Next-Best-View Visual SLAM for Bounded-Error Area Coverage

Ayoung Kim and Ryan M. Eustice

***Abstract*—Navigating an unexplored environment using simultaneous localization and mapping (SLAM) requires that the robot's trajectory include revisit actions in order to produce loop-closure constraints; however, efficient area coverage requires that the robot's trajectory be minimally redundant in its path. This paper reports on a next-best-view SLAM algorithm that balances the trade-off between exploration and revisiting actions in order to simultaneously achieve efficient target area coverage and bounded-error navigation performance. Since area coverage efficiency and bounded localization performance represent competing objectives, the proposed algorithm computes the next-best control action required for localization and area coverage performance. The proposed algorithm, called perception-driven navigation (PDN), represents an integrated navigation solution to the robotic area coverage problem whereupon visual SLAM perception uncertainty is explicitly accounted for. Results are shown for simulated monocular visual SLAM trajectories representative of the type of area coverage problem encountered in autonomous underwater ship hull inspection.**

I. Introduction

This paper focuses on visual simultaneous localization and mapping (SLAM) for the robotic area coverage problem, where the goal is to achieve 100% coverage over a target area of interest while maintaining bounded-error navigation performance. In this problem, exploring to cover the given area versus revisiting to obtain visual loop-closures introduces two competing objectives.

In many typical cases, a robot carries out a survey of the target area following a preplanned trajectory (Fig. 1). This nominal trajectory provides efficient 100% area coverage, but implicitly assumes that visual SLAM navigation performance is uniformly adequate everywhere over the target area. This trajectory is preplanned in advance without knowledge of the actual visual feature distribution in the environment, even though this feature distribution significantly affects SLAM's ability to perform successful loop-closure.

To obtain useful loop-closures, revisit actions are needed. As illustrated in Fig. 1, during execution of the survey, when the uncertainty of the robot position increases, the robot needs to control itself for a revisit action because revisiting provides loop closures, which reduce the uncertainty in the robot's pose. This revisit action is typically preplanned or initiated by a human operator. However, using preplanned or human piloted revisit actions not only deteriorates the robot's autonomy, it can also be ineffective and/or inefficient because the actual feature distribution is not known in advance. To tackle this problem, we introduce perception-driven navigation (PDN)—an integrated navigation algorithm that automatically achieves efficient target area coverage while maintaining good visual SLAM navigation performance. PDN aims to provide an intelligent and fully autonomous online control scheme for efficient, bounded-error area-coverage that strikes a balance between revisit and exploration actions in a decision theoretic way.

This work was supported by the Office of Naval Research under awards N00014-07-1-0791 and N00014-12-1-0092.

A. Kim is with the Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI 48109, USA ayoungk@umich.edu

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

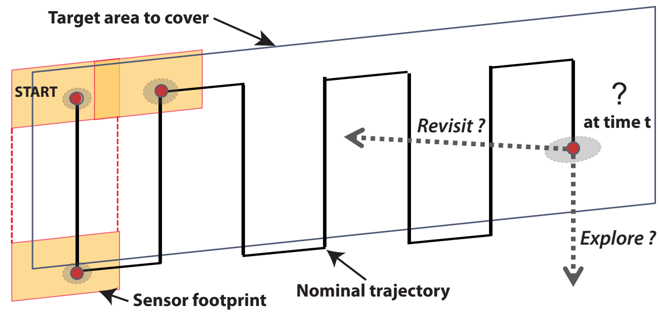


Fig. 1: Illustration of perception-driven navigation (PDN)—a nominal trajectory aims for efficient 100% coverage over the given target area and revisit detours are periodically made to achieve bounded-error SLAM navigation performance. The black line depicts the nominal preplanned coverage trajectory that the robot follows; shown in orange is the sensor footprint. The red dots indicate robot poses associated with image keyframes, and their 3-sigma uncertainty ellipsoids are overlaid. At any given time, t, PDN solves for the next-best-view that results in bounded navigation error performance while maintaining efficient survey coverage.

