Opportunistic Sampling-based Planning for Active Visual SLAM

Stephen M. Chaves, Ayoung Kim, and Ryan M. Eustice

***Abstract*— This paper reports on an active visual SLAM path planning algorithm that plans loop-closure paths in order to decrease visual navigation uncertainty. Loop-closing revisit actions bound the robot's uncertainty but also contribute to redundant area coverage and increased path length. We propose an opportunistic path planner that leverages sampling-based techniques and information filtering for planning revisit paths that are coverage efficient. Our algorithm employs Gaussian Process regression for modeling the prediction of camera registrations and uses a two-step optimization for selecting revisit actions. We show that the proposed method outperforms existing solutions for bounding navigation uncertainty with a hybrid simulation experiment using a real-world dataset collected by a ship hull inspection robot.**

I. Introduction

In order to perform tasks like exploration, search and rescue, reconnaissance, and target tracking, autonomous mobile robots must solve three main subproblems: localization, mapping, and planning. While the simultaneous localization and mapping (SLAM) problem has been well-studied in the past decades, research toward integrating planning and SLAM into a comprehensive framework is still in its early years. Typically, SLAM solutions assume that a path is given ahead of time. Likewise, path planning algorithms assume that accurate localization is achievable and at least partial information about the environment is known. Yet, real-world robots operate at the intersection of these three areas where the assumptions about prior information are not always true. Real-world robotic systems demand an all-encompassing architecture that localizes, maps, and plans concurrently.

This paper focuses on integrating path planning into the SLAM framework, an area of research often called *active* SLAM. We propose a path planning algorithm for finding and deciding revisit actions in the SLAM formulation that are useful for closing loops and driving down the robot's navigation uncertainty. We consider a robot with the goal of efficiently surveying an *a priori* unknown target environment subject to a localization uncertainty threshold. Exploring new areas of the environment tends to grow navigation uncertainty, so the robot must perform loop-closure revisit actions to decrease uncertainty while accounting for overall path length.

*This work was supported by the Office of Naval Research under award N00014-12-1-0092; S. Chaves was supported by The SMART Scholarship for Service Program by the Department of Defense.

S. Chaves is with the Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI 48109, USA schaves@umich.edu.

A. Kim is with Electronics and Telecommunications Research Institute (ETRI), Daejun, S. Korea ayoungk@etri.re.kr.

R. Eustice is with the Department of Naval Architecture & Marine Engineering, University of Michigan, Ann Arbor, MI 48109, USA eustice@umich.edu.

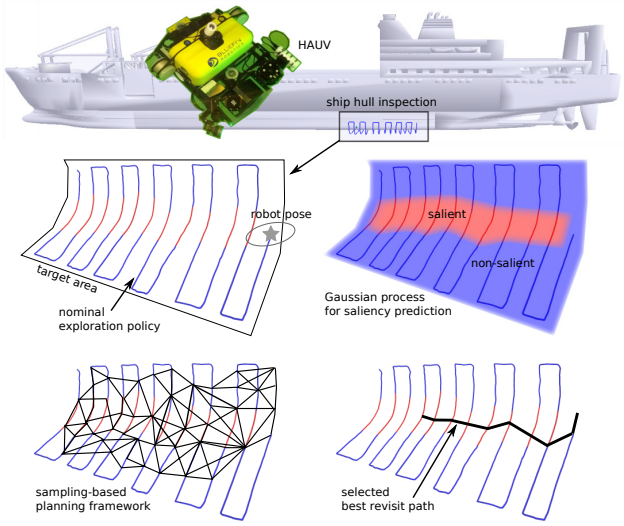


Fig. 1: Gist of the proposed path planning algorithm. Given a target area and nominal exploration policy, the robot explores the environment subject to an acceptable navigation uncertainty. We use GP regression to predict the visual saliency of the environment and a sampling-based planning framework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

With the integrated path planning algorithm proposed in this paper, we

1) employ Gaussian Process (GP) regression for accurate prediction and probabilistic modeling of the loop-closure utility of unmapped areas,
2) combine sampling-based planning with information filtering to efficiently search hundreds of paths through the environment for their expected utility, and
3) develop a two-step optimization process for selecting revisit actions that yields an opportunistic planning framework.

An illustration of the proposed algorithm is given in Fig. 1. We evaluate our proposed planning method in a hybrid simulation using a real-world dataset from a ship hull inspection robot, using both synthetic and real camera imagery, and show that it outperforms other methods.

