation algorithm for the visual simultaneous localization and mapping (SLAM) robotic area coverage problem. In the robotic area coverage problem, the goal is to explore and map a given target area within a reasonable amount of time. This goal necessitates the use of minimally redundant overlap trajectories for coverage efficiency; however, visual SLAM's navigation estimate will inevitably drift over time in the absence of loop-closures. Therefore, efficient area coverage and good SLAM navigation performance represent competing objectives. To solve this decision-making problem, we introduce perception-driven navigation, an integrated navigation algorithm that automatically balances between exploration and revisitation using a reward framework. This framework accounts for SLAM localization uncertainty, area coverage performance, and the identification of good candidate regions in the environment for visual perception. Results are shown for both a hybrid simulation and real-world demonstration of a visual SLAM system for autonomous underwater ship hull inspection.

1 Introduction

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

In particular, this paper considers the task of area coverage (i.e., to cover a certain area of interest) under the constraint of bounded navigation error. Specifically, our area coverage objective seeks a balanced control between exploration and revisiting in order to achieve better SLAM performance. Without loop-closure, SLAM will inevitably accumulate navigation drift over time; thus, we need to revisit portions of the map to bound error growth. At the same time we need to pursue exploration, which is a competing objective that requires mapping the entire area within a reasonable amount of time. Furthermore, and more importantly, SLAM, path planning, and control are interwoven and, thus, inseparable problems. For example, imprecise SLAM results affect the accuracy of the area coverage and, thus, the planning accuracy, while a misplanned trajectory deteriorates SLAM's performance. In this paper, we introduce the idea of perception-driven navigation (PDN), a mathematical framework that seeks to balance the competing objectives between SLAM, control, and exploration for the autonomous robotic area coverage problem.

2 Background

We first briefly summarize studies from topic areas of SLAM, path planning, exploration and their integrated approaches in relation to PDN.

Draft manuscript, September 25, 2014.

This paper was presented in part at the IEEE International Conference on Robotics and Automation, Karlsruhe, Germany 2013 (Kim and Eustice, 2013a) and the IEEE/RSJ International Conference on Intelligent Robots and Systems Workshop on Active Semantic Perception, Vilamoura, Portugal 2012 (Kim and Eustice, 2012).

A. Kim was with the Department of Mechanical Engineering, University of Michigan, Ann Arbor, Michigan, USA during the tenure of this work. She is now with the Department of Civil and Environmental Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea ayoungk@kaist.ac.kr.

R. Eustice is with the Department of Naval Architecture & Marine Engineering, University of Michigan, Ann Arbor, Michigan, USA eustice@umich.edu.

2.1 Related Work

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

Belief Space Path Planning: There have been some efforts on merging SLAM and path planning into an integrated framework. The major difficulty in coupling path planning and SLAM is that path planning assumes that a map is known a priori while SLAM assumes that a path is given. Even stochastic path planning algorithms (LaValle and Kuffner, 1999; Kavraki et al., 1996; Kaelbling et al., 1995) start with the assumption of a known map. They focus on how to sample nodes from the area and then plan a path over them. Recently, some approaches have evaluated the resulting uncertainty in optimizing the path, such as the work of Prentice and Roy (2009) in Belief Roadmaps (BRMs), Valencia et al. (2011) on planning paths on SLAM constrained maps, or Levine (2010) on calculating possible information gain on a planned path using Rapidly-exploring Random Trees (RRTs). Of these, the BRM approach is closest to our own in that it considers the state's uncertainty when it computes the objective function, though it is different from us in that the nodes are sampled from a map that is learned offline. In the work of Valencia et al., the authors perform SLAM first, then use the resulting pose-graph to plan a path to a goal position considering information gain through the graph.

Importantly, exploration was not considered in these previous studies. Their optimality was evaluated only by the uncertainty of the robot and the map, not by time nor area. This is because exploration was excluded in their evaluation, where the main focus was instead on point-to-point path planning. Our approach broadens the optimality definition to take into account area coverage together with SLAM performance by including the area coverage rate in the cost function itself (since our focus is on area covering navigation).

Active SLAM: The SLAM community has also made some efforts to add exploration functionality within SLAM, termed "active SLAM". Stemming from the seminal work of *active perception* by Bajcsy (1988), which pointed out that control can improve the quality of sensor data, these studies assume that a default exploration policy exists and then undertake some variations for improvements. Active SLAM builds upon that notion and seeks to find the optimal trajectory that can improve both map building and localization performance.

Most active SLAM research consists of two parts: *(i)* defining a metric to be used as a measure of information gain and *(ii)* optimizing this measure to find control policies that maximize information gain. This line of research is found in the work of Feder et al. (1999), who used Fisher information (FI) as a metric in the objective function to construct an adaptive control action. More recently, Sim (2005) and Sim and Roy (2005) used FI to improve exploration, reporting the need to consider the path in localization and mapping. Their work pointed out the instability of the update step, which has been further extended to account for the control action by Davison et al. (2007). In this work, the authors considered a discrete set of actions to reduce state uncertainty. Similarly, in Bryson and Sukkarieh (2005), simulation results for unmanned aerial vehicles using a similar approach for on-line path planning was presented. Their work determined the proper action and strategies to improve the overall map quality on the basis of mutual information (MI). Although this approach established a basis for combining the control architecture with SLAM, it only sought the optimal control input and did not globally solve for path planning. Frintrop and Jensfelt (2008) presented an active gaze control algorithm for SLAM by defining the usefulness of a landmark and tracking these useful landmarks.

Aside from the active SLAM research, an optimal control strategy-based approach was introduced by Huang et al. (2005), where the authors optimized over the uncertainty of the last pose within a finite time window using a variant of model predictive control. In their work, they pointed out that the computational cost increases exponentially with the number of landmarks, the size of the map, and the size of the time window. In all of these prior studies, optimality has been defined very similarly to the belief space path planning in that it has considered only the localization uncertainty in the cost function and not area coverage.

Integrated Navigation: Toward addressing the problem of perception-driven navigation within a fully in-

tegrated approach, some studies have focused on solving for exploration strategies that simultaneously consider both navigation and exploration performance. Although not directly addressing the SLAM problem, Whaite and Ferrie (1997) considered the reduction in model uncertainty through motion as a way to explore. Gonzalez-Banos and Latombe (2002) proposed exploration strategies analogous to the next-best-view (NBV) problem in computer vision. First discussed by Connolly (1985), NBV seeks to find the best view of the scene that reveals the model details, and thus can be considered similar to the active exploration problem.

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

2.2 Our Approach

An attempt at categorizing the assumptions, objective functions and computational costs of previous studies are presented in Table 1. As seen, there is a gap between the areas of SLAM, exploration, and path planning, which is mainly due to the fact that each approach makes different assumptions on what priors are available. Active SLAM only focuses on the robot uncertainty from SLAM, and area coverage planning only solves for the optimal coverage plan without accounting for the actual SLAM performance. Unlike these studies, which are constrained to only one aspect of the navigation problem, our work pursues a balanced strategy for both SLAM and area coverage in an integrated framework. Integrating SLAM and planning is also a focus of belief space planning, however, our approach solves for an area coverage problem and differs from the belief space planning in this regard, since belief space planning is typically only point-to-point. While integrated active exploration is most similar to our own approach, they impose an optimistic assumption of obtaining all measurements predicted in the evaluation phase. This assumption fails when perceptually-derived measurements are not uniformly available throughout the environment, which is the case in our underwater visual SLAM application, for example. Specifically, the novelty of our work is in consideration of the visual perception measurement likelihood within an integrated framework of SLAM and planning.

3 Visual Saliency

In visual SLAM, not all images are equal. This is especially true in the underwater environment, where the spatial distribution of good visual features is not uniformly abundant. In this work, we adopt the visual saliency metrics defined earlier by Kim and Eustice (2011, 2013b), which include a *local saliency* (S_L) and *global saliency* (S_G) measure. Each measure provides a normalized image score from 0 to 1, where 1 indicates highly salient imagery and 0 indicates nonsalient imagery. These metrics are computed from a bag-of-words (BoW) representation of the imagery using a vocabulary that is built online. We refer readers to Kim and Eustice (2013b) for more details.

Fig. 1 depicts local and global saliency maps as applied to a portion of a SLAM survey taken from an underwater hull inspection mission. As shown, local saliency measures the intra-image texture richness of the scene, which is highly related to the ability to make successful pairwise keyframe registrations, while global saliency measures the inter-image rarity of a keyframe. As depicted, 'G' and 'H' represent nonsalient images, 'A' to 'D' are locally and globally salient images, and 'E' and 'F' are locally but not globally salient images. In our application, the robot measures these two saliency levels (i.e., local and global) for every keyframe it inserts into the SLAM pose-graph, and it uses them within PDN's reward calculation to determine when and where to execute loop-closure revisits.

4 Perception-Driven Navigation

Typically, SLAM is formulated as a passive process that localizes and builds a map using whatever data sequence and exploration trajectory it is provided. PDN is designed to sit one layer above SLAM in that it represents an integrated framework to evaluate rewards and execute actions to guide the robot for better SLAM navigation and area coverage performance. In this work, we have adopted the incremental smoothing and mapping

Table 1: Summary of related works. The previous studies are summarized with respect to prior knowledge, objective function, consideration of measurement likelihood, and computational cost. In the calculation of time complexity, algorithms are compared assuming an n element state vector. Area coverage planning (Hert et al., 1996; Choset, 2001; Baek et al., 2011) focuses on computing an optimal path offline where the memory complexity scales with the size of the map and the planning parameters, and the major operation and time complexity are not indicated in the table. In many studies, the EKF has been a popular choice for the SLAM back-end (Bourgault et al., 2002; Bryson and Sukkarieh, 2005; Davison et al., 2007). When the objective function includes MI-based terms, computing MI requires calculation of the covariance matrix determinant, even when using a sparse information filter (Valencia et al., 2011). In the integrated navigation studies (Bourgault et al., 2002; Stachniss et al., 2005), which are most similar to PDN, computational cost appears in two terms, one related to the SLAM inference and the other related to the action path length.

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

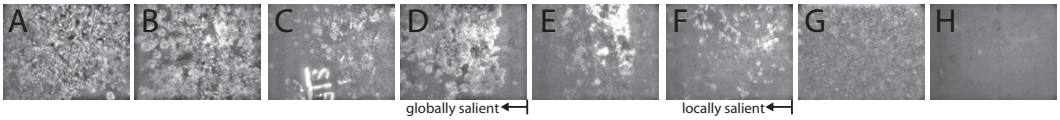
[a] n indicates state vector dimension.