II. Related Work

The idea of actively controlling a robot to obtain loop-closures is not new. In fact, early seminal work in this area can be attributed to Bajcsy [1], who coined the term *active perception*, and was one of the first to note that control can improve the quality of sensor data. This line of thinking inspired a number of follow-on works in robotics. Feder et al. [2] developed an adaptive control framework using Fisher information (FI) to assess SLAM performance. Sim [3], and Sim and Roy [4], determined the next-best control action for SLAM performance also using FI. Davison et al. [5] used mutual information (MI) for gaze control in what they termed active SLAM.

A different, but related line of study has been in the area of point-to-point planning, where the map is known a priori and the robot chooses an optimal path through the map in terms of perceptual localization. Example works in this area include the Belief Roadmap (BRM) algorithm by Prentice and Roy [6], and planning in pose-graph belief space by Valencia et al. [7]. In these works, the robot undertakes what may be non-shortest distance paths through the map in order to achieve the lowest localization uncertainty. These studies, however, assume that the perception sensor provides uniformly good performance anywhere in the map. In the case of visual perception, this is often not true, as imagery is typically not uniformly good for multi-view registration in terms of its visual feature content.

The proposed perception-driven navigation algorithm represents a novel contribution in that it explicitly attempts to model the success of visual perception constraints when formulating the active SLAM exploration problem. In this regard, PDN is related to the next-best-view (NBV) problem in computer vision [8], which solves for the next best view of the scene in order to reveal the desired details of a model. Gonzalez-Banos and Latombe [9] proposed exploration strategies analogous to the NBV problem, but within a robotics context. Whaite and Ferrie [10] introduced an exploration algorithm that considered the uncertainty of a prior model, but made no attempt to build that model online. The most related work to PDN can be found in that of Makarenko et al. [11], Bourgault et al. [12], and Stachniss et al. [13], where integrated approaches to the online SLAM and exploration problem were also studied.

III. PERCEPTION-DRIVEN NAVIGATION (PDN)

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 (iSAM) algorithm by Kaess et al. [14], [15] as the SLAM back-end. In our application, constraints from odometry, monocular camera, attitude, and pressure depth are fused within iSAM [16].

Given the desired target area to cover and user defined allowable navigation uncertainty, PDN provides an intelligent solution to the area coverage planning problem while considering SLAM's navigation performance. As depicted in Fig. 2, PDN's calculated reward measures the utility of revisiting candidate waypoints for loop-closure versus continuing exploration for area coverage. By comparing the maximum reward for revisiting vs. exploring, the robot is able to choose the next best control step. The core steps within PDN consist of: *(i)* quantifying the scene's visual saliency, *(ii)* clustering salient keyframes into a set of candidate revisit waypoints, *(iii)* planning point-to-point paths for candidate revisit waypoints, *(iv)* computing rewards for revisiting candidate waypoints versus exploring actions, and *(v)* choosing the action that provides maximal reward.

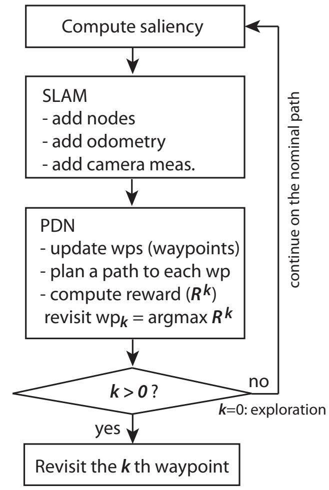
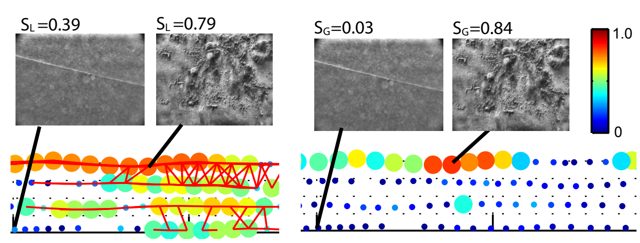


Fig. 2: Block-diagram of our integrated navigation framework.