II. Related Work

Research in the area of active SLAM stems from the seminal works of [1]–[3] and generally focuses on finding control actions to reduce the uncertainty or maximize the information of the SLAM posterior. While some of these approaches do not include a measure of path length in the objective function [4], [5], others focus on reducing uncertainty while performing efficient exploration [6]–[8]. Our

formulation is similar in that we minimize an objective that considers both path length and robot navigation uncertainty, but we additionally consider the plausibility of expected camera measurements when evaluating the utility of a loop-closure path. Valencia et al. [9]–[11] presented an active SLAM framework that is very 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.

Sampling-based planning originates from traditional path planners like the Rapidly-exploring Random Tree (RRT) and Probabilistic Roadmap (PRM) [12], but was only recently extended to include a notion of uncertainty during the planning process [13]. Karaman and Frazzoli [14] 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 [15] extended the RRG even further to include stochasticity of measurements and robot dynamics, a formulation generalized by [16] to accomplish information gathering. We take a similar approach and adopt the use of the RRG, but present a proposed algorithm integrated with SLAM.

In order to predict the likelihood of making camera measurements over unmapped areas, we use GP regression to model a measure of the loop-closure utility of the environment. Recently, GPs have been widely used in learning spatial distributions and predicting sensor data. In [17], the authors accurately predict an occupancy map using a GP with a trained neural network kernel, despite dealing with noisy and sparse sensor information. For underwater applications like our own, [18] used a GP to predict bathymetric data in unmapped patches in order to perform SLAM without any actual sensor overlap. They implemented the GP over a large dataset by selecting a sparse kernel covariance function [19].

Much of the work in this paper extends the perception-driven navigation (PDN) method for active visual SLAM presented earlier by Kim and Eustice [20]. The PDN approach considers the same problem formulation, however it plans revisit paths by clustering salient nodes in the SLAM pose-graph to form waypoints, planning a point-to-point path to each waypoint, and computing a reward for revisiting along each path. In contrast, the proposed algorithm reported here improves upon this restriction by removing the need for clustering while evaluating many more paths with a more principled approach based upon GP regression and sampling-based planning techniques.

III. Problem Formulation

We consider a robot performing pose-graph SLAM to survey an *a priori* unknown environment, subject to an acceptable uncertainty threshold. The robot is equipped with proprioceptive sensors to perform dead-reckoning and exteroceptive sensors for perception, which define the robot's

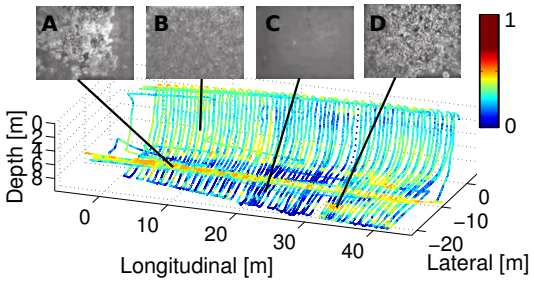


Fig. 2: Representative sample images captured on the below-water portion of the *SS Curtiss*' ship hull. The underwater feature distribution is highly diverse, ranging from feature-less (B,C) to feature-rich (A,D) images. The nodes are color-coded by their local saliency score, S_L.

ability to survey the environment and gather loop-closure measurements. Given this description, we are interested in performing an underwater visual inspection of a large ship hull with a Hovering Autonomous Underwater Vehicle (HAUV) [21]. Since the HAUV is capable of hovering, we assume holonomic kinematic constraints. The HAUV uses a Doppler velocity log (DVL) and depth sensor for dead-reckoning and is equipped with an underwater camera for capturing images.

The problem is formulated with the following assumptions and definitions:

1) The boundaries of the target coverage area are given.
2) A nominal exploration policy covers the target area in efficient time (vertical tracklines for the HAUV, see Fig. 1).
3) The user defines a navigation uncertainty threshold.
4) No other prior information on the environment is provided. Planning is performed online in conjunction with SLAM.

We adopt the incremental smoothing and mapping (iSAM) algorithm by Kaess et al. [22], [23] 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 [21]. For planning, we use a linearized snapshot of the system in the form of an extended information filter (EIF). The EIF allows for accurate prediction without relinearizing since the expected measurements occur at the mean of the distribution.

Before we describe algorithmically how the proposed method finds candidate revisit paths, we explain in the next section the mathematical details pertaining to how revisit paths are evaluated.

IV. Revisit Path Utility

The utility of a revisit path in our active SLAM formulation depends highly on the expected loop-closures. We use registrations between overlapping camera images (via a homography mapping or the Essential matrix) to close loops in the SLAM pose-graph, resulting in revisit paths that must 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 visually 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 [24], [25], including *local saliency* (S_L). 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 future poses along not-yet-traveled revisit paths. Thus, estimating successful registrations hinges on accurate saliency prediction.