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

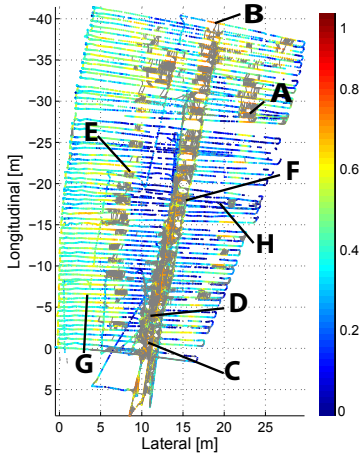
[c] From pre-sampled nodes, the algorithm considers k edges between nodes each of length l.

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

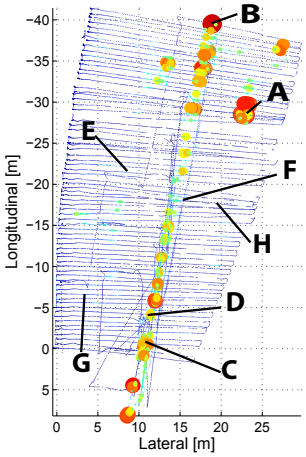
[e] $S(n)$ is the cost for covariance recovery in the reward computation.



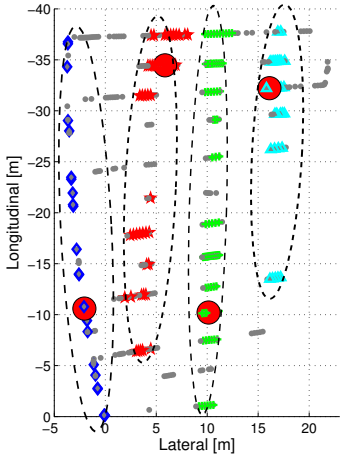
(a) Sample images



(b) Local saliency S_L



(c) Global saliency S_G



(d) Waypoint generation

Fig. 1: Depiction of local and global saliency maps for a underwater hull inspection survey of the *SS Curtiss* conducted in 2011 (Kim and Eustice, 2013b) and how they relate to PDN. Shown is a top-down view of the robot's xy trajectory. Nodes are color-coded by their saliency level, where 0 indicates nonsalient and 1 is salient. (a) 'A' to 'H' depict representative sample images from the hull—labels correspond to denoted locations. (b) Successful pairwise image links are shown as gray edges, which spatially correlate well with local saliency. (c) The node size is scaled with the global saliency level for easier visualization. (d) A depiction of the waypoint generation process. Gray dots represent a set of locally salient nodes in the graph. Each cluster is denoted by a different color and shape (red stars, blue diamonds, green crosses, magenta circles, and cyan triangles). The resulting waypoint for each cluster is shown as a red dot determined from global saliency.

(iSAM) algorithm by Kaess et al. (2008, 2010) as the SLAM back-end. In our application, constraints from odometry, monocular camera, attitude, and pressure depth are fused within iSAM (Kim and Eustice, 2013b; Hover et al., 2012).

While the normal SLAM process passively localizes itself and builds a map, PDN represents an active approach to SLAM that (*i*) clusters salient nodes into a set of candidate revisit waypoints, (*ii*) plans a point-to-point path for each candidate revisit waypoint, and (*iii*) computes a reward for revisiting each waypoint candidate. As depicted in Fig. 2, the calculated reward measures the utility of revisiting that waypoint (i.e., loop-closure) versus continuing exploration for area coverage. If the revisit action is selected to be the next best control, the robot will revisit the selected waypoint and then return to the release point to resume the nominal mission. By comparing the maximal reward for revisiting versus exploring, the robot is able to choose the next best control step.

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

1. The boundaries of the target coverage area are given.
2. The vehicle nominally executes a default exploration policy.
3. The desired target navigation uncertainty is defined by the user, and will be used to trigger PDN's revisit decision.
4. No other prior information on the environment is provided. Planning and evaluation will be performed online while the SLAM pose-graph is built.

Given the above assumptions, PDN solves for an intelligent solution to the area coverage planning problem while considering SLAM's navigation performance. In this work, the vehicle's default exploration policy is a typical boustrophedon pattern that provides nominal area coverage of the ship hull exterior (Vaganay et al., 2009).

4.1 Waypoint Generation

Although all nodes in the pose-graph could be considered as possible waypoints, evaluating the outcome for all possible revisits is impractical. Moreover, due to the uneven spatial distribution of feature-rich areas in the environment, not all nodes are visually plausible candidates for registration anyway. Therefore, PDN computes expected rewards for only a subset of candidate nodes selected for their visual saliency level, resulting in only a subset of locally and globally salient nodes being identified as candidate waypoints.

First, we threshold acquired keyframes based upon their local saliency level to generate a set of candidate nodes with strong local saliency scores. In PDN, a local saliency threshold of $S_L^{\mathsf{wp}} = 0.5$ is used. Next, an online clustering algorithm, Density-Based Spatial Clustering of Applications with Noise (DBSCAN) (Ester et al., 1996; Daszykowski et al., 2001), groups locally salient nodes into local neighborhoods, forming clusters. Finally, within each cluster, we select a representative waypoint node by considering both its visual uniqueness (i.e., high global saliency level) and usefulness for loop-closure (i.e., lowest pose uncertainty). This process allows us to compute the N_{wp} best candidate waypoints for revisitation.

Fig. 1(d) shows a typical result. Within each cluster of locally salient nodes, the node with the lowest pose uncertainty among the globally salient nodes is selected as the representative waypoint. For example, 'C', 'D' and 'B' belong to a cluster while 'D' has been selected as the waypoint for this cluster. The clustered waypoints are then sorted by time and assigned with a waypoint number from 1 to N_{wp}, where waypoint number 0 is reserved for the exploration action.

4.2 Path Generation

Using this set of waypoints, PDN evaluates the reward that can be obtained by revisiting versus exploring. The robot computes a shortest path from its current pose to each waypoint in order to evaluate the expected reward along that path. In our application, finding the shortest path can be viewed as a traditional point-to-point path planning problem. We use the global A* algorithm (Russell and Norvig, 2003), but with the heuristic function weighted by local saliency:

$$d(\mathbf{x}_i, \mathbf{x}_k) = w(S_L^k) \cdot \sqrt{(x_i - x_k)^2 + (y_i - y_k)^2 + (z_i - z_k)^2}. \quad (1)$$

The weight term, $w(S_L^k)$, is modeled in such a way as to double the Euclidean distance to nodes with zero saliency while preserving the original distance to nodes with full saliency (recall that $S_L \in [0, 1]$),

$$w(S_L^k) = 2 - S_L^k. \quad (2)$$

Repeated bisection of sample nodes in the pose-graph yields a sequence of intermediate nodes, called milestones. During the generation of these milestones, we interpolate between milestones if necessary to complete a path, $\mathcal{P} = \{\mathbf{x}_p\} = \{\mathbf{x}_i\}_{i=0}^{p-1}$, which represents a candidate loop-closure action. Fig. 3 depicts sample point-to-point paths for two different synthetic local saliency maps. Using the current robot node as the starting point, the computed paths reach the waypoint as the

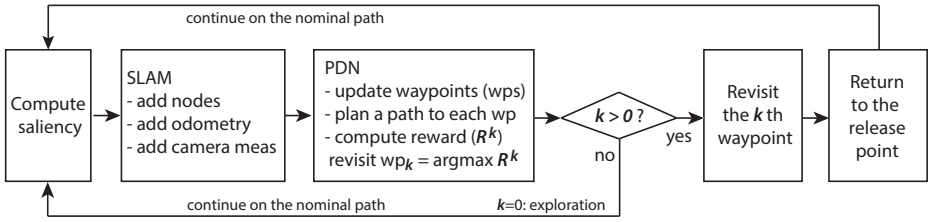
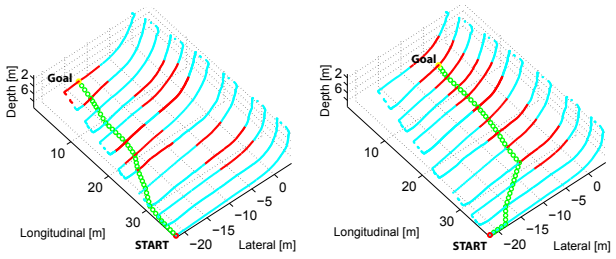


Fig. 2: A block-diagram of the PDN framework. Given access to the current SLAM pose-graph and saliency map, PDN selects a set of candidate waypoints, plans paths to these waypoints, and computes the rewards associated with these revisit paths. The reward, $\mathcal{R}^k$, is computed for each waypoint k, where $k = 0$ corresponds to the reward from exploration (i.e., $\mathcal{R}_{exp} = \mathcal{R}^0$). Lastly, either a revisiting or exploration action is executed to yield the maximum reward.



(a) Distributed saliency (b) Concentrated saliency

Fig. 3: Point-to-point path planning example using a synthetic saliency map for illustration. Cyan dots (light gray in black and white print) represent nonsalient nodes in the graph and salient regions are marked red (dark gray in black and white print). The planned paths are depicted with a sequence of green circles linking the start and goal positions. (a) and (b) are sample point-to-point paths for two different local saliency distributions. Note that saliency weighted A* results in paths biased toward the salient regions in the environment.

goal node. Note that the resulting paths are biased toward passing through salient regions in the environment due to the saliency weighted heuristic function (1). Because local minima may occur with the saliency weighting heuristic, we detect and avoid these using a perturbation action (Kavraki and LaValle, 2008) that tests using a pure Euclidean distance heuristic.

4.3 Reward for a Path

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

4.3.1 Saliency-based Measurement Probability

For each point-to-point candidate path, we solve for the estimated round-trip robot pose uncertainty. Using expected odometry and camera measurements along the path, the robot can estimate the final terminating pose covariance along that trajectory. However, estimating the amount of information gained from future camera measurements is not exact, and we need to develop a way of approximating the camera measurement likelihood. Camera measurements are binary, either success (one) or failure (zero), with the camera link event, L, being a Bernoulli random variable. When adding a set of expected camera measurements, we use local saliency to empirically model the probability of successful pairwise image registration. This probability is in turn used to compute the expected camera measurement information gain along the candidate path. The observation is that we can model this probability using statistics from prior SLAM and saliency results.