A. Visual Saliency

In visual SLAM, not all images are equal in terms of their utility for keyframe registration. This is especially true in the underwater environment, where the spatial distribution of good visual features is not uniformly abundant. In this paper, we adopt two visual saliency metrics defined in [17]—*local saliency* (S_L) and *global saliency* (S_G)—to measure keyframe saliency for visual SLAM. Each measure provides a normalized score from 0 to 1, where 1 indicates highly salient imagery and 0 indicates non-salient imagery. Fig. 3 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 with respect to all others. In our application, the



(a) Local saliency map (b) Global saliency map

Fig. 3: A depiction of local and global saliency scores overlaid on an underwater visual SLAM hull inspection result. Nodes are depicted with dots and color-coded with respect to their saliency level, and enlarged for those with high saliency score ($S_L, S_G \geq 0.4$). Red lines between nodes indicate successful pairwise camera constraints used in the SLAM result; sample keyframes and their numerical saliency score are depicted above for a couple of the nodes. Local saliency (a) correlates well with where successful pairwise camera measurements occur, and global saliency (b) only reports high scores for rare scenes encountered in the environment.

robot measures these two saliency levels for every keyframe it inserts into the pose-graph; as it proceeds on a mission, it uses them in PDN's reward calculation.

B. Waypoint Generation

The complexity of PDN scales linearly with the number of candidate revisit nodes in the graph, called *waypoints*. These waypoints are automatically chosen and represent distinctive feature-rich regions in the environment with a high probability of loop-closure as measured by their visual saliency. Waypoint generation consists of two parts: salient node clustering and waypoint selection within each cluster. First, we threshold the keyframes in the pose-graph based upon their local saliency in order to generate a candidate set of visually informative nodes (in our work a threshold of $S_L^{\mathrm{wp}}=0.5$ is used). Locally salient nodes represent texture-rich scenes and, thus, identify feature-rich areas in the environment. Next, we cluster locally salient nodes into spatial neighborhoods. For this we use the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm [18], [19] to extract N_{wp} clusters. Finally, within each cluster we select a representative waypoint node by considering both its visual uniqueness (maximum global saliency level) and usefulness for loop-closure (lowest pose uncertainty). This process allows us to compute the N_{wp} best candidate waypoints in the graph for revisitation.

C. Path Generation

Given a set of candidate waypoints, we compute a point-to-point path to each waypoint over which the robot evaluates its expected reward for revisitation. Point-to-point paths are determined using a shortest distance criterion that allows for slight detours away from the shortest route in order to tour salient regions along the way. To do so, the global A* algorithm [20] is used and its heuristic criterion is modified to be weighted by local saliency. The weight is calculated as $2 - S_L$, which doubles the Euclidean distance to nodes of zero saliency while preserving the original distance to nodes of maximal saliency. Due to this weighting, the path may fall into local minima. To cope with this, we impose a perturbation action [21] by evaluating a pure Euclidean distance heuristic in the occurrence of local minima. Repeated bisection of nodes in the graph yields a sample trajectory that the robot can follow for waypoint revisitation. Fig. 6 depicts some example paths.

D. Reward for a Path

Reward for each candidate revisitation path is calculated considering both (*i*) the robot's navigation uncertainty and (*ii*) its area coverage performance. The expected round-trip terminating pose covariance and area coverage ratio are combined to compute the reward term, which the robot uses to decide the next-best-action.

1) Robot Uncertainty ($\mathcal{U}_{robot}$): For the robot uncertainty term, we evaluate the robot's expected pose covariance for the round-trip revisitation action versus the exploration action (Fig. 4). Given a path, the expected odometry information to be gained, Λ_{odo}, is computed by adding all expected odometry