Previously, PDN used a simple interpolation/extrapolation 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 framework that accounts for the spatial distribution of visual features and yields distributions of the saliency values to include in the planning formulation.

A. Saliency Prediction using Gaussian Process Regression

The objective for 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 $\mathcal{D} = \{\mathrm{X}, \mathbf{y}\}$ 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, $\mathbf{y} = \{S_{L_i}\}_{i=1}^n$. Similar to Barkby et al. [18], we use a simple stationary covariance function suitable for large scale data [19]:

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

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 $W = \mathrm{diag}(1/\Delta_d^2, 1/\Delta_h^2, 1)$, where Δ_d is the distance between two along-track SLAM poses and Δ_h is the distance between two cross-track SLAM poses.

Fig. 3 shows the GP prediction performance over unmapped areas. Fig. 3(a) depicts the robot following the nominal exploration policy. We use the GP to predict saliency scores both in the future—throughout the remaining survey area, and in the past—in unmapped areas between previously-traveled tracklines. Fig. 3(b) shows the associated variance for the predictions, where the variance increases as the predictions transition to farther-out areas covered by the nominal exploration policy. Between past tracklines, the saliency prediction closely matches the actual value with small variance, shown closer in Fig. 3(c). This zoomed view also shows that the GP prediction works as well as 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.

B. Evaluation of a Revisit Path

For every candidate revisit path, $\mathcal{P}$, we use an exactly sparse delayed-state filter (ESDF) [26] to construct an EIF 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 two sources of delta information corresponding to the the expected odometry and camera registrations available along the 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, $\boldsymbol{\mu}_{\mathtt{path}} = \boldsymbol{\mu}_{\mathtt{slam}} \cup \{\mathbf{x}_{v_0}, \cdots, \mathbf{x}_{v_{p-1}}\}$, and the posterior information, given by

$$\Lambda_{\mathtt{path}} = \Lambda_{\mathtt{slam}} + \Lambda_{\mathtt{odo}} + \Lambda_{\mathtt{cam}}. \tag{1}$$

The odometry delta information is block-tridiagonal and found as the sum of relative-pose increments between sequential virtual poses in the path, $(\mathbf{x}_{v_i}, \mathbf{x}_{v_{i+1}}) \in \mathcal{P}$:

$$\Lambda_{\mathtt{odo}} = \sum_{i=0}^{p-1} \mathrm{H}_{\mathtt{odo}_{i,i+1}}^\top \mathrm{Q}_{\mathtt{odo}_{i,i+1}}^{-1} \mathrm{H}_{\mathtt{odo}_{i,i+1}}, \tag{2}$$

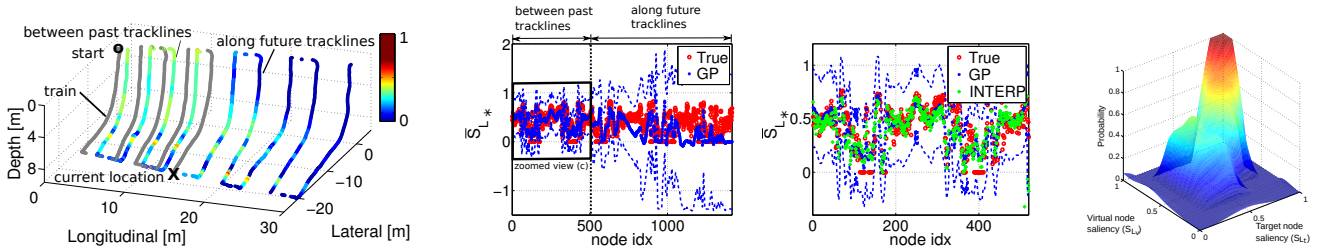
where $\mathrm{H}_{\mathtt{odo}_{i,i+1}}$ is the sparse Jacobian [27] and $\mathrm{Q}_{\mathtt{odo}_{i,i+1}}$ is the noise.

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 [24] for proposing camera registration link hypotheses between a virtual pose on a path and up to n_{plink} existing target poses in the SLAM pose-graph that may contain spatially-overlapping image views. Each link hypothesis has an empirical probability of successful registration, P_L, computed as a function of target pose saliency and virtual pose saliency for overlapping pairs, $P_L = g(S_{L_t}, S_{L_v})$, shown in Fig. 3(d) and described further in [20]. The GP prediction returns a distribution of virtual pose saliency scores with pdf $f(S_{L_v})$, which we transform into the censored pdf $f'(S_{L_v})$ [28], with mean values between 0 and 1, in order to compute the expected probability of the link as

$$P_L = \int_0^1 g(S_{L_t}, S_{L_v}) f'(S_{L_v}) dS_{L_v}. \tag{3}$$

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

$$\Lambda_{\mathtt{cam}} = \sum_{i=0}^{p-1} \sum_{t \in \mathcal{L}_i} P_L \cdot \mathrm{H}_{\mathtt{cam}_{t,i}}^\top \mathrm{R}^{-1} \mathrm{H}_{\mathtt{cam}_{t,i}}, \tag{4}$$