For the Bernoulli random variable, L, we seek to model its probability of success, P_L. Because each link is associated with two local saliency levels—the current node saliency, S_{L_c}, and the target node saliency, S_{L_t}—we can build a probability of link success parameterized by these two saliency levels:

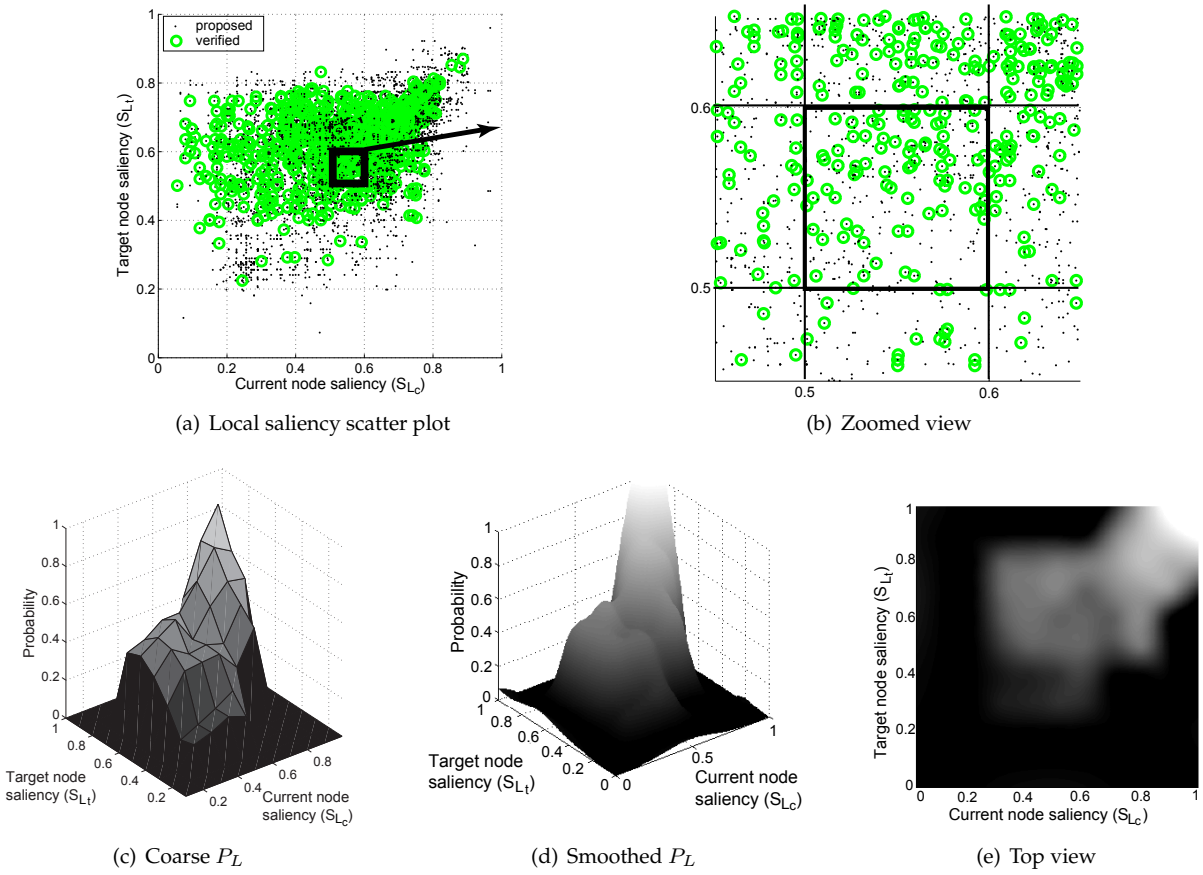
$$P_L = P_L(l = 1; S_{L_c}, S_{L_t}) \sim \text{Bernoulli}. \tag{3}$$

To empirically measure this probability, we generate a scatter plot from prior data and divide it into a set of bins with bin size of $\delta = 0.1$ (Fig. 4(a)–(b)). The empirical probability of link success is then calculated by counting the number of proposed links versus the number of verified links in each bin, which builds up a coarse model (Fig. 4(c)) as a function of the two associated saliency values (the current node saliency and the target node saliency). Then, this coarse model is smoothed using surface fitting (Fig. 4(d)–(e)).

4.3.2 Robot Uncertainty Term $\mathcal{U}^k_{robot}$

We use Fisher information for evaluating the resulting covariance matrix for integrated SLAM and path planning. Because camera measurements are not certain, we compute the expected information gain along a path, from which we evaluate the expected terminating covariance matrix. We use the determinant of this covariance matrix as a measure of navigation uncertainty.

The process of evaluating this terminating covariance is illustrated in Fig. 5. Note that only the outbound portion of the revisit action is depicted for visual clarity (PDN computes the information for the round-trip).



(a) Local saliency scatter plot

(b) Zoomed view

(c) Coarse P_L

(d) Smoothed P_L

(e) Top view

Fig. 4: Empirical probability of link success, P_L, as computed from prior data. The model is generated as a function of pairwise saliency levels (S_{L_c} and S_{L_t}). To model this, we use a scatter plot of link trials, (a) and (b), using data from previous missions from three different vessels (the *R/V Oceanus*, the *USCGC Venturous*, and the *USCGC Seneca*) each of different vessel size and visual feature distribution (Kim and Eustice, 2013b). A coarse distribution for $P_L(l = 1; S_{L_c}, S_{L_t})$ is then built by calculating the ratio of verified links to the number of proposed links in each bin of the scatter plot (c). Then, surface-fitting to this coarse result yields the final smooth distribution (d) and (e).

The approach is to construct a small SLAM instance by adding a set of odometry constraints and a set of expected camera measurements in the form of delta information to the current information matrix, Λ_0. In the toy example shown, a robot starts from node A and moves along the thick line, reaching the current node, denoted 0, at time t. To evaluate the terminating covariance of revisiting node A, two sources of delta information are added: one from odometry, $\Lambda_{\mathtt{odo}}$, and the other representing camera constraints, $\Lambda_{\mathtt{cam}}$. Summing these three information matrices builds PDN's expected information matrix:

$$\Lambda_{\mathtt{pdn}} = \Lambda_0 + \Lambda_{\mathtt{odo}} + \Lambda_{\mathtt{cam}}. \tag{4}$$

The expected delta information from odometry measurements, $\Lambda_{\mathtt{odo}}$, is built from a sequence of virtual nodes. Starting from the current node, $\mathbf{x}_0$, the odometry noise covariance, Q, and path, $\mathcal{P} = \{\mathbf{x}_i\}_{i=0}^{p-1}$, are summed for all expected odometry measurements for the round-trip travel to the waypoint along the revisit path,

$$\begin{aligned}\Lambda_{\mathtt{odo}} &= \sum_{i=0}^{p-1} \mathrm{H}_{\mathtt{odo}_{i+1,i}}^{\top} \cdot \mathrm{Q}_{i+1,i}^{-1} \cdot \mathrm{H}_{\mathtt{odo}_{i+1,i}} \quad \text{Outbound} \\ &+ \sum_{i=p}^{1} \mathrm{H}_{\mathtt{odo}_{i-1,i}}^{\top} \cdot \mathrm{Q}_{i-1,i}^{-1} \cdot \mathrm{H}_{\mathtt{odo}_{i-1,i}} \quad \text{Inbound.}\end{aligned} \tag{5}$$

The noise, $\mathrm{Q}_{j,i}$, for the odometry constraint is scaled with the travel distance between nodes $\mathbf{x}_i$ and $\mathbf{x}_j$. The odometry measurement model is the relative-pose between two sequential nodes and can be represented using the tail-to-tail operation by Smith et al. (1990). The resulting Jacobian, $\mathrm{H}_{\mathtt{odo}_{j,i}}$, is sparse with nonzero block matrices on the i^{th} and $(i+1)^{\text{th}}$ elements. Hence, summing all odometry information results in a block-tridiagonal matrix (Fig. 5).

For the camera measurements, we similarly add all expected camera measurements along the revisit path. Because PDN proposes the same number of link hypotheses, n_{plink}, as in the normal SLAM process, there

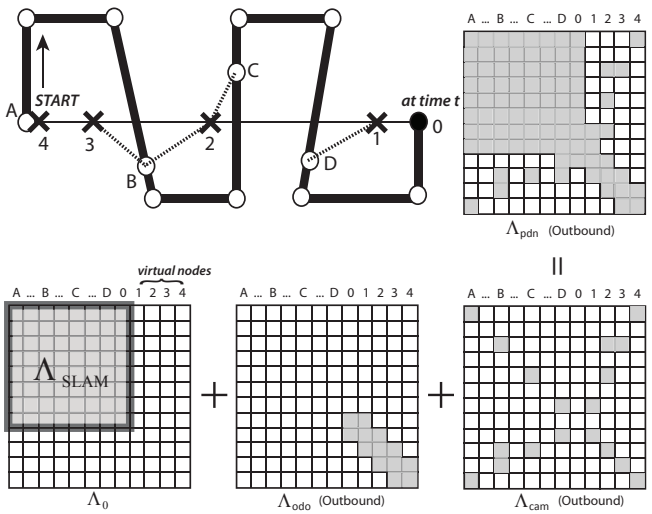


Fig. 5: Robot pose uncertainty propagation in PDN. Only the outbound revisit action is illustrated for simplicity (PDN computes the information for the round-trip). The robot starts from node A moving along the thick line, and reaches the current node 0 at time t. This illustration shows construction of the PDN information matrix when the robot executes a revisit action from the current node 0 to a revisit point A (nonzero information is depicted by the gray matrix elements). The revisit action is marked with a thin line linking 0 and A with virtual nodes 1, 2, 3, and 4 along the revisit path to A. Nodes A, B, C, and D are existing nodes in the pose-graph, and also are the candidate nodes that these virtual nodes make camera measurements with. The expected camera measurements are marked with a dotted line between 1–4 and A–D.

are multiple expected camera measurements per each virtual node along the path. When a virtual node is $\mathbf{x}_i$ and the candidate paired for camera measurement is node $\mathbf{x}_c$, the first-order expected information gain from camera measurements is calculated as

$$\begin{aligned}\Lambda_{\mathtt{cam}} &= \sum_{i=0}^{p-1}\sum_{c\in\mathcal{L}_i} P_L \cdot \mathrm{H}_{\mathtt{cam}_{c,i}}^{\top}\mathrm{R}^{-1}\mathrm{H}_{\mathtt{cam}_{c,i}} \quad \text{Outbound} \\ &+ \sum_{i=p}^{1}\sum_{c\in\mathcal{L}_i} P_L \cdot \mathrm{H}_{\mathtt{cam}_{c,i}}^{\top}\mathrm{R}^{-1}\mathrm{H}_{\mathtt{cam}_{c,i}} \quad \text{Inbound}, \end{aligned} \tag{6}$$

where $\mathrm{H}_{\mathtt{cam}_{c,i}}$ is the camera measurement Jacobian (Kim and Eustice, 2009), R is the (fixed) camera measurement noise covariance, $\mathcal{L}_i$ is the index set of camera measurement candidates associated with virtual node $\mathbf{x}_i$, and $P_L = P_L(l=1; S_{L_c}, S_{L_t})$ is the empirical probability (3) of the link to be successful. Unlike odometry measurements, not all expected camera measurements result in registration success, which depends greatly upon the visual feature distribution in the environment, this effect is captured by P_L.

