Variable-Resolution Virtual Maps for Autonomous Exploration with Unmanned Surface Vehicles (USVs)

Ye Li[1], Yewei Huang[2], Wenlong GaoZhang[1], Alberto Quattrini Li[2], Brendan Englot[3], and Yuanchang Liu[1]

***Abstract*— Autonomous exploration by unmanned surface vehicles (USVs) in near-shore waters requires reliable localisation and consistent mapping over extended areas, but this is challenged by GNSS degradation, environment-induced localisation uncertainty, and limited on-board computation. Virtual map–based methods explicitly model localisation and mapping uncertainty by tightly coupling factor-graph SLAM with a map uncertainty criterion. However, their storage and computational costs scale poorly with fixed-resolution workspace discretisations, leading to inefficiency in large near-shore environments. Moreover, overvaluing feature-sparse open-water regions can increase the risk of SLAM failure as a result of imbalance between exploration and exploitation. To address these limitations, we propose a Variable-Resolution Virtual Map (VRVM), a computationally efficient method for representing map uncertainty using bivariate Gaussian virtual landmarks placed in the cells of an adaptive quadtree. The adaptive quadtree enables an area-weighted uncertainty representation that keeps coarse, far-field virtual landmarks deliberately uncertain while allocating higher resolution to information-dense regions, and reduces the sensitivity of the map valuation to local refinements of the tree. An expectation–maximisation (EM) planner is adopted to evaluate pose and map uncertainty along frontiers using the VRVM, balancing exploration and exploitation. We evaluate VRVM in the VRX Gazebo simulator, using a realistic marina environment.**

I. Introduction

Autonomous exploration of near-shore waters is vital for applications including harbour inspection, canal surveying, and berth monitoring, particularly because these environments pose major localisation challenges for unmanned surface vehicles (USVs). Human-made structures, such as quay walls, bridges, and port superstructures, obstruct or reflect global navigation satellite system (GNSS) signals, resulting in significant biases and outages [1], [2]. Moreover, harbour operations impose strict time constraints on USV missions, as vessel traffic can change rapidly and the presence of large moving vessels introduces significant safety risks, making it nontrivial for missions to be completed safely, accurately and efficiently.

In such GNSS-degraded near-shore environments, USVs must rely on simultaneous localisation and mapping (SLAM) for pose estimation and mapping. However, extended waterways and open basins are often feature-sparse [3], [4], and the available geometric structure is highly uneven—dense and stable near shorelines and quays, but largely absent in open-water regions. This imbalance increases the risk of SLAM drift during open-water traversal, while exhaustive surveying of low-information regions wastes valuable mission time under strict operational constraints. As a result, exploration strategies in these environments must jointly consider exploration efficiency and also the performance of the USV's localisation and mapping processes.

Existing exploration strategies only partially address these considerations. Early frontier-based [5], [6], [7] and next-best-view (NBV) [8], [9] methods rely on simple heuristic utility functions to enable fast exploration, but do not explicitly model localisation or mapping uncertainty. As a result, when executed in near-shore environments, these methods will often overvalue long excursions into open water, accumulating localisation drift and producing blurred or inconsistent map overlap, which leads to repeated and inefficient traversals.

Most information-theoretic methods [10], [11], [12], [13], [14] consider entropy associated with both localisation and mapping, but operating directly on probabilistic belief representations is computationally expensive. As a result, these approaches often rely on simplifying independence assumptions, such as weak coupling between the robot state and the map or independence among map grid cells. Consequently, their performance degrades as long-range correlations accumulate in large environments, and may lead to localisation drift in environments with uneven geometric structures.