(a) GP predicted saliency map (b) Variance (c) Zoomed view (d) Link probability (P_L)

Fig. 3: (a) The predicted saliency map from the GP. 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. (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. (d) The empirical probability of the camera registration to be successful, used in (3). The probability is a function of target node saliency (S_{L_t}) and virtual node saliency (S_{L_v}).

where $\mathrm{H}_{\mathtt{cam}_{t,i}}$ is the camera measurement Jacobian [29], R is the camera measurement noise (assumed constant for convenience), and $\mathcal{L}_i$ is the index set of camera registrations associated with virtual pose $\mathbf{x}_{v_i}$. 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.

The cost function we use to determine the utility of a 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

$$\mathcal{C}(\mathcal{P}) = \alpha \cdot \frac{\mathcal{U}(\mathcal{P}) - \mathcal{U}_{\mathtt{lower}}}{\mathcal{U}_{\mathtt{upper}} - \mathcal{U}_{\mathtt{lower}}} + (1-\alpha) \cdot \frac{d(\mathcal{P})}{d_{\mathtt{upper}}}, \quad (5)$$

where $\mathcal{U}(\mathcal{P}) = h(\Sigma_{nn})$ is a function of the terminating pose covariance of a path and used as a measure of navigation uncertainty. Both the A-optimal trace and the D-optimal determinant can be used for $h(\cdot)$ [4]. The uncertainty upper bound ($\mathcal{U}_{\mathtt{upper}}$) is specified by the user as the acceptable threshold for the nominal exploration policy, and the lower bound ($\mathcal{U}_{\mathtt{lower}}$) represents an uncertainty that the user is willing to accept as near the best-possible. $d(\mathcal{P})$ is the revisit (redundant coverage) distance of the path and $d_{\mathtt{upper}}$ 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 [30], we recover the terminating covariance of a revisit path, Σ_{nn}, using $\Lambda_{\mathtt{path}}\Sigma_{*n} = \mathrm{I}_{*n}$, 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.

V. Algorithm Description

We now discuss algorithmically how the proposed method searches for candidate revisit paths. We leverage a sampling-based approach to quickly explore the configuration space of the robot while evaluating candidate paths using the formulation from §IV. The planning algorithm is outlined in Algorithm 1 and described in detail below.

A. Graph Construction

The underlying structure for the path planner is the RRG [14]–[16], which incrementally constructs 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 $\mathcal{P}_i$) 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, $\boldsymbol{\mu}_{\mathtt{path}}$, and associated information, $\Lambda_{\mathtt{path}}$, from (1). The virtual poses in the mean vector, $\{\mathbf{x}_{v_0}, \cdots, \mathbf{x}_{v_{p-1}}\}$, arise from traveling along edges, and the information matrix $\Lambda_{\mathtt{path}}$ is the sum of the SLAM information, $\Lambda_{\mathtt{slam}}$, and the delta information gathered along the way. Fig. 4(a) 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 two sources of delta information from (1) 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, (1) can be rewritten as

$$\Lambda_{\mathtt{path}} = \Lambda_{\mathtt{slam}} + \Lambda^1_{edge} + \ldots + \Lambda^k_{edge}, \quad (6)$$

where

$$\Lambda^i_{edge} = \Lambda^i_{\mathtt{odo}} + \Lambda^i_{\mathtt{cam}}, \quad (7)$$

and Λ^i_{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, Λ^i_{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 [11] and is analogous to the one-step transfer function used by the Belief Roadmap (BRM) [13].

When an edge is traversed, however, 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-degree of freedom (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 further increase the benefit of the information form parameterization, we can significantly reduce the dimensionality of the delta information of an edge during its construction by using marginalization.

Consider the example edge shown in Fig. 4(b); rather than explicitly representing all the factors along the edge, we condense their combined delta information (Λ^i_{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 the marginalized delta information, Λ^i_{marg}, which replaces Λ^i_{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, and visualized in Fig. 4(b). Now, (6) takes the final form,

$$\Lambda_{\texttt{path}} = \Lambda_{\texttt{slam}} + \Lambda^1_{marg} + \ldots + \Lambda^k_{marg}. \tag{8}$$

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. Edge construction is accomplished within the `Connect()` function of the proposed algorithm.