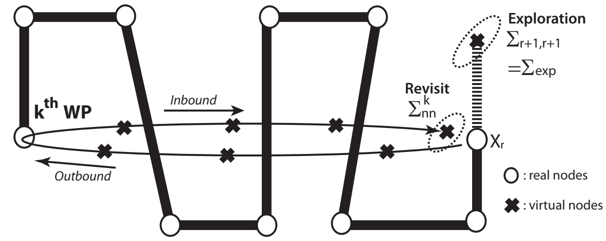


Fig. 4: An illustration of the round-trip expected covariance calculation used in PDN. The terminating uncertainties from revisiting and exploration are compared. Real nodes on the graph are shown as circles whereas virtual nodes along the candidate revisit path are marked with 'X'. Index r refers to the current robot node, which is also the last node in the existing pose-graph.

measurements for round-trip travel to the candidate waypoint along the revisit path:

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

Here, $\mathrm{H}_{\mathrm{odo}_{i,i+1}}$ is the Jacobian of the tail-to-tail operation [22] between nodes $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$, and $\mathrm{Q}_{i,i+1}$ is the odometry measurement noise covariance (in our case scaled with the travel distance between nodes).

For camera measurements, not all attempted pairwise keyframe registrations will result in success, which strongly depends upon the visual feature distribution in the environment. To account for this, we model the probability of a link to be successful, P_L, as a function of the local saliency scores associated with the two keyframes with the candidate link. The camera link event, L, is a Bernoulli random variable, and we model the probability of its success, P_L, using statistics from prior SLAM and saliency results. 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 an empirical model for the probability of a link to be successful as a function of these two saliency levels:

$$P_L = P_L(l = 1; S_{L_c}, S_{L_t}) \sim \text{Bernoulli}. \tag{2}$$

Based upon historical link proposal results from missions on previous vessels, we count the number of successful links versus the number of proposed links to obtain the empirical probability. To model the probability as a function of two saliency scores, all proposed links are divided into bins depending on their saliency levels, then the successful rate is evaluated within each bin to complete the modeling. Using this statistical model, the expected information gain for round-trip camera measurements is calculated as

$$\begin{aligned}\Lambda_{\texttt{cam}} = &\sum_{i=0}^{p-1}\sum_{m\in\mathcal{L}_i} P_L\cdot \mathrm{H}_{\texttt{cam}_{m,i}}^\top \mathrm{R}^{-1}\mathrm{H}_{\texttt{cam}_{m,i}} \quad \text{Outbound}\\ &+\sum_{i=p}^{1}\sum_{m\in\mathcal{L}_i} P_L\cdot \mathrm{H}_{\texttt{cam}_{m,i}}^\top \mathrm{R}^{-1}\mathrm{H}_{\texttt{cam}_{m,i}} \quad \text{Inbound,}\end{aligned} \tag{3}$$

where R is the camera measurement noise covariance[1], $\mathcal{L}_i$ is the set of candidates nodes in the graph for attempted registration with pose $\mathbf{x}_i$, $\mathrm{H}_{\texttt{cam}_{m,i}}$ is the Jacobian of the pairwise camera observation model between $\mathbf{x}_m$ and $\mathbf{x}_i$ [16], and P_L is the probability of success based upon local saliency.

Finally, to evaluate the expected round-trip terminating covariance, Σ_{nn}^k, for revisiting waypoint k, we first compute the expected cumulative delta information from odometry ($\Lambda_{\texttt{odo}}$) and camera constraints ($\Lambda_{\texttt{cam}}$), and then add them to the current snapshot of the SLAM information matrix (Λ_0). This results in the expected terminating information associated with revisitation ($\Lambda_{\texttt{pdn}}$):

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

From this, we recover the expected round-trip terminating pose covariance, Σ_{nn}^k, by solving for its covariance block-column as per [24] (which avoids inverting the entire information matrix $\Lambda_{\texttt{pdn}}$).