Instead of evaluating map entropy based on an occupancy grid map, the virtual map quantifies uncertainty over a visited area using marginal covariances associated with robot state estimates and the corresponding sensing model [15]. The covariances of virtual landmarks are updated to predict how future robot observations will affect map uncertainty, using an expectation–maximisation (EM) framework. Experimental results demonstrate that the virtual map achieves a balance between accurate map-building and efficient exploration on both ground robots [16] and autonomous underwater vehicles [17]. However, a uniformly discretised virtual map is less effective for USVs operating in unevenly structured coastal environments. In such settings, a grid-structured virtual map can lead to overly conservative behaviour when the vehicle repeatedly explores structure-free regions. Computational resources are expended maintaining low-entropy virtual landmarks in feature-poor areas, while high-uncertainty regions are not prioritised in a manner consistent with the vehicle's actual sensing capability.

[1]Department of Mechanical Engineering, University College London, London, UK, {ye.li, wenlong.gaozhang, yuanchang.liu}@ucl.ac.uk

[2]Department of Computer Science, Dartmouth College, Hanover, NH, USA, {yewei.huang, alberto.quattrini.li}@dartmouth.edu

[3]Department of Mechanical Engineering, Stevens Institute of Technology, Hoboken, NJ, USA, benglot@stevens.edu