Finally, adding these three information matrices as in equation (4) yields the expected information matrix for pursuing a virtual path to the waypoint. For the reward calculation, we present two possible ways to efficiently obtain the terminating covariance. The first is to use an exactly sparse delayed-state filter (Eustice et al., 2006a) to evaluate Σ_{nn}^k, from the expected information matrix by recovering the n^{th} block-column of the covariance matrix, Σ_{*n}^k, as per Eustice et al. (2006b),

$$\Lambda_{\mathrm{pdn}}\Sigma_{*n}^k = \mathrm{I}_{*n}, \tag{7}$$

where I_{*n} is the n^{th} block-column of the $n \times n$ block identity matrix. This formulation is computationally efficient and avoids inverting the entire information matrix to recover the round-trip pose covariance for the k candidate waypoints.

Alternatively, the terminating covariance can be efficiently recovered from the square root information matrix when available (e.g., this is accessible in iSAM). Since we use iSAM as our SLAM back-end, we evaluate this covariance in an efficient way by directly adding expected measurement information to a copy of the square root information matrix using the same Givens rotation mechanism as described in Kaess et al. (2008). Using iSAM's efficient covariance recovery (Kaess and Dellaert, 2009), the terminating covariance can be evaluated with constant time complexity $O(1)$ (assuming that the virtual nodes have been appended onto the information matrix).

Next, the terminating covariance for exploration is computed by propagating forward the current SLAM pose covariance by one step. From the current SLAM node, we compute the resulting covariance,

$$\Sigma_{\text{exp}} = \Sigma_{r+1,r+1} = \mathrm{H}_{\mathtt{odo}_{r+1,r}}\Sigma_{rr}\mathrm{H}_{\mathtt{odo}_{r+1,r}}^{\top}. \tag{8}$$

Here, index r refers to the current robot node, which is also the last node in the existing pose-graph (all nodes later than r are virtual). Assuming a zero order hold on the previous odometry measurement, $\mathbf{x}_{r-1,r}$, then

$$\mathbf{x}_{r+1,r} = \mathbf{x}_r \oplus \mathbf{x}_{r-1,r}, \tag{9}$$

and $\mathrm{H}_{\mathtt{odo}_{r+1,r}}$ is its head-to-tail Jacobian.

Lastly, the penalty term for robot uncertainty, $\mathcal{U}_{\text{robot}}^k$, is computed as the ratio of the localization uncertainty for the next-best-action to the user-defined target navigation uncertainty, Σ_{target}. For the k^{th} waypoint, the robot uncertainty is defined as

$$\begin{aligned}\mathcal{U}_{\text{robot}}^{k=0} &= \begin{cases} 0, & \text{if } \frac{|\Sigma_{\text{exp}}|}{|\Sigma_{\text{target}}|} < 1 \\ \frac{|\Sigma_{\text{exp}}|^{\frac{1}{6}}}{|\Sigma_{\text{target}}|^{\frac{1}{6}}}, & \text{otherwise} \end{cases}, \\ \mathcal{U}_{\text{robot}}^{k>0} &= \frac{|\Sigma_{nn}^k|^{\frac{1}{6}}}{|\Sigma_{\text{target}}|^{\frac{1}{6}}}, \quad k = 1, \cdots, N_{wp} \end{aligned} \tag{10}$$

where the sixth root of the 6-degree of freedom (DOF) pose determinant is used (Carrillo et al., 2012; Kiefer, 1974) to yield a measure with SI units of $\mathrm{m}\cdot\mathrm{rad}$.

Essentially, PDN compares the two propagated uncertainties from revisiting and exploring, and then

chooses the smaller one whenever the exploration uncertainty exceeds the desired target uncertainty. When the revisit action has the same or less value than pursuing exploration, the revisit does not produce enough loop-closures to overcome the increased navigation uncertainty from detouring. Note that in the previous studies by Bourgault et al. (2002), Makarenko et al. (2002), and Stachniss et al. (2005), revisiting is always expected to be beneficial since there is no consideration for the actual likelihood of obtaining the loop-closure. In our approach, however, PDN has a realistic expectation for the likelihood of camera loop-closures based upon visual saliency.

4.3.3 *Area Coverage Term* $\mathcal{A}^k_{map}$

As a final step in the PDN evaluation, we evaluate an area coverage term to promote efficient coverage. Our purpose is to cover a target area in a timely manner while considering SLAM's navigation performance. In other words, without an area coverage term, there will be a trivial solution to this problem—to repeatedly revisit to keep the localization uncertainty small. To prevent this, the area coverage term for the k^{th} waypoint is defined as the ratio of area-to-cover with respect to the target-coverage-area,

$$\mathcal{A}^k_{\text{map}} = \frac{\mathcal{A}_{\text{to\_cover}}}{\mathcal{A}_{\text{target}}} = \frac{\mathcal{A}_{\text{target}} - \mathcal{A}_{\text{covered}} + \mathcal{A}^k_{\text{redundant}}}{\mathcal{A}_{\text{target}}}. \quad (11)$$

where the target coverage area is provided by the user. Here, $\mathcal{A}_{\text{target}}$ is the target coverage area, $\mathcal{A}_{\text{covered}}$ is the survey area already explored, and $\mathcal{A}_{\text{redundant}}$ is the expected redundant area coverage produced by a revisiting action. This additional area is proportional to the round-trip revisit path length and sensor field of view width. Its effect is to penalize long revisit actions.

4.3.4 *Combined PDN Reward Function*

To fuse the two penalty terms, we introduce a weight α that controls the balance between the pose uncertainty and area coverage terms. The uncertainty increase term corresponds to the penalty for SLAM, where the action with minimal uncertainty increase is preferred. The area coverage metric is the penalty in area coverage when performing an action. By taking a weighted sum of these two penalties, we can evaluate the total cost, $\mathcal{C}^k$, for each waypoint k. Reward is defined as negative cost, and PDN selects an action with the largest reward, or in other words, the one with minimal cost/penalty:

$$\mathcal{C}^k = \alpha \cdot \mathcal{U}^k_{\text{robot}} + (1-\alpha) \cdot \mathcal{A}^k_{\text{map}}, \quad (12)$$

$$\mathcal{R}^k = -\mathcal{C}^k. \quad (13)$$

By adjusting α, we can change the emphasis on robot navigation uncertainty versus area coverage performance in the reward evaluation. When $\alpha = 0$, no weight is imposed on the pose uncertainty and the algorithm tries to cover the area as fast as possible. This corresponds to an open-loop survey over the target area. On the other hand, when $\alpha = 1$, full weight is on the pose uncertainty and the robot will revisit whenever it exceeds the target navigation uncertainty. In other words, the value α controls the softness of the desirable uncertainty constraint. With $\alpha = 1$, the desirable uncertainty acts as a hard constraint on the system whereas the desirable uncertainty has no constraints on performance when it is zero. Our approach allows intuitive selection of weight as it balances between two normalized terms, whereas weighting factors are experimentally determined in other works (Du et al., 2011). The effect of α's selection is explored in §5.1.2.

The revisiting waypoint, k^*, is determined by maximizing the reward,

$$k^* = \underset{k}{\text{argmax}}\, \mathcal{R}^k = \underset{k}{\text{argmin}}\, \mathcal{C}^k, \quad (14)$$

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

5 Results

In this section, we present an evaluation of PDN as applied to a hybrid simulation and a real-world ship hull inspection experiment.

5.1 Hybrid Simulation

For the hybrid simulation, we use a dataset collected from a survey of the *SS Curtiss* conducted in February 2011 using the Hovering Autonomous Underwater Vehicle (HAUV) (Kim and Eustice, 2013b). A dense SLAM result (Fig. 6) was built from this dataset and is used in the hybrid simulation. Using this densely sampled SLAM result as a baseline, we plan a simulated mission by subsampling from it a set of nominal trajectory nodes—unused nodes are reserved for PDN to simulate revisit actions. In the first set of tests, we impose a synthetic saliency distribution on the mapping area to evaluate the performance of PDN using known ground-truth. In the second set of tests, we conduct a hybrid simulation using the actual recorded hull imagery.