Next, we evaluate the expected terminating covariance for exploration, Σ_{exp}, by computing the one-step propagated covariance assuming that the previous odometry measurement holds for this time step too. Lastly, the reward term for robot uncertainty, $\mathcal{U}_{robot}^k$, is computed as the ratio of the localization uncertainty for the next-best-action to the user-defined allowable navigation uncertainty, Σ_{allow}. For the k^{th} waypoint, the robot uncertainty term is defined as

$$\begin{aligned}\mathcal{U}_{robot}^{k=0} &= \begin{cases} 0, & \text{if } \frac{|\Sigma_{\text{exp}}|}{|\Sigma_{\text{allow}}|} < 1 \\ \frac{|\Sigma_{\text{exp}}|^{\frac{1}{6}}}{|\Sigma_{\text{allow}}|^{\frac{1}{6}}}, & \text{otherwise}\end{cases}\\ \mathcal{U}_{robot}^{k>0} &= \frac{|\Sigma_{nn}^k|^{\frac{1}{6}}}{|\Sigma_{\text{allow}}|^{\frac{1}{6}}}, \quad k = 1,\cdots,N_{wp}\end{aligned} \tag{5}$$

where $k=0$ is the candidate exploration action, $k>0$ are the $1,\cdots,N_{wp}$ candidate revisit waypoints, and we have taken the 6^{th} root of the 6-DOF pose determinant in the numerator and denominator terms so that individually their units are $\mathrm{m}\cdot\mathrm{rad}$, which provides a more physically meaningful length scale for taking ratios.

When the expected exploration covariance is below the allowable covariance, the cost in the robot pose uncertainty term, $\mathcal{U}_{robot}^0$, becomes zero, leading the robot to pursue exploration. On the other hand, when the exploration covariance exceeds the allowable covariance, then the robot pose uncertainty term for exploration, $\mathcal{U}_{robot}^0$, is compared against all candidate revisit actions, $\mathcal{U}_{robot}^{k>0}$, which will be smaller when revisiting is likely to obtain enough loop-closures to overcome the increased navigation uncertainty from detouring. Unlike previous studies in active exploration [11]–[13], where the authors did not consider the actual likelihood of obtaining perceptual loop-closures, our approach introduces a realistic expectation in the reward calculation for the likelihood of camera loop-closures based upon visual saliency.

2) Area Coverage ($\mathcal{A}_{map}$): For the area coverage term, we evaluate the ratio of area-to-cover with respect to the target-coverage-area, where the target area is provided by the user. The area coverage term for the k^{th} waypoint is defined as

$$\mathcal{A}_{map}^k = \frac{\mathcal{A}_{\text{to\_cover}}}{\mathcal{A}_{\text{target}}} = \frac{\mathcal{A}_{\text{target}} - \mathcal{A}_{\text{covered}} + \mathcal{A}_{\text{redundant}}^k}{\mathcal{A}_{\text{target}}}, \tag{6}$$

where $\mathcal{A}_{\text{target}}$ is the predefined target area as set in the mission planning phase, $\mathcal{A}_{\text{covered}}$ is the amount of target area covered thus far, and $\mathcal{A}_{\text{redundant}}$ is the expected redundant area coverage produced by a revisiting action. This additional area is the result of multiplication of the revisit path with the sensor field of view width and has nonzero value (it is 0 for exploration).

3) Reward: Finally, to calculate the total reward, we need to combine the uncertainty and area reward terms. To do so, we introduce a weight, α, that controls the amount of emphasis placed on pose uncertainty versus area coverage:

$$\mathcal{C}^k = \alpha\cdot\mathcal{U}_{robot}^k + (1-\alpha)\cdot\mathcal{A}_{map}^k. \tag{7}$$

As defined, $\mathcal{C}^k$ represents a penalty term that we wish to minimize. In order to turn it into a reward, PDN uses the minus of this penalty

$$\mathcal{R}^k = -\mathcal{C}^k, \tag{8}$$

and selects the action with the largest reward, or in other words, the action with minimal penalty,

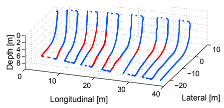
$$k^* = \text{argmax } \mathcal{R}^k = \text{argmin } \mathcal{C}^k. \tag{9}$$

The next-best-action is determined by choosing this maximal reward from the k candidate revisit waypoints $k \in \{0,1,2,\cdots N_{wp}\}$, where $k=0$ corresponds to the exploration action. When $\alpha = 0$, no weight is imposed on the pose uncertainty and the algorithm tries to cover the area as fast as possible. On the other hand, when $\alpha = 1$, full weight is given to the pose uncertainty and the robot will revisit whenever it exceeds the allowable uncertainty.