B. 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 [15], [16], but we are willing to employ a more aggressive heuristic that sacrifices optimality for a large increase in speed, as done in [16] for submodular objective functions. Thus, we maintain a partial ordering of candidates at each vertex according to the distance and uncertainty metrics of (5):

$$\mathcal{P}_a > \mathcal{P}_b \Leftrightarrow d(\mathcal{P}_a) < d(\mathcal{P}_b) \wedge \mathcal{U}(\mathcal{P}_a) < \mathcal{U}(\mathcal{P}_b) + \epsilon, \tag{9}$$

where ϵ is a small factor to aid in pruning when the candidates are quite similar [15]. When (9) is true, candidate $\mathcal{P}_b$ has both a longer revisit distance and a higher uncertainty than $\mathcal{P}_a$, so $\mathcal{P}_b$ 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 $\mathcal{P}*$
Initialize: vertices $V = \{v_0\}$, edges $E = \{\}$, queue $Q = \{\}$

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

Function: $\texttt{ExtendToNearest}(\mathbf{x}_i)$

$v_{nearest} = \texttt{FindNearestVertex}(\mathbf{x}_i)$
$v_{new} = \texttt{SteerToward}(\mathbf{x}_i, v_{nearest})$
$V = V \cup v_{new}$
$E = E \cup \{\texttt{Connect}(v_{nearest}, v_{new}), \texttt{Connect}(v_{new}, v_{nearest})\}$
$Q = Q \cup \{\text{all } \mathcal{P} \text{ at } v_{nearest}\}$
return v_{new}

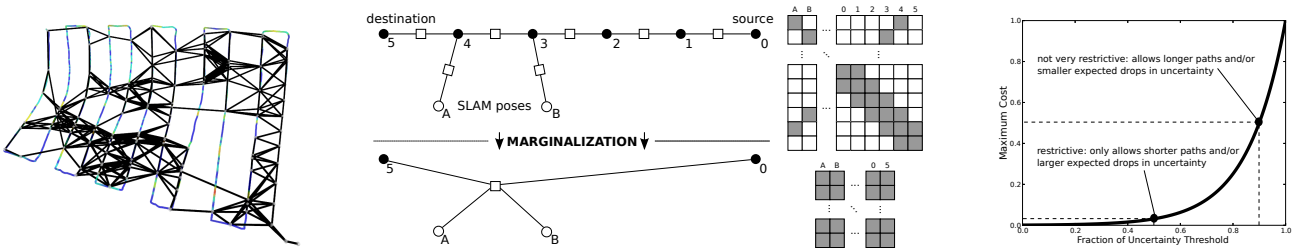
Function: `ProcessQueue()`

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

C. Opportunistic Look-Ahead Planning

The planning algorithm is integrated with the visual SLAM framework in order to provide online decision-making for the robot. The planning process is not triggered by breaching an uncertainty measure as in [20]; rather, the planner is initialized with the SLAM process and runs in parallel for the duration of the mission. To this end, we design a planning framework that is opportunistic in nature, such that the selection of a revisit action is based on its expected benefit, not on a discrete triggering event. Thus, we develop a two-step optimization for the planner 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* of the next *m* steps from the nominal exploration policy



(a) Planning roadmap (b) Edge construction (c) Maximum cost curve

Fig. 4: (a) An example roadmap constructed by the proposed planning algorithm. (b) The factor graph and delta information matrix arising from an edge before (top, Λ_{edge}) and after marginalization (bottom, Λ_{marg}) during construction. (c) The curve used to determine the maximum allowable cost of a candidate revisit path during the planning process.

in order to put an upper bound ($\mathcal{C}_{max}$) on the cost of a selected revisit path, and

2) we search for the candidate path with minimum cost that is less than this upper bound.

We incorporate the look-ahead horizon by adding the poses of the next *m* steps from the nominal exploration policy as vertices in the RRG and designating each as a valid goal point for the planner. (We use $m = 100$, corresponding to a 20 m horizon in our case.) Incorporation of the horizon gives the planner a sense of the expected measurements and predicted uncertainty, $\mathcal{U}_{\texttt{exp}}$, for exploring with the nominal policy. We query the planner for $\mathcal{U}_{\texttt{exp}}$ in order to compute the maximum acceptable cost for a candidate that diverts within the look-ahead horizon, given by

$$\mathcal{C}_{max} = \beta^{(\frac{\mathcal{U}_{\texttt{exp}}}{\mathcal{U}_{\texttt{upper}}} - 1)}, \tag{10}$$

where β is selected by the user. (We use $\beta = 1000$.) This exponential function yields a higher allowable cost as the uncertainty $\mathcal{U}_{\texttt{exp}}$ increases (see Fig. 4(c)). Instituting this upper bound is the first step in the optimization, accomplished within `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 $\mathcal{C}_{max}$, performed by the `UpdateBestPath()` function. If no revisit paths are found with cost below $\mathcal{C}_{max}$, the exploration action of traveling along the look-ahead steps is selected, provided this action does not exceed the uncertainty threshold $\mathcal{U}_{\texttt{upper}}$.