In all cases of evaluation PDN is compared against two typical survey patterns: "open-loop survey (OPL)" and "deterministic revisit (DET)". OPL follows a nominal boustrophedon area-coverage exploration policy without any revisiting, while DET does the same but with additional deterministic revisit actions to achieve loop-closures. This deterministic revisit strategy is typical of underwater vehicle operations, is usually passively preplanned or executed by a human pilot, and serves as a practical real-world benchmark. In the reported simulation, DET commands the vehicle to return

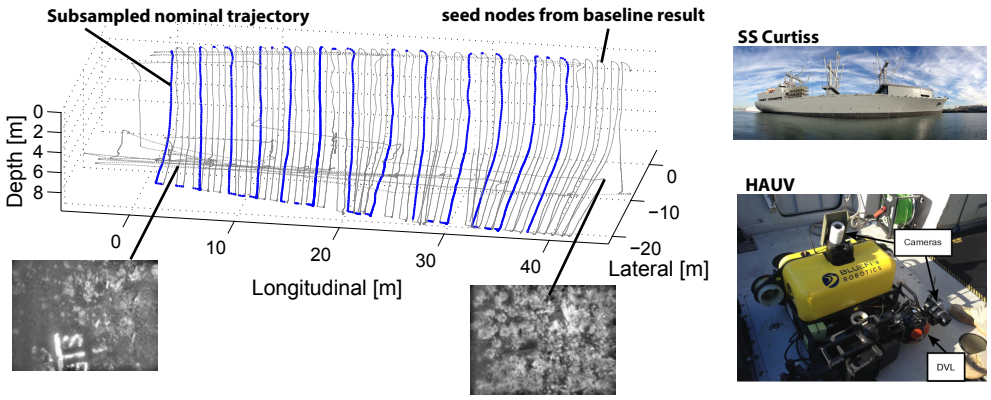
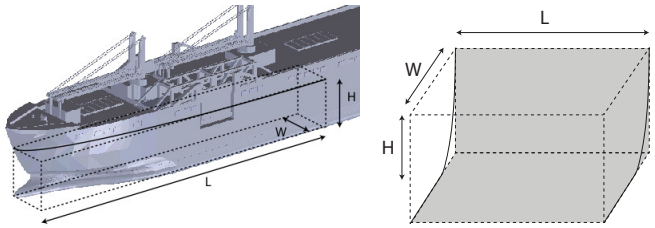


Fig. 6: Experimental setup of the hybrid simulation used for PDN's evaluation. Gray dots are nodes from the baseline dense trajectory SLAM result. The sampled nominal trajectory in blue mimics a simulated mission by sub-sampling from the baseline SLAM result. The nodes not used in the nominal trajectory planning are used as seed nodes in the simulated control phase. Note that each seed node is associated with a real underwater image. Thumbnails show the HAUV, the vessel we surveyed (*SS Curtiss*), and sample imagery from its hull.



(a) Box approximation (b) Target coverage area

Fig. 7: Target coverage area calculation. The area of interest is approximated by a bounding box where the target coverage area is computed using the vessel's length L, width (half beam) W, and draft H.

back to the first trackline on every other survey leg, regardless of the actual feature distribution along the revisit trajectory.

In our application, the SLAM navigation uncertainty is dominated by xy positional uncertainty because depth, z, is bounded with absolute pressure-depth measurements. For a given desired allowable positional uncertainty, the overall target navigation covariance bound used in equation (10) is set as

$$|\Sigma_{\text{target}}| = \sigma_{x,\text{target}}^2 \cdot \sigma_{y,\text{target}}^2 \cdot \sigma_z^2 \cdot \sigma_r^2 \cdot \sigma_p^2 \cdot \sigma_h^2, \qquad (15)$$

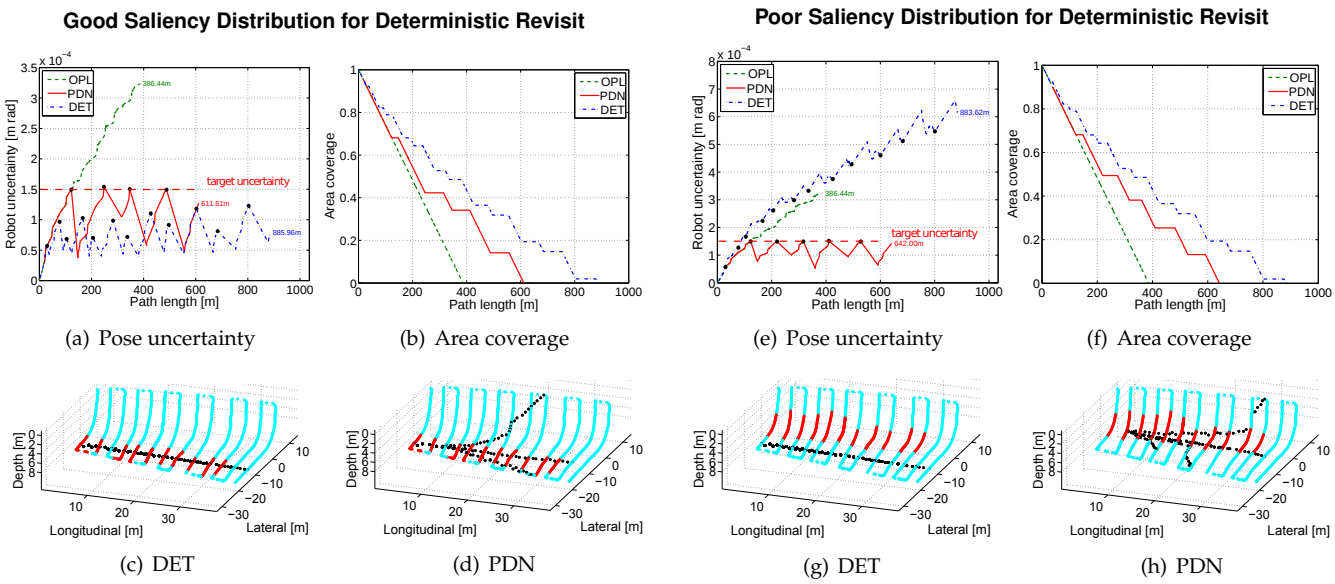
where in this evaluation we used positional uncertainty $\sigma_{x,\text{target}} = \sigma_{y,\text{target}} = 0.25\,\text{m}$, depth uncertainty $\sigma_z = 0.01\,\text{m}$, and attitude uncertainty $\sigma_r = \sigma_p = \sigma_h = 0.1^\circ$ (roll, pitch and heading), which evaluates to $|\Sigma_{\text{target}}|^{\frac{1}{6}} = 1.49 \times 10^{-4}\,\text{m}\cdot\text{rad}$. The target coverage area (Fig. 7) was computed using the desired survey longitudinal length L, survey width (ship half-beam) W, and survey depth (ship draft) H, which for this experiment was set as $\mathcal{A}_{\text{target}} = L \times (W + H) = 40\,\text{m} \times (20\,\text{m} + 10\,\text{m}) = 1200\,\text{m}^2$.

5.1.1 PDN with Synthetic Saliency Map

The first set of tests are with a synthetic saliency map imposed over the area. We set $\alpha = 1$ so that full weight is given to the pose uncertainty. The PDN action is evaluated for two different types of saliency distributions, distributed and concentrated, for which we compare the results of PDN to DET and OPL exploration policies. For DET, the robot is commanded to revisit a point on the first trackline in every other survey leg. In this setup, the revisit happens on a path along the bottom of the hull (i.e., the keel). Because in practice this revisit path is typically preplanned without knowledge of the actual feature distribution in the environment, we use the same deterministic revisit path for both the distributed and concentrated saliency simulations.

Fig. 8 shows the SLAM navigation results for the two different saliency scenarios. When we have a biased feature distribution, as in our simulation, the deterministic revisit path can either be always on the salient regions, Fig. 8(c), or never pass through the salient regions, Fig. 8(g). A measure of the robot's pose uncertainty, $|\Sigma_{rr}|^{\frac{1}{6}}$, is plotted in Fig. 8(a) and (e). Fig. 8(b) and (f) plot the ratio of the remaining area to cover with respect to the path length, where the black dots indicate instances when revisits occurred.

When all of the deterministic revisit paths land on the salient region, the likelihood of obtaining loop-closures during the revisit is higher, and DET achieves tightly bounded uncertainty for the robot pose. On the other hand, when none of the revisit paths are on salient regions, as in the case of Fig. 8(g), DET basically performs worse than OPL. Without meaningful loop-closures, the revisit excursion just increases path length and slows the overall coverage rate, as can be seen in Fig. 8(e) and (f)—in practice, the visual saliency distribution cannot



(a) Pose uncertainty (b) Area coverage (e) Pose uncertainty (f) Area coverage

(c) DET (d) PDN (g) DET (h) PDN

Fig. 8: Simulation results for synthetic saliency maps that result in either a good or poor visual feature distribution for deterministic revisit strategies. The performance of DET strongly depends upon the spatial distribution of feature-rich regions in the target area and their intersection with the preplanned revisit path, while for PDN it is able to automatically adapt to the saliency distribution in an intelligent way to maintain consistent SLAM navigation performance. (a), (b), (e), (f) Pose uncertainty and area coverage performance are compared for OPL (green), DET (blue), and PDN (red). (c), (d), (g), (h) Trajectory of the robot with nodes color-coded by their saliency level—red/salient, cyan/nonsalient—black dots indicate revisit trajectories.

be known in advance. Note that for both cases (concentrated and distributed saliency), the total path length and area coverage rate stay the same for DET since it is preplanned. On the other hand, PDN shows even performance over both saliency cases since it is able to automatically adapt its revisit actions to the environment, yielding consistent navigation and area coverage performance under both scenarios.

5.1.2 *Effect of α in PDN*

Next, we examine the effect of the parameter α on PDN's performance. The parameter α controls how much weight is given to the pose uncertainty versus area coverage. When $\alpha = 0$, PDN does not assign importance to the pose uncertainty, and the framework works the same as OPL. When $\alpha = 1$, full weight is given to the pose uncertainty, and PDN tries to reduce the uncertainty once it reaches the target uncertainty threshold. In other words, the effect of α is to delay the execution of revisiting by PDN.