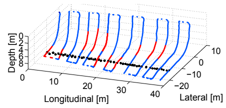
IV. Results

In this section, we evaluate the performance of PDN in two ways on a simulated environment. In the first set of experiments, we benchmark PDN against results from *(i)* preplanned exhaustive revisit and *(ii)* open-loop. In open-loop, no revisit actions are executed and the area is covered following only the preplanned nominal trajectory, while in exhaustive revisit the robot deterministically detours from the nominal trajectory in order to revisit a waypoint on the first trackline in every other leg of its survey. In the second set of experiments, we evaluate the effect of the weight factor, α, as a control parameter between navigation performance and area coverage.

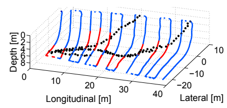
[1]The camera measurement is a 5-degree of freedom (DOF) bearing-only measurement with azimuth, elevation, and relative orientation changes [23]. For the camera measurement covariance, we assume $\pm 1^\circ$ noise for azimuth and elevation, and $\pm 0.1^\circ$ noise for the orientation change.



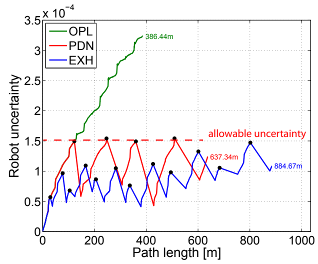
(a) Open-loop



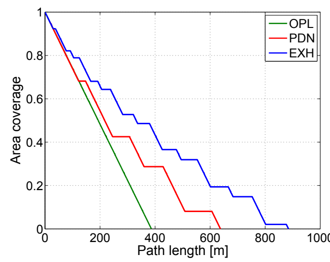
(b) Exhaustive



(c) PDN



(d) Pose uncertainty



(e) Area coverage

Fig. 5: Comparison of pose uncertainty and area coverage performance for open-loop, exhaustive revisit, and PDN. (a), (b) and (c) depict the final trajectory of the robot. Nodes on the nominal trajectory are color coded by their saliency level, blue for non-salient nodes ($S_L = 0.15$), red for salient nodes ($S_L = 0.98$), while nodes on the revisit action are emphasized with black dots. (d) depicts the 6th root of the determinant of the robot pose covariance versus the path length for open-loop (green), exhaustive revisit (blue), and PDN (red), where the black dots indicate points when revisitation occurred. (e) shows the ratio of the remaining area to cover versus path length. The uncertainty is not bounded for open-loop, but results in the fastest area coverage.

A. Simulation Setup

The experimental evaluation consists of a simulated underwater hull inspection survey that has four spatially concentrated regions of high saliency. The map consists of two saliency levels, $S_L = 0.15$ for non-salient nodes and $S_L = 0.98$ for salient nodes. A nominal survey trajectory is specified for the robot to follow, over which it must perform SLAM to build a map and localize. While performing SLAM, PDN evaluates the expected rewards from revisiting waypoints versus exploration, and determines the next-best-action to perform.

We set the user defined target coverage area by considering a bounding box around the survey region, which for this experiment is set to $\mathcal{A}_{target} = L \times (W + H) = 40\text{ m} \times (20\text{ m} + 10\text{ m}) = 1200\text{ m}^2$. Next, the user defined allowable navigation uncertainty, $|\Sigma_{\text{allow}}|$, is set as

$$|\Sigma_{\text{allow}}| = \sigma_{xy,allow}^2 \cdot \sigma_{xy,allow}^2 \cdot \sigma_d^2 \cdot \sigma_r^2 \cdot \sigma_p^2 \cdot \sigma_h^2.$$

The 1-σ allowable x and y positional uncertainties are each set to $\sigma_{xy,allow} = \pm 0.25\text{ m}$, the depth uncertainty is set to $\sigma_d = \pm 0.01\text{ m}$, and attitude uncertainties are set to $\sigma_r = \sigma_p = \sigma_h = \pm 0.1°$ (roll, pitch and heading, accordingly).