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

D. Comments on Computation

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 [14], although we include the additional computation required to predict the measurements available along each edge. Similarly, [16] presents the complexity of propagating candidates throughout the RRG, which is exponential in the worst case but much better for aggressive pruning strategies like the one we propose. Evaluation of the cost of a candidate using the Cholesky decomposition is generally $\mathcal{O}(n^3)$, but lower with methods for sparse systems. Other techniques can help reduce overall computation time, like the proposed edge marginalization step, biased sampling toward salient regions (which we also employ), or adjusting parameters related to RRG connectivity and pruning aggressiveness. Additionally, the incremental sampling-based nature of the algorithm finds coarse solutions quickly but improves the best path as computation time remains. Still, reducing planning time remains a large area of future work.

We can achieve online operation in parallel with SLAM in our application by pausing the HAUV at times during the inspection to plan over the upcoming look-ahead horizon. For 50 calls to the planner during experiments like those in §VI, the average planning time was 54.4 s. The average time for a single iteration within the algorithm was 1.25 s. Propagation and evaluation of a candidate path over an edge averaged 33.2 ms.

VI. Results

We tested the proposed planning algorithm with a hybrid simulation using data collected during a ship hull inspection of the *SS Curtiss* by the HAUV in February 2011. More information about the experimental setup and this dataset can be found in [20]. See also the accompanying video attachment to this paper for visualization.

A. Synthetic Saliency Map

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 in simulation. We compare the proposed planning algorithm against three other planning methods: an open-loop survey with no revisit actions, a preplanned deterministic strategy with revisit actions along every trackline, and PDN. 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 loop-closures and which revisit paths to take.

TABLE I: Summarized Results

METHOD	PATH LENGTH	AVG. UNCERTAINTY
Open-loop	386.5 [m]	121.8 [% of $\mathcal{U}_{\texttt{upper}}$]
SYNTHETIC SALIENCY		
DET (non-sal)	885.2	151.7
DET (sal)	876.2	33.4
PDN	583.3	71.3
Proposed	524.5	42.5
REAL IMAGERY		
DET	865.8	64.3
PDN	527.2	67.2
Proposed	557.3	53.7

Results from the synthetic saliency simulation are shown in Fig. 5 and Fig. 6(a). While the path length of the deterministic method is inherently long, its performance at bounding the uncertainty of the robot is completely dependent on whether the preplanned revisit paths travel a salient region. To this end, 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 hence 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 triggering the planning process. 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 all the hybrid simulations are presented in Table I.

B. Real Image Data

Here we present results using the hybrid simulation with the real imagery collected during the dive. Images from the dataset densely cover the entire target environment of the *SS Curtiss*, as shown in Fig. 2. We use the same exploration policy as the previous simulation but instead use the real camera images recorded by the robot to calculate saliency scores and make loop-closure registrations. The proposed algorithm is again compared to the three other planning methods. Results are presented in Fig. 5 and Fig. 6(b).

This time, the deterministic revisits happen to 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.) 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. Refer to Table I for a summary of results.

VII. CONCLUSION

In this paper, we proposed a path planning algorithm integrated with visual SLAM in order to find and execute loop-closure revisit paths 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. We demonstrated the proposed method using a hybrid simulation with both synthetic and real camera imagery and showed that it outperforms other representative SLAM planning methods.

REFERENCES

[1] R. Bajcsy, "Active perception," *Proc. of the IEEE*, vol. 76, no. 8, pp. 996–1005, 1988.

[2] P. Whaite and F. Ferrie, "Autonomous exploration: Driven by uncertainty," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 3, pp. 193–205, 1997.

[3] H. J. S. Feder, J. J. Leonard, and C. M. Smith, "Adaptive mobile robot navigation and mapping," *Int. J. Robot. Res.*, vol. 18, no. 7, pp. 650–668, 1999.

[4] R. Sim and N. Roy, "Global A-optimal robot exploration in SLAM," in *Proc. IEEE Int. Conf. Robot. and Automation*, Barcelona, Spain, 2005, pp. 661–666.

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

[6] H. H. Gonzalez-Banos and J.-C. Latombe, "Navigation strategies for exploring indoor environments," *Int. J. Robot. Res.*, vol. 21, no. 10–11, pp. 829–848, 2002.