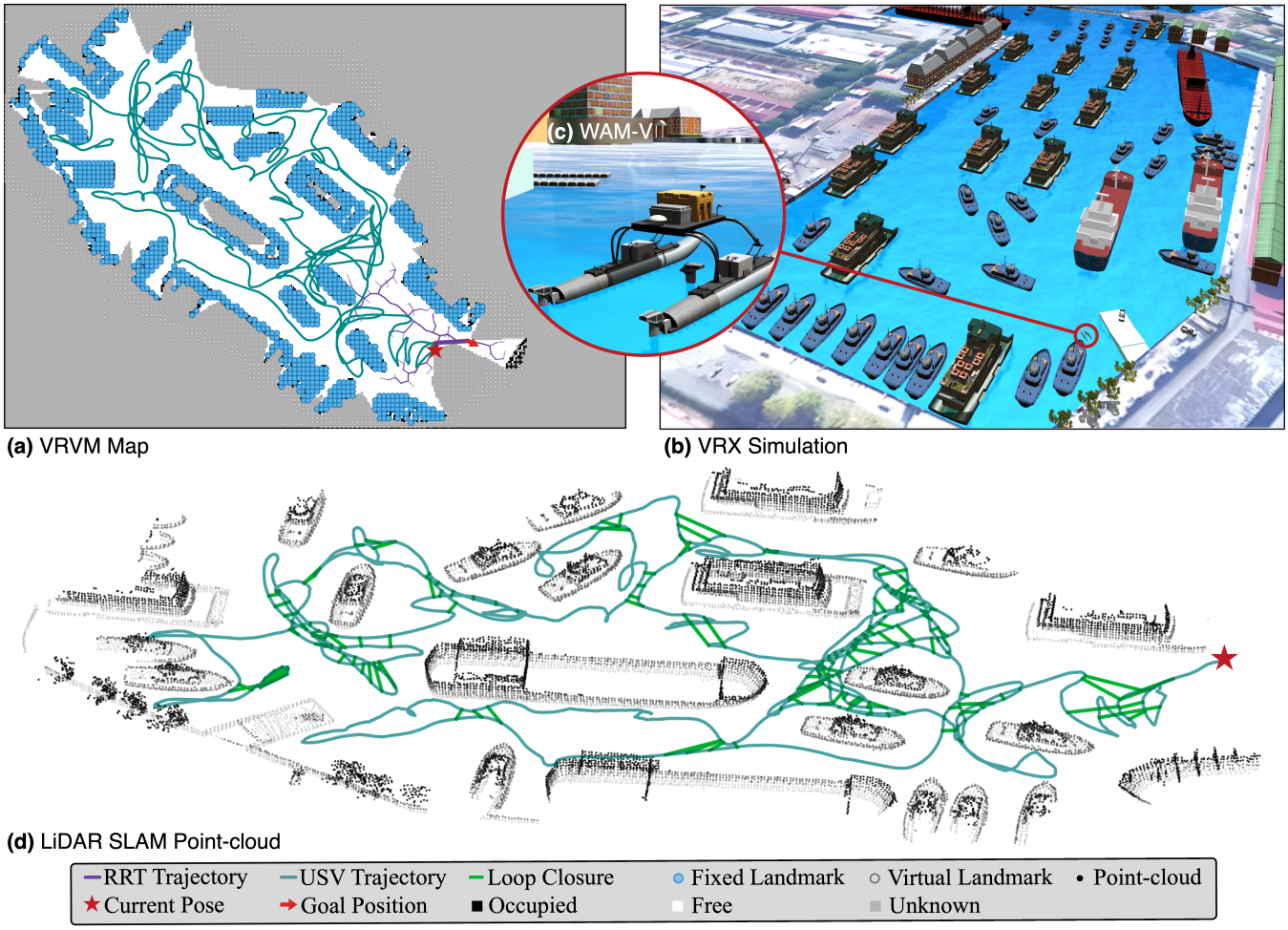


Fig. 1: Illustration of the WAM-V (c) in our $210\,\mathrm{m} \times 500\,\mathrm{m}$ Harbour Basin scene (b). The VRVM map (a) and corresponding LiDAR SLAM point cloud (d) are shown. The USV is depicted in red, with the trajectory history in green. White cells indicate observed space, and gray cells denote unobserved space. Ellipses represent the covariance associated with each virtual landmark.

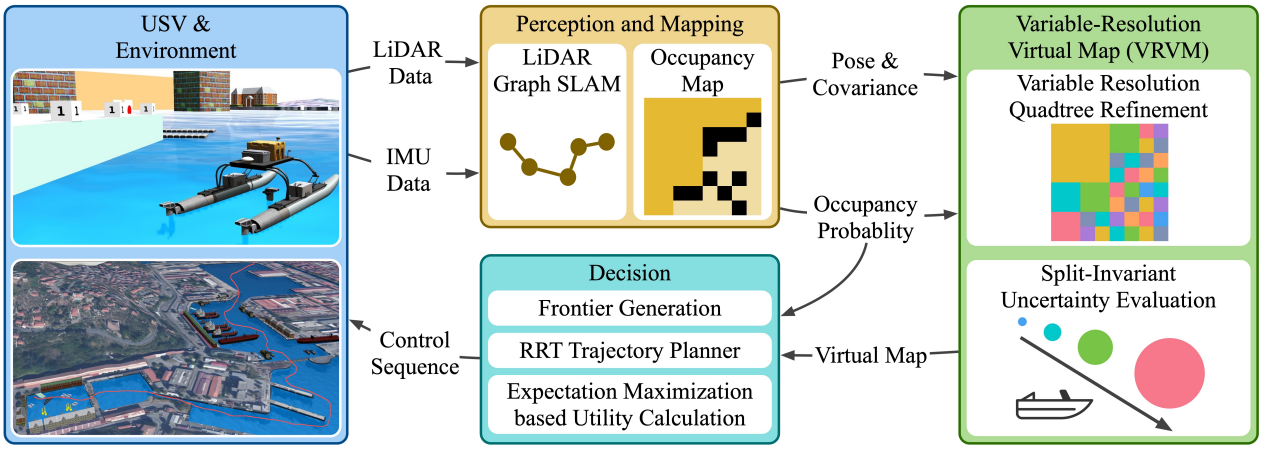


Fig. 2: Overview of the navigation pipeline. The factor graph produced by LiDAR-based Graph SLAM is used to construct the variable-resolution virtual map (VRVM), which provides uncertainty-aware information for decision making.

Building upon virtual maps, we propose a variable-resolution virtual map (VRVM). While VRVM is applicable to various robotic platforms, it is specifically designed for USVs exploring large near-shore scenes, where uneven structures require a careful balance between exploration and localisation stability. VRVM maintains Gaussian virtual landmarks on a quadtree and refines only uncertainty- or occupancy-ambiguous regions inside the sensor range. This makes the per-cycle cost scale with the size of the observable region rather than with the map discretisation. We introduce an area-weighted map valuation that reduces dependence on the current split pattern and avoids spuriously rewarding trajectories that pass through feature-sparse water. On top of this representation, we use an EM planner that selects frontiers by combining trajectory uncertainty, predicted reductions in virtual-map uncertainty, and path cost.