B. Pose-uncertainty-only PDN

Fig. 5 shows the comparative results of PDN, open-loop, and exhaustive revisit. For this set of tests, we set $\alpha = 1$ in PDN to make it trigger absolutely on navigation uncertainty. Since the actual distribution of salient regions cannot be known a priori, the exhaustive revisit path is arbitrarily preplanned to follow a path along the bottom of the hull. Fig. 5(a), (b) and (c) depict the final trajectories of the robot and any revisit detours made along the way.

A comparison of the robot's pose uncertainty versus path-length is plotted in Fig. 5(d), and a measure of its uncovered area versus path length is shown in Fig. 5(e). In Fig. 5(d), the robot's pose uncertainty is computed as $\sqrt[6]{|\Sigma_{rr}|}$ and is shown for open-loop (green), exhaustive revisit (blue) and PDN (red); the black dots indicate instances when revisitation occurred. In Fig. 5(e), the ratio of the remaining area to cover with respect to the target area is shown.

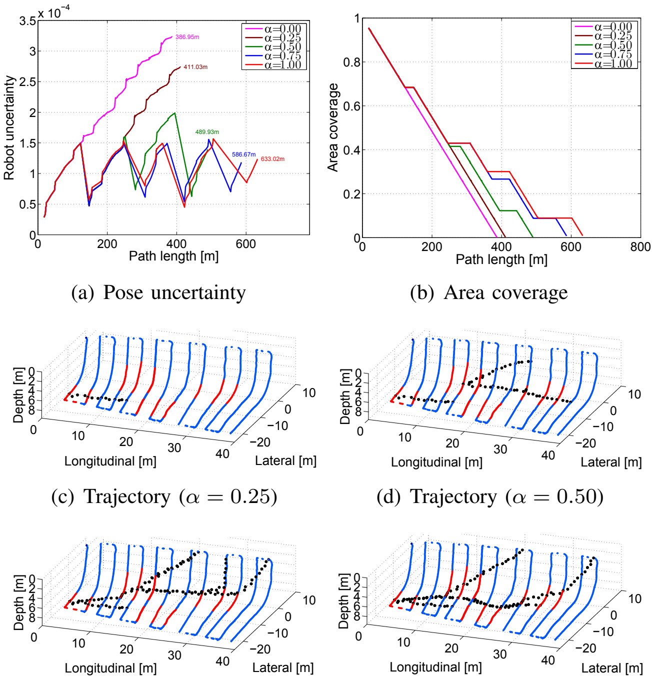
Looking at the graphs, some observations can be made:

- The uncertainty is not bounded for open-loop, but results in the fastest area coverage because it does not execute any revisit actions.
- Exhaustive revisit has the longest path length and slowest area coverage due to its regular revisits. Though it is capable of bounding the uncertainty very tightly when there are sufficient loop-closures (as shown above), it is not guaranteed to do so—for example, it could perform very poorly if it were to always drive through non-salient regions since loop-closures would fail to occur. Preplanned regular revisits lead the robot to revisit regardless of its localization performance—it will execute unnecessary revisits even when the current robot pose uncertainty is small or, conversely, not command any revisits even when the localization performance is getting uncertain.
- Given allowable uncertainty with $\sigma_{xy,allow} = \pm 0.25\text{ m}$, PDN performs in between open-loop and exhaustive revisit in terms of both path length and area coverage performance. PDN enables full control over the uncertainty level of the robot and keeps it under the user-defined allowable uncertainty level. In total, for this simulation, PDN revisits only 4 times as compared to the 12 times of exhaustive revisit, yet it is still able to achieve comparable localization performance.

C. Effect of α in PDN

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 open-loop. When $\alpha = 1$, full weight is given to the pose uncertainty, and PDN tries to reduce the uncertainty once it reaches the allowable uncertainty threshold. In other words, the effect of α is to delay the execution of revisiting by PDN.