[7] F. Bourgault, A. A. Makarenko, S. B. Williams, B. Grocholsky, and H. F. Durrant-Whyte, "Information based adaptive robotic exploration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, Lausanne, Switzerland, 2002, pp. 540–545.

[8] C. Stachniss, G. Grisetti, and W. Burgard, "Information gain-based exploration using Rao-Blackwellized particle filters," in *Proc. Robot.: Sci. & Syst. Conf.*, Cambridge, MA, USA, 2005.

[9] R. Valencia, J. Andrade-Cetto, and J. Porta, "Path planning in belief space with pose SLAM," in *Proc. IEEE Int. Conf. Robot. and Automation*, Shanghai, China, May 2011, pp. 78–83.

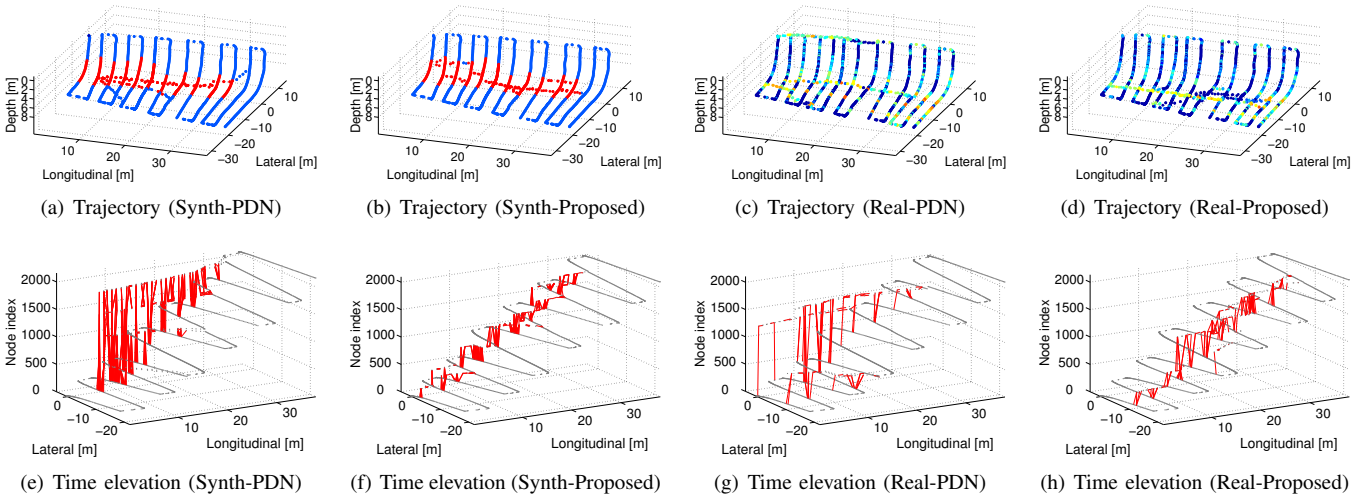
[10] R. Valencia, J. Miro, G. Dissanayake, and J. Andrade-Cetto, "Active pose SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, Vilamoura, Portugal, Oct 2012, pp. 1885–1891.

[11] R. Valencia, M. Morta, J. Andrade-Cetto, and J. Porta, "Planning reliable paths with pose SLAM," *IEEE Trans. Robot.*, vol. 29, no. 4, pp. 1050–1059, 2013.

[12] H. Choset, K. M. Lynch, S. Hutchinson, G. A. Kantor, W. Burgard, L. E. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*. Cambridge, MA: MIT Press, 2005.

[13] S. Prentice and N. Roy, "The belief roadmap: Efficient planning in belief space by factoring the covariance," *Int. J. Robot. Res.*, vol. 28, no. 11-12, pp. 1448–1465, 2009.

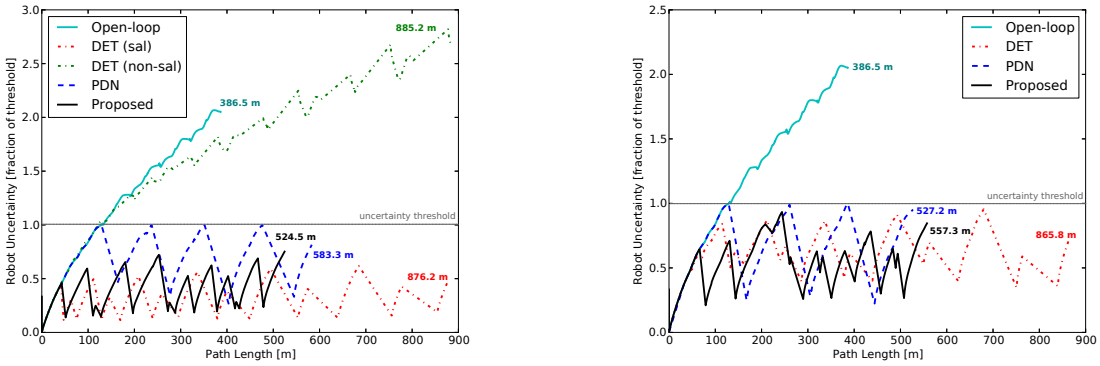
[14] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 846–894, 2011.