As summarized in Fig. 1, we evaluate the VRVM algorithm by deploying it in the VRX Gazebo simulator [18], using a LIO-SAM backbone [19], in a representative near-shore environment. VRVM achieves a compelling accuracy–coverage trade-off, with modest computational expense. VRVM's area-weighted map valuation strategy achieves a compelling balance between exploration and exploitation in structurally unbalanced nearshore coastal environments.

II. Problem Description

We consider an autonomous exploration problem that is tightly coupled with simultaneous localisation and mapping (SLAM) using factor-graph optimisation. A USV explores within a fixed 2D workspace $\mathcal{W} \subset \mathbb{R}^2$ defined in the world frame $\{W\}$, and carries LiDAR and IMU sensors. The overall goal for the USV is to traverse the workspace, achieve full boundary coverage of all structures in $\mathcal{W}$, and estimate their geometry with high mapping accuracy. During the exploration, the robot decides its own incremental goal and plans the path accordingly. Once the current goal is reached, a new set of goal candidates (*frontiers*) is selected from both the boundary of the explored region and the structure-rich areas within the known map. This selection strategy accounts for the localisation uncertainty introduced by sensor noise and environmental variations, such as wind and hydrodynamic effects, and enables the USV to balance *exploration* (gaining additional coverage) with *exploitation* (revisiting known regions to support loop closures). An overview of the exploration pipeline is presented in Fig. 2.

A. Robot and Sensor Model

The robot considered in this paper is a USV equipped with an IMU and a LiDAR sensor. We denote the USV state (position and heading) at time step t by $\mathbf{x}_t = [\,x_t,\ y_t,\ \psi_t\,]^\top \in \mathbb{R}^3$. x_i and y_i denote the USV position in the world frame $\{W\}$ and ψ_i is the orientation. The USV motion model is defined as:

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t) + \mathbf{w}_t, \tag{1}$$

where $\mathbf{u}_t$ denotes the control input and $\mathbf{w}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_t)$ is the zero-mean process noise with covariance $\mathbf{Q}_t$. $f(\cdot)$ denotes the discrete-time motion model.

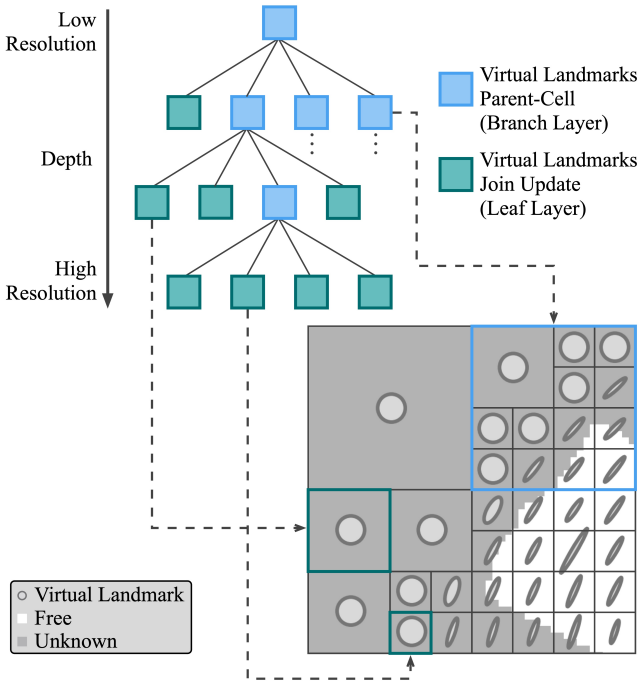


Fig. 3: Visualization of a portion of the variable-resolution virtual map hierarchy over a quadtree. A correspondence between the leaf and branch cells in the quadtree and their physical surface regions is shown.