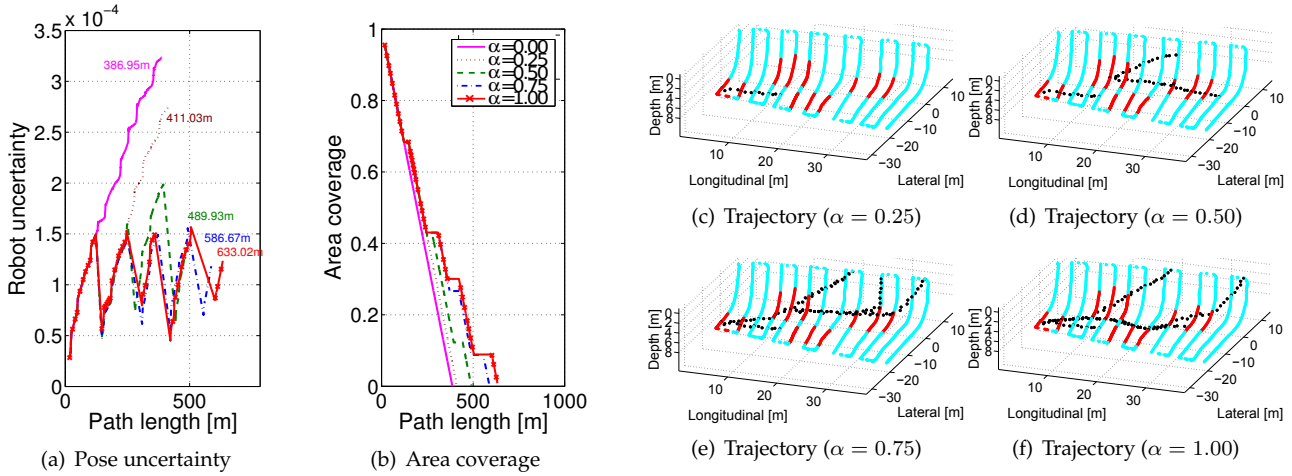
The effect of α can be seen in Fig. 9, which presents several SLAM trajectories with different α weight factors. The uncertainty is most well bounded when $\alpha = 1$, and relaxes as α decreases. For area coverage, $\alpha = 0$ shows the fastest coverage rate, which is slowed as α increases (i.e., weights pose uncertainty more). As the weight on pose uncertainty increases (from 0 to 1 in 0.25 increments), PDN tends to revisit the furthest waypoint more often to result in larger loop-closures. When the weight is small, however, PDN allows the pose uncertainty to increase in order to cover the area faster. In this case, revisit waypoints are likely to be nearby positions so as not to delay area coverage performance.

5.1.3 *PDN with Hybrid Simulation*

In this experiment, we evaluate PDN's performance using the actual recorded underwater imagery from the 2011 *SS Curtiss* dataset, the images are used for both saliency generation and actual image registration. This time a weighting factor of $\alpha = 0.75$ is used so that navigation uncertainty is only given a mild preference over area coverage performance.

Like in the previous synthetic saliency map experiment, the uncertainty and area coverage graph for PDN are compared with OPL and DET. Based upon our knowledge of the actual resulting saliency distribution obtained in the baseline SLAM result, we preplanned the DET revisit path to be over a visually salient region on the hull to provide the best possible case for comparison with PDN. Because the DET revisit is intentionally planned over the salient region, the resulting graph for DET shows an optimistic SLAM performance—maintaining low uncertainty, but producing a large number of revisits (twelve) and longer path length (867.53 m).

Fig. 10(a) and (b) show the uncertainty change and area coverage rate, respectively, for OPL, DET, and PDN. As shown in Fig. 10(e), PDN adapted its trajec-

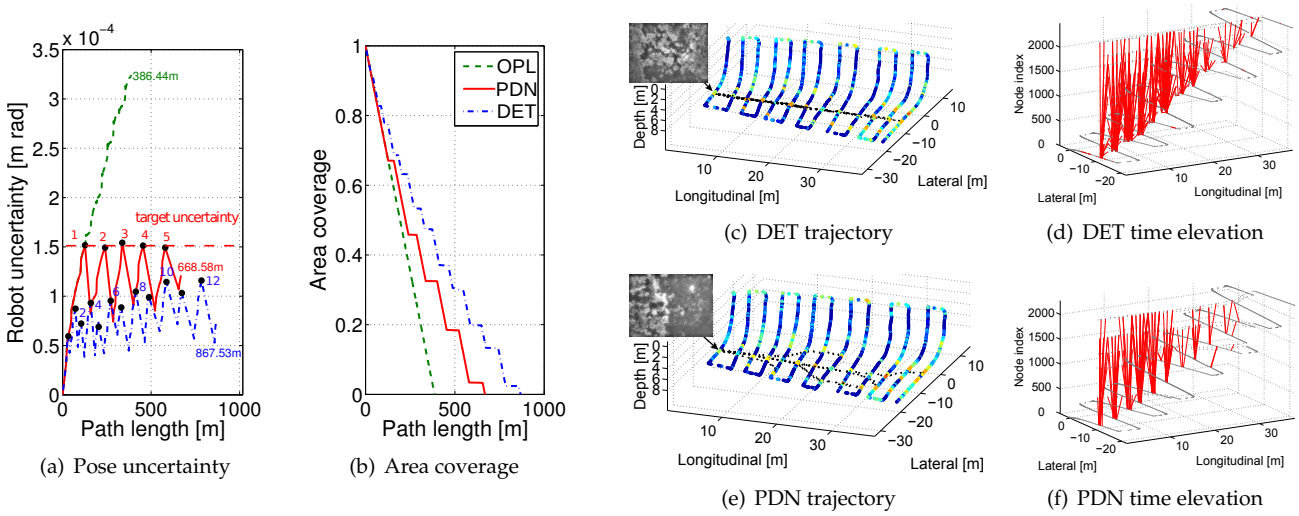


(a) Pose uncertainty (b) Area coverage

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

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

Fig. 9: Simulation results comparing PDN's performance for different values of the α weighting factor when given a spatially distributed saliency distribution. When $\alpha = 0$, PDN performs open-loop control; when $\alpha = 1$, PDN reacts instantly once the pose uncertainty exceeds the target uncertainty level. (a) The change in pose uncertainty with respect to α. (b) Area coverage rate in terms of α. (c)–(f) PDN-aided SLAM trajectories for the different values of α.



(a) Pose uncertainty (b) Area coverage

(c) DET trajectory (d) DET time elevation

(e) PDN trajectory (f) PDN time elevation

Fig. 10: Hybrid simulation results using actual real imagery from the February 2011 *SS Curtiss* dataset. Similar to the synthetic saliency case, (a) and (b) show the pose uncertainty and area coverage performance, respectively, with respect to the path length—annotated black dots indicate the revisit trajectories. In surveys (c) and (e), nodes are color-coded by their measured visual saliency from the real imagery; for visual clarity, nodes included in the revisit path are shown as black dots. The deterministic revisit was purposely preplanned over the salient region, note that PDN is able to automatically find this same optimal path to follow (e). In the time elevation graphs (d) and (f), PDN shows a similar number of successful loop-closures to DET.

tory to obtain visual loop-closures to reduce the uncertainty whenever it exceeds the target covariance bound. Note that the number of revisits by PDN (five) is substantially smaller than DET (twelve)—PDN uses a fewer number of revisits while still maintaining full control over the navigation uncertainty level. The loop-closing camera measurements are clearly illustrated in the time elevation graph of Fig. 10(d) and (f). The red lines in the graph depict the camera measurements made by the loop-closures. Because the inter-trackline spacing is wide, there is no image overlap between adjacent tracklines, and all of the camera measurements in the graph are from revisit actions. As can be seen in the time elevation graphs, PDN obtained a similar number of loop-closures as compared to DET. A detailed depiction of PDN operating during the hybrid simulation is available as multimedia attachment Extension 1.

For statistical significance, we repeated this experiment over 10 trials of the hybrid simulation to evaluate the performance of PDN. To conduct this test we sampled 10 nominal seed trajectories from the baseline SLAM result, along with two different deterministic

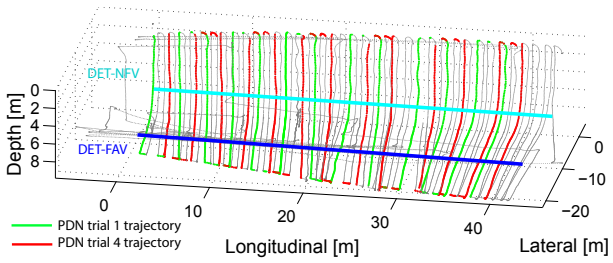


Fig. 11: Representative nominal policy trajectories (two of the 10 trials are shown) and DET favorable and nonfavorable revisit paths used in the 10-run multi-trial hybrid simulation.

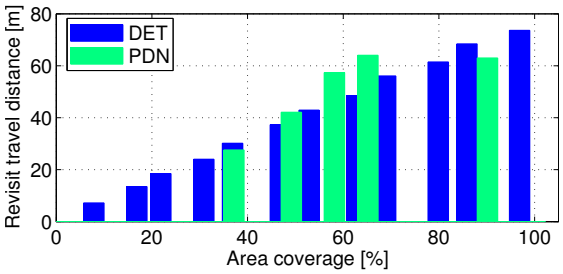


Fig. 13: A plot of DET and PDN revisit travel distances versus percentage survey area completion, sample results are shown for trial 2 of the multi-run hybrid simulation. Each bar represents a loop-closure revisit detour where the height of each bar indicates the round-trip distance traveled to accomplish the revisit. Notice that DET makes 12 revisits, which occur with a regular cadence (i.e., DET revisits on every other trackline) whereas PDN makes only 5 revisits, which occur adaptively based upon the target navigation uncertainty. Overall, PDN completes the survey in a shorter total mission path length (663 m) as compared to DET (879 m).

cases. The first deterministic case, "DET-FAV", results in a favorable revisit plan over visually salient regions in the environment, like previously depicted in Fig. 10(c). The second deterministic case, "DET-NFV", results in a nonfavorable revisit path and arbitrarily crosses along the nonsalient region located alongside the hull. Fig. 11 depicts two trials from this multi-trial experiment along with the two deterministic plans used.

Fig. 12 summarizes the 10-run multi-trial results. In Fig. 12(a) and (b) we see the pose uncertainty and area coverage rate, respectively, versus mission path length (only five of the available ten sample trajectories are shown for visual clarity). In Table 2 we see a tabulated comparison of PDN's performance versus OPL and DET for all 10 trials. Overall, PDN outperforms OPL and DET in all sample trajectories, especially in the case of DET-NFV, which is preplanned over an unfavorable (feature-poor) visual region. Coverage-wise, all of PDN's sample trajectories are able to complete the area coverage task faster than either of the DET cases, using anywhere from five to eight automatically selected revisits versus DET's twelve. Fig. 13 depicts the round-trip revisit actions for trial 2 of the multi-trial results and clearly shows PDN's adaptive revisit behavior as opposed to DET's nonadaptive behavior, which revisits regularly without regard to navigation uncertainty or visual saliency.