(a) Trajectory (Synth-PDN) (b) Trajectory (Synth-Proposed) (c) Trajectory (Real-PDN) (d) Trajectory (Real-Proposed)

(e) Time elevation (Synth-PDN) (f) Time elevation (Synth-Proposed) (g) Time elevation (Real-PDN) (h) Time elevation (Real-Proposed)

Fig. 5: The SLAM trajectories and associated time elevation graphs resulting from PDN and the proposed method in the hybrid simulation. Both the synthetic saliency tests (a) (b) (e) (f) and real imagery tests (c) (d) (g) (h) are shown.



(a) Synthetic saliency results (b) Real imagery results

Fig. 6: Plots of the uncertainty metric $\mathcal{U}(\mathcal{P})$, as a percentage of the threshold, along the nominal exploration trajectory versus path length for the hybrid simulation. Comparisons for the synthetic saliency tests (a) and real imagery tests (b) are presented for four methods: the open-loop survey (OPL), deterministic revisit actions (DET), PDN, and the proposed algorithm.

[15] A. Bry and N. Roy, "Rapidly-exploring random belief trees for motion planning under uncertainty," in *Proc. IEEE Int. Conf. Robot. and Automation*, Shanghai, China, May 2011, pp. 723–730.

[16] G. Hollinger and G. Sukhatme, "Sampling-based motion planning for robotic information gathering," in *Proc. Robot.: Sci. & Syst. Conf.*, Berlin, Germany, June 2013.

[17] S. T. O'Callaghan and F. T. Ramos, "Gaussian process occupancy maps," *Int. J. Robot. Res.*, vol. 31, no. 1, pp. 42–62, 2012.

[18] S. Barkby, S. B. Williams, O. Pizarro, and M. V. Jakuba, "Bathymetric SLAM with no map overlap using Gaussian processes," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, San Francisco, CA, 2011, pp. 1242–1248.

[19] A. Melkumyan and F. Ramos, "A sparse covariance function for exact Gaussian process inference in large datasets," in *Proc. Int. Joint Conf. Artif. Intell.*, Pasadena, CA, 2009, pp. 1936–1942.

[20] A. Kim and R. M. Eustice, "Perception-driven navigation: Active visual SLAM for robotic area coverage," in *Proc. IEEE Int. Conf. Robot. and Automation*, Karlsruhe, Germany, May 2013, pp. 3181–3188.

[21] F. S. Hover, R. M. Eustice, A. Kim, B. Englot, H. Johannsson, M. Kaess, and J. J. Leonard, "Advanced perception, navigation and planning for autonomous in-water ship hull inspection," *Int. J. Robot. Res.*, vol. 31, no. 12, pp. 1445–1464, 2012.

[22] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Incremental smoothing and mapping," *IEEE Trans. Robot.*, vol. 24, no. 6, pp. 1365–1378, 2008.

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

[24] A. Kim and R. M. Eustice, "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.*, San Francisco, CA, Sept. 2011, pp. 1647–1654.

[25] ——, "Real-time visual SLAM for autonomous underwater hull inspection using visual saliency," *IEEE Trans. Robot.*, vol. 29, no. 3, pp. 719–733, 2013.

[26] R. M. Eustice, H. Singh, and J. J. Leonard, "Exactly sparse delayed-state filters for view-based SLAM," *IEEE Trans. Robot.*, vol. 22, no. 6, pp. 1100–1114, 2006.

[27] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," in *Autonomous Robot Vehicles*, I. Cox and G. Wilfong, Eds. Springer-Verlag, 1990, pp. 167–193.

[28] D. J. Hand, *Statistics: A Very Short Introduction*. Oxford: Oxford University Press, 2008.

[29] A. Kim and R. M. Eustice, "Pose-graph visual SLAM with geometric model selection for autonomous underwater ship hull inspection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, St. Louis, MO, Oct. 2009, pp. 1559–1565.

[30] R. M. Eustice, H. Singh, J. J. Leonard, and M. R. Walter, "Visually mapping the RMS Titanic: Conservative covariance estimates for SLAM information filters," *Int. J. Robot. Res.*, vol. 25, no. 12, pp. 1223–1242, 2006.