Two types of sensor measurements are considered: the IMU odometry measurement $\mathbf{z}_t^{\text{imu}}$ and the LiDAR measurement $\mathbf{z}_t^{\text{lidar}}$. The IMU odometry measurement model is defined as:

$$\mathbf{z}_t^{\text{imu}} = h_{\text{imu}}(\mathbf{x}_t, \mathbf{x}_{t+1}) + \mathbf{n}_t^{\text{imu}}, \tag{2}$$

where $\mathbf{n}_t^{\text{imu}} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_t^{\text{imu}})$ is zero-mean IMU measurement noise with covariance $\mathbf{R}_t^{\text{imu}}$. The function $h_{\text{imu}}(\cdot)$ returns the relative motion between two consecutive states obtained through IMU preintegration [20]. The LiDAR sensor measurement model is defined as

$$\mathbf{z}_t^{\text{lidar}} = h_{\text{lidar}}(\mathbf{x}_t, \mathcal{M}) + \mathbf{n}_t^{\text{lidar}}, \tag{3}$$

where $\mathbf{n}_t^{\text{lidar}} \sim \mathcal{N}(\mathbf{0},\ \mathbf{R}_t^{\text{lidar}})$ is zero-mean LiDAR measurement noise with covariance $\mathbf{R}_t^{\text{lidar}}$. The function $h_{\text{lidar}}(\cdot)$ returns the LiDAR ranges at pose $\mathbf{x}_t$ given the occupancy map $\mathcal{M}$. Thus, the inverse sensor model for LiDAR $g_{\text{lidar}}(\cdot)$ is given by:

$$p(\mathcal{M} \mid \mathbf{z}_t^{\text{lidar}}, \mathbf{x}_t) \propto g_{\text{lidar}}\big(\mathbf{z}_t^{\text{lidar}}, \mathbf{x}_t\big)\,. \tag{4}$$

III. Variable-Resolution Virtual Map

The variable-resolution virtual map (VRVM) runs in a receding-horizon [21] loop that couples a variable-resolution virtual map with visibility-limited prediction and an EM planner. During each control cycle, the quadtree map is adaptively refined based on depth-aware, uncertainty-driven rules, where only visible leaves are updated using the inverse sensor model. The uncertainty of reliably occupied leaves containing obstacles is held fixed to prevent redundant updates. This quadtree map serves as a reference for ranking

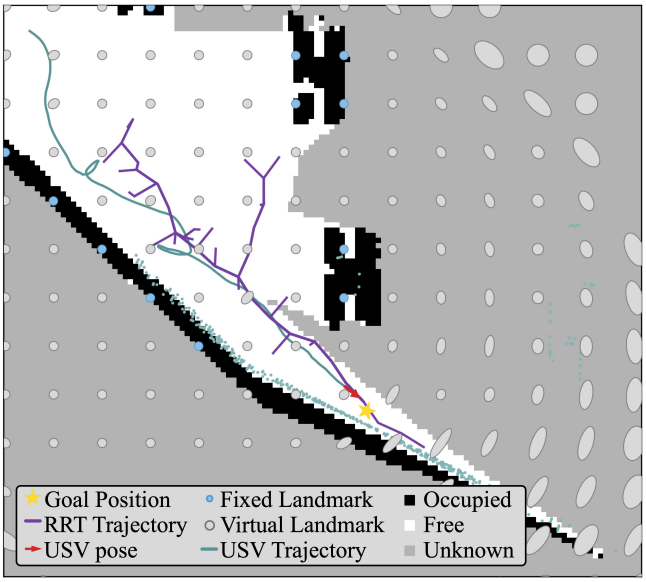


Fig. 4: Example of the EM–planner selecting a goal and generating an RRT trajectory over the VRVM. The USV trajectory (teal), RRT tree (purple), and goal position (yellow star) are shown together with fixed landmarks (blue), virtual landmarks (gray), and the occupancy map, where black, white, and gray denote occupied, free, and unknown regions.