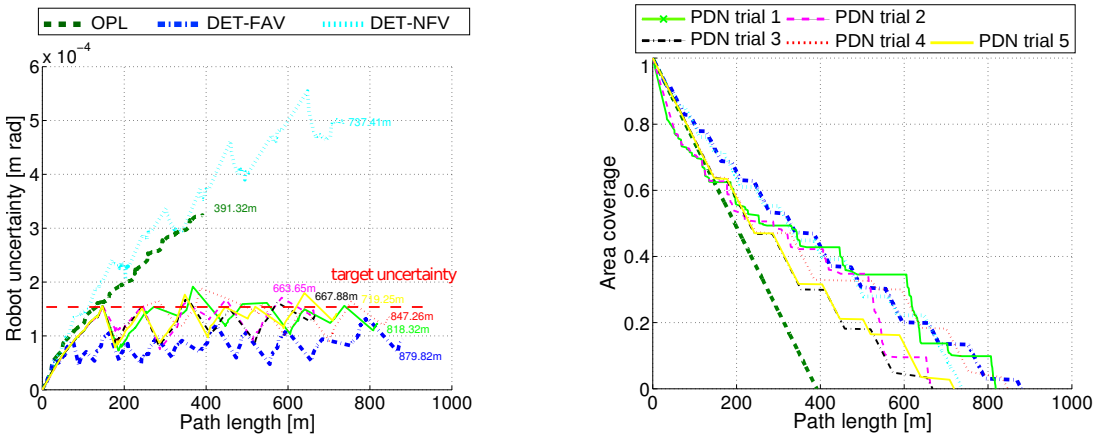
5.2 PDN with Real-World Evaluation

In February 2013 the *SS Curtiss* was again surveyed by the HAUV, but this time with PDN running in real-time during the experiments. Similar to the survey in 2011 (Kim and Eustice, 2013b), the HAUV was equipped with a monocular camera running at 2 Hz, 1200 kHz Doppler velocity log (DVL) for odometry, and performed PDN on top of a default boustrophedon exploration policy. Following this default mission policy, PDN clustered salient nodes to generate target revisit points (§4.1), calculated paths to each target revisit waypoint (§4.2), and proposed the revisit path with the highest reward. When a proposed revisit was requested, the HAUV executed the round-trip revisit path using the waypoint navigation method described by Hover et al. (2012).

In this set of experiments, we compare the actual real-time results from PDN versus an actual preplanned deterministic revisit mission and an actual open-loop survey mission, just like in the hybrid simulation case. A value of $\alpha = 0.75$ and target coverage area of $A_{\text{target}} = 1200\ \text{m}^2$ were used in generating PDN's real-time performance, just like in the simulation, however for this experiment we reduced the target xy positional uncertainty from 0.25 m to 0.075 m for more precise localization, which resulted in a target navigational uncertainty of $|\Sigma_{\text{target}}|^{\frac{1}{6}} = 6.69 \times 10^{-5}\ \text{m} \cdot \text{rad}$.

Because in the 2011 survey the feature distribution was mostly concentrated at the bottom of the *SS Curtiss'* hull (see Fig. 10(c)/(e)), we preplanned our 2013 DET revisit path to follow along the bottom of the hull, assuming that a similar visual feature distribution would be available as seen in 2011. However, unbeknownst to us, the *SS Curtiss* was dry-docked and its hull cleaned sometime during the two year interval between 2011 and 2013, resulting in a drastically different visual feature distribution on the hull. In the 2013 experiments, visual features were mostly concentrated along the side of the hull, with few to any appearing at the bottom of the hull. As a result of this, the preplanned DET path was unfavorable and yielded very few loop-closures. Fig. 14(a) illustrates the resulting unbounded uncertainty caused by DET's unsuccessful revisits. In fact, the HAUV was able to reduce pose uncertainty only once with one of the DET revisit paths, which highlights the practical real-time need for in-situ active SLAM trajectory planning.

In comparison, PDN was able to detect and properly adapt its revisit trajectory to the actual visual feature distribution that it learned online from the visual saliency map. PDN attempted three loop-closure revis-



(a) Pose uncertainty (b) Area coverage

Fig. 12: Hybrid simulation results for the 10-run multi-trial experiment. As visualized in Fig. 11, the favorable deterministic case, "DET-FAV", has a preplanned revisit path over the visually feature-rich region while the nonfavorable deterministic case, "DET-NFV", has an arbitrarily preplanned revisit path over a feature-poor region. (a) and (b) show the pose uncertainty and area coverage performance with respect to the mission path length; the legend is common between the two plots. Only five of the available 10 trials are depicted for visual clarity.

Table 2: Hybrid simulation results for the 10-run multi-trial experiment. The average (AVG), maximum (MAX), and minimum (MIN) value obtained during the 10 trials are provided for total path length, maximum pose uncertainty, and total number of revisits. The ratio to open-loop path length and the ratio to the target navigation uncertainty (i.e., $|\Sigma_{\text{target}}|^{\frac{1}{6}}$) are given in the parentheses. Positional error in xy-direction is given below. AVG (90%) indicates the mean excluding 5% of the highest and 5% of lowest values.

		OPL	DET-NFV	DET-FAV	PDN
Path length [m]	AVG	392.77	872.55 (222.15%)	875.29 (222.85%)	796.37 (202.76%)
	MAX	393.75	898.35 (228.16%)	882.05 (224.02%)	992.63 (252.10%)
	MIN	391.32	737.41 (188.44%)	866.81 (221.51%)	663.65 (169.59%)
$\lvert\Sigma_{rr,\max}\rvert^{\frac{1}{6}}$ [m · rad]	AVG	3.23E-4 (216.8%)	6.63E-4 (444.0%)	1.47E-4 (98.2%)	1.87E-4 (125.3%)
	MAX	3.25E-4 (217.9%)	7.25E-4 (485.8%)	1.80E-4 (120.5%)	2.72E-4 (182.4%)
	MIN	3.22E-4 (215.5%)	5.55E-4 (317.6%)	1.33E-4 (89.2%)	1.66E-4 (111.3%)
Revisits [#]	AVG	0	12	12	6.5
	MAX	0	12	12	8
	MIN	0	12	12	5
Loop-closures [#]	AVG	0.0	0.1	198.5	102.2
	MAX	0.0	1.0	307.0	199.0
	MIN	0.0	0.0	149.0	36.0
xy positional error [m]	AVG (90%)	1.28	1.23	0.38	0.55
	MAX	2.59	5.29	1.08	1.47

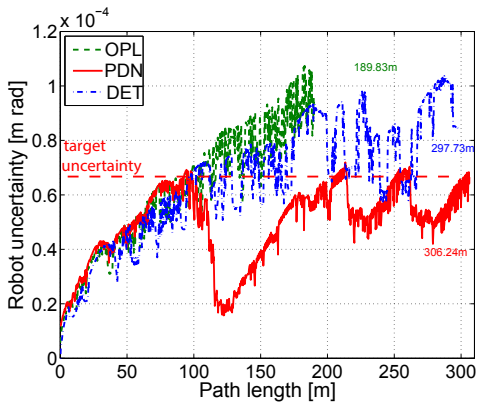
its during the area-coverage mission based upon our preset target navigation uncertainty. Even in the case where PDN's second planned revisit could not result in a significant uncertainty drop, PDN properly scheduled a third revisit in order to keep the navigation uncertainty bounded. Although PDN and DET show a similar overall mission path length (PDN 306.24 m and DET 297.73 m), only PDN was able to maintain a bounded pose uncertainty within the target navigation uncertainty threshold (Fig. 14(a)). Table 3 summarizes the real-world implementation result in terms of total path length, maximum pose uncertainty, and the number of revisits. A video depicting PDN's real-world performance versus DET is provided in Extension 2.

Table 3: Real-world results for the February 2013 experiments on the *SS Curtiss*. Shown are the total path length, maximum pose uncertainty, and total number of revisits for OPL, DET, and PDN. Like Table 2, the ratio to open-loop path length and the ratio to the target navigation uncertainty (i.e., $|\Sigma_{\text{target}}|^{\frac{1}{6}}$) are given in the parentheses.

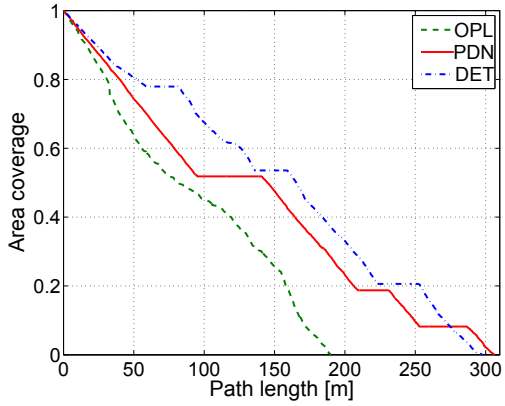
	OPL	DET	PDN
Path length [m]	189.8 (100.0%)	297.7 (156.8%)	306.2 (161.3%)
$\lvert\Sigma_{rr,\max}\rvert^{\frac{1}{6}}$ [m · rad]	1.07E-4 (160.6%)	1.03E-4 (155.0%)	0.72E-4 (107.6%)
Revisits [#]	0	12	7

6 Conclusion

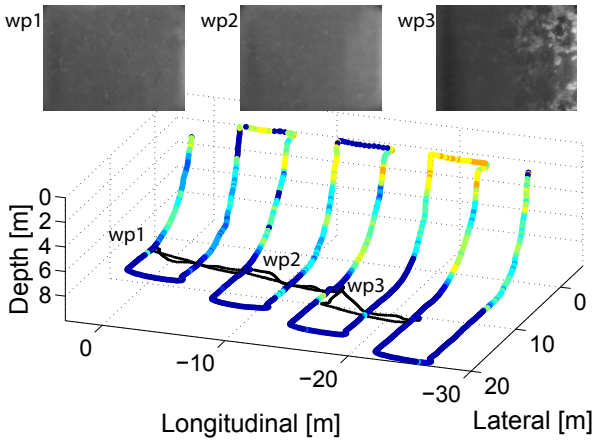
This paper presented perception-driven navigation, an active visual SLAM algorithm that takes into account area coverage and navigation uncertainty performance to efficiently explore a target area of interest. A weighting factor, α, provides control over this balance. A hy-