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). The effect of α can be seen in Fig. 6, which presents several SLAM trajectories with different α weight factors. As the weight on pose uncertainty increases (from 0.25 to 1.00 in 0.25 increments), PDN tends to



(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. 6: A comparison of PDN's performance for different values of the weight, α. When $\alpha = 0$, PDN performs open-loop control. When $\alpha = 1$, PDN reacts instantly once the pose uncertainty exceeds the allowable uncertainty level. (a) shows the change of pose uncertainty with respect to α. (b) shows the area coverage rate in terms of α. $\alpha = 0$ shows the fastest coverage rate, which is slowed as α increases and weights the pose uncertainty more. (c)–(f) PDN-aided SLAM trajectories for different values of α.

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 the area coverage performance.

V. Conclusion

This paper presented an integrated framework for active SLAM with exploration, called perception-driven navigation. By integrating PDN's decision-making module within the SLAM framework, PDN is able to solve for the next-best-action considering both navigation and area coverage performance. A weight factor, α, is used to control the relative emphasis between navigation performance and area coverage rate. PDN's integrated SLAM control scheme was evaluated for a simulated underwater hull inspection mission, and compared favorably against two other preplanned mission profiles.

References

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

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

[3] R. Sim, "Stable exploration for bearings-only SLAM," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, Apr. 2005, pp. 2411–2416.

[4] R. Sim and N. Roy, "Global a-optimal robot exploration in SLAM," in *Proc. of the IEEE Int. Conf. on Robotics 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. on Pattern Analysis and Machine Intell.*, vol. 29, pp. 1052–1067, 2007.

[6] S. Prentice and N. Roy, "The belief roadmap: Efficient planning in linear POMDPs by factoring the covariance," *Int. J. of Robotics Research*, vol. 8, no. 11-12, pp. 1448–1465, Dec. 2009.

[7] R. Valencia, J. Andrade-Cetto, and J. Porta, "Path planning in belief space with pose SLAM," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, May. 2011, pp. 78–83.

[8] C. Connolly, "The determination of next best views," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, vol. 2, Mar. 1985, pp. 432–435.

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

[10] P. Whaite and F. Ferrie, "Autonomous exploration: Driven by uncertainty," *IEEE Trans. on Pattern Analysis and Machine Intell.*, vol. 19, no. 3, pp. 193–205, mar 1997.

[11] A. A. Makarenko, S. B. Williams, F. Bourgault, and H. F. Durrant-Whyte, "An experiment in integrated exploration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, 2002, pp. 534–539.

[12] 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.*, 2002, pp. 540–545.

[13] C. Stachniss, G. Grisetti, and W. Burgard, "Information gain-based exploration using rao-blackwellized particle filters," in *Proc. of the Robotics: Science & Systems Conf.*, Cambridge, MA, USA, 2005.

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

[15] M. Kaess, H. Johannsson, and J. Leonard, "Open source implementation of iSAM," http://people.csail.mit.edu/kaess/isam, 2010.

[16] 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. of Robotics Research*, 2012, In Press.

[17] A. Kim and R. M. Eustice, "Combined visually and geometrically informative link hypothesis for pose-graph visual SLAM using bag-of-words," in *Proc. of the IEEE/RSJ Int. Conf. on Intell. Robots and Syst.*, San Francisco, CA, Sep. 2011, pp. 1647–1654.

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

[19] M. Daszykowski, B. Walczak, and D. L. Massart, "Looking for natural patterns in data part 1: Density-based approach," *Chemometrics and Intelligent Laboratory Systems*, vol. 56, no. 2, pp. 83–92, 2001.

[20] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Pearson Education, 2003.

[21] L. E. Kavraki and S. M. LaValle, "Motion planning," in *Springer Handbook of Robotics*, 2008, pp. 109–131.

[22] 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.

[23] R. M. Eustice, O. Pizarro, and H. Singh, "Visually augmented navigation for autonomous underwater vehicles," *IEEE J. of Oceanic Engineering*, vol. 33, no. 2, pp. 103–122, Apr. 2008.

[24] 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. of Robotics Research*, vol. 25, no. 12, pp. 1223–1242, 2006.