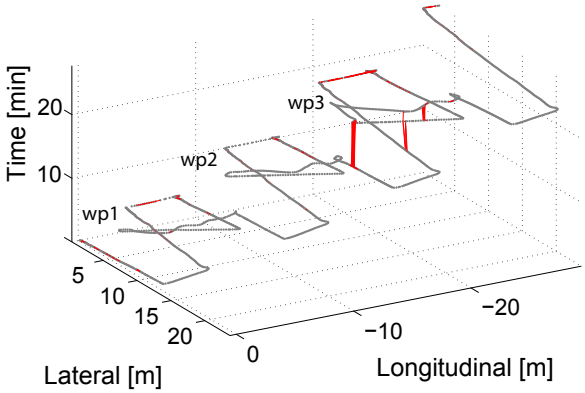
(a) Pose uncertainty



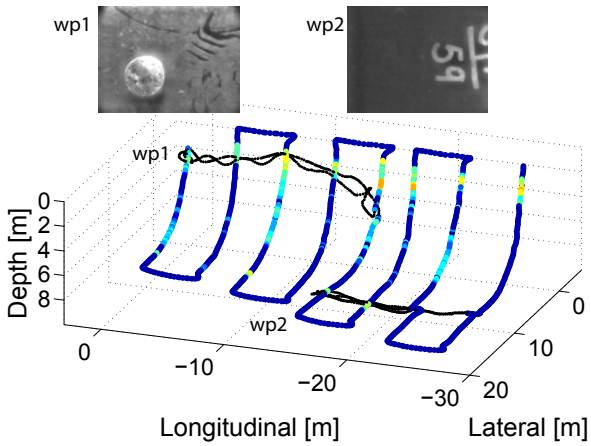
(b) Area coverage



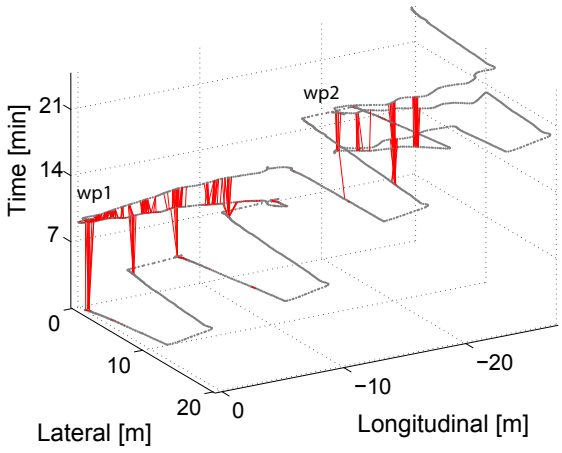
(c) DET trajectory



(d) DET time elevation



(e) PDN trajectory



(f) PDN time elevation

Fig. 14: Real-world results for the real-time PDN implementation used during a February 2013 survey of the *SS Curtiss* with the HAUV. (a) and (b) depict the pose uncertainty and area coverage performance, respectively, with respect to the path length. Consistent with other figures, the same color code scheme is used (i.e., OPL-green, DET-blue, and PDN-red). (c) and (e) are trajectories of the robot with nodes color coded by their visual local saliency level. For visual clarity, nodes included in revisit paths are marked with black dots. The revisit waypoints (wp) are denoted on the trajectory and their associated image keyframes are depicted above. (d) and (f) depict time elevation graphs highlighting successful camera loop-closures. Unlike the very small number of successful loop-closures in the deterministic revisit case, PDN proves its ability to automatically plan a feasible path to achieve visual loop-closures effectively.

brid simulation using trajectories with both synthetic and real underwater images were tested to evaluate PDN's performance. We also presented results from the first-ever, real-time, real-world implementation of PDN on a underwater ship hull inspection application, showing its ability to autonomously plan visually plausible revisit paths for loop-closure while controlling the navigation uncertainty level and achieving efficient area coverage rates.

PDN is a greedy approach in that it evaluates only one-step forward exploration versus revisit actions. Extending the look-ahead horizon further into the future could improve the overall strategy by allowing the algorithm to delay the revisitation action for more opportunistic revisits. In the current PDN framework, the saliency value for not-yet-mapped areas is linearly interpolated in our calculation of expected information gain. In order to extend the look-ahead horizon beyond one-step, the algorithm must be modified to predict the feature distribution and measurement likelihood into yet-unknown areas of the horizon. Machine learning (Rasmussen and Williams, 2005), sparse representation (Rubinstein et al., 2010), and compressive sensing (Baraniuk, 2007) approaches shed some light toward this direction of study.

Acknowledgments

This work was supported by the Office of Naval Research under grants N00014-12-1-0092 and N00014-07-1-0791, monitored by Dr. T. Swean and T. Kick. We would like to thank J. Vaganay and K. Shurn of Bluefin Robotics for their exceptional support during our 2011 field experiments.

References

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

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

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

Baraniuk, R. (2007). Compressive sensing. *IEEE Signal Process. Mag.*, pages 118–120.

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

Bourgault, F., Makarenko, A. A., Williams, S. B., Grocholsky, B., and Durrant-Whyte, H. F. (2002). Information based adaptive robotic exploration. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, pages 540–545.

Bryson, M. and Sukkarieh, S. (2005). An information-theoretic approach to autonomous navigation and guidance of an uninhabited aerial vehicle in unknown environments. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, pages 3770–3775.

Carrillo, H., Reid, I., and Castellanos, J. A. (2012). On the comparison of uncertainty criteria for active SLAM. In *Proc. IEEE Int. Conf. Robot. and Automation*, pages 2080–2087.

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

Connolly, C. (1985). The determination of next best views. In *Proc. IEEE Int. Conf. Robot. and Automation*, volume 2, pages 432–435.

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

Davison, A. J., Reid, I. D., Molton, N. D., and Stasse, O. (2007). MonoSLAM: Real-time single camera SLAM. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29:1052–1067.

Du, J., Carlone, L., Kaouk Ng, M., Bona, B., and Indri, M. (2011). A comparative study on active SLAM and autonomous exploration with particle filters. In *IEEE/ASME Int. Conf. on Advanced Intell. Mechatronics*, pages 916–923.

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

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

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

Feder, H. J. S., Leonard, J. J., and Smith, C. M. (1999). Adaptive mobile robot navigation and mapping. *Int. J. Robot. Res.*, 18(7):650–668.

Frintrop, S. and Jensfelt, P. (2008). Active gaze control for attentional visual SLAM. In *Proc. IEEE Int. Conf. Robot. and Automation*, pages 3690–3697.

Gonzalez-Banos, H. H. and Latombe, J.-C. (2002). Navigation strategies for exploring indoor environments. *Int. J. Robot. Res.*, 21(10–11):829–848.

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

Hover, F. S., Eustice, R. M., Kim, A., Englot, B., Johannsson, H., Kaess, M., and Leonard, J. J. (2012). Advanced perception, navigation and planning for autonomous in-water ship hull inspection. *Int. J. Robot. Res.*, 31(12):1445–1464.

Huang, S., Kwok, N. M., Dissanayake, G., Ha, Q. P., and

Fang, G. (2005). Multi-step look-ahead trajectory planning in SLAM: Possibility and necessity. In *Proc. IEEE Int. Conf. Robot. and Automation*, pages 1091–1096.

Julian, B. J., Angermann, M., Schwager, M., and Rus, D. (2012). Distributed robotic sensor networks: An information theoretic approach. *Int. J. Robot. Res.*, 31(10):1134–1154.

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

Kaess, M. and Dellaert, F. (2009). Covariance recovery from a square root information matrix for data association. *Robot. and Auton. Syst.*, 57(12):1198–1210.

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

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

Kavraki, L., Svestka, P., Latombe, J., and Overmars, M. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robot. Autom.*, 12:566–580.

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

Kiefer, J. (1974). General equivalence theory for optimum designs (approximate theory). *The Annals of Statistics*, 2(5):849–879.

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.

Kim, A. and Eustice, R. M. (2011). Combined visually and geometrically informative link hypothesis for pose-graph visual SLAM using bag-of-words. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, pages 1647–1654, San Francisco, CA.

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

Kim, A. and Eustice, R. M. (2013a). Perception-driven navigation: Active visual SLAM for robotic area coverage. In *Proc. IEEE Int. Conf. Robot. and Automation*, pages 3181–3188, Karlsruhe, Germany.

Kim, A. and Eustice, R. M. (2013b). Real-time visual SLAM for autonomous underwater hull inspection using visual saliency. *IEEE Trans. Robot.*, 29(3):719–733.

Kollar, T. and Roy, N. (2008). Trajectory optimization using reinforcement learning for map exploration. *Int. J. Robot. Res.*, 27(2):175–196.

LaValle, S. M. and Kuffner, J. J. (1999). Randomized kinodynamic planning. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, volume 1, pages 473–479.

Levine, D. S. (2010). Information-rich path planning under general constraints using rapidly-exploring random trees. Master's thesis, Massachusetts Institute of Technology, Department of Aeronautics and Astronautics, Cambridge MA.

Li, W. and Cassandras, C. G. (2005). Distributed cooperative coverage control of sensor networks. In *Proc. IEEE Conf. Decision Control*, pages 2542–2547.

Makarenko, A. A., Williams, S. B., Bourgault, F., and Durrant-Whyte, H. F. (2002). An experiment in integrated exploration. In *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, pages 534–539.

Prentice, S. and Roy, N. (2009). The belief roadmap: Efficient planning in linear POMDPs by factoring the covariance. *Int. J. Robot. Res.*, 28(11-12):1448–1465.

Rasmussen, C. E. and Williams, C. K. I. (2005). *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press.

Rubinstein, R., Bruckstein, A., and Elad, M. (2010). Dictionaries for sparse representation modeling. *Proceedings of the IEEE*, 98(6):1045–1057.

Russell, S. J. and Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*. Pearson Education.

Sim, R. (2005). Stable exploration for bearings-only SLAM. In *Proc. IEEE Int. Conf. Robot. and Automation*, pages 2411–2416.

Sim, R. and Roy, N. (2005). Global A-optimal robot exploration in SLAM. In *Proc. IEEE Int. Conf. Robot. and Automation*, pages 661–666, Barcelona, Spain.

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

Stachniss, C., Grisetti, G., and Burgard, W. (2005). Information gain-based exploration using rao-blackwellized particle filters. In *Proc. Robot.: Sci. & Syst. Conf.*, Cambridge, MA, USA.

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.*, pages 1–14, Durham, NH.

Valencia, R., Andrade-Cetto, J., and Porta, J. M. (2011). Path planning in belief space with pose SLAM. In *Proc. IEEE Int. Conf. Robot. and Automation*, pages 78–83.

Whaite, P. and Ferrie, F. P. (1997). Autonomous exploration: Driven by uncertainty. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(3):193–205.

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	Hybrid simulation of perception-driven navigation (PDN) using the *SS Curtiss* 2011 dataset.
2	Video	Real-world PDN implementation on the *SS Curtiss* 2013.