Algorithm 1: Adaptive Quadtree Refinement (VRVM)

Root $\mathbf{v}_{q_{\text{root}}}$; size floor r_0; depth cap $d_{\max}$; thresholds $\tau^0_{\text{det}}; \tau_p$; init variance Σ_0. Updated leaf set $\mathcal{V}$.
Function `Refine(`$\mathbf{v}_q$`)`**:**
  if $\mathbf{v}_q$ *is not a leaf* **then**
    foreach *child* $\mathbf{v}_c$ *of* $\mathbf{v}_q$ **do**
      `Refine(`$\mathbf{v}_c$`)`
  else if $\big(\det(\Sigma_q) \geq \tau^0_{\text{det}} 4^{-d_q} \;\vee\; |P_{\text{v}}(\mathbf{v}_q) - 0.5| \leq \tau_p\big) \;\wedge\; h_q > r_0/2 \;\wedge\; d_q < d_{\max}$ **then**
    `Split(`$\mathbf{v}_q$`) ; // into four children`
    for $i \leftarrow 1$ **to** 4 **do**
      $\mathbf{v}_c \leftarrow \text{child}_i(\mathbf{v}_q)$; $P_{\text{v}}(\mathbf{v}_c) \leftarrow P_{\text{v}}(\mathbf{v}_q)$
      $\mu_c \leftarrow$ subcell centre; $\Sigma_c \leftarrow \Sigma_0$
      `Refine(`$\mathbf{v}_c$`)`

`Refine(`$\mathbf{v}_{q_{\text{root}}}$`)`

kinodynamically feasible trajectories using a unified utility, $U(\pi)$, that combines the end-of-horizon SLAM optimised pose log-determinant ($U_{\text{traj}}(\pi)$), the area-weighted map log-determinant over the visible set ($U_{\text{map}}(\pi)$), and a path-length penalty ($U_{\text{length}}(\pi)$). The resulting formulation maintains computational complexity approximately linear in the visible-set size and remains compatible with incremental smoothing factor graph SLAM.

A. Variable-Resolution Quadtree Refinement and Occupancy Locking

Our variable-resolution virtual map is implemented using a quadtree rather than a uniform grid. In this quadtree $\mathcal{V}$

Algorithm 2: Projective Occupancy Locking

Projected occupancy grid $\mathcal{M}$; threshold θ_{occ}; fixed variance Σ_{fix}. Locked leaves removed from future updates. **foreach** *finest-resolution leaf* $\mathbf{v}_q$, $\mathbf{v}_q \in \mathcal{V}$ **do**
  $(i,j) \leftarrow$ `WorldToGrid`$(\mu_q, \mathcal{M})$ **if** $\mathcal{M}[i,j] \geq \theta_{\text{occ}}$ **then**
    $\Sigma_q \leftarrow \Sigma_{\text{fix}}$; `// mark q as fixed`

(Fig. 3), each leaf $\mathbf{v}_q$ represents a virtual landmark and carries an area indicator specifying the spatial coverage of the leaf. During exploration, the quadtree is refined adaptively to ensure that the virtual map provides the resolution needed for planning.

At the start of exploration, the virtual map consists of a quadtree with a single leaf. As the robot begins to observe and explore the environment, the tree is recursively refined. As shown in Fig. 1, refinement is applied only where it benefits planning: regions with high posterior uncertainty are maintained at high resolution, whereas structure-free regions remain coarse.

A leaf $\mathbf{v}_q$ is split only when it is observed and its local uncertainty $\mathbf{\Sigma}_q$ remains high at its current resolution, or when its occupancy state is ambiguous. Formally,

$$\det(\mathbf{\Sigma}_q) \;\geq\; \tau_{\text{det}}(d_q) \quad \text{or} \quad \big|\, P_{\text{v}}(\mathbf{v}_q) - 0.5 \,\big| \;\leq\; \tau_p, \tag{5}$$

where d_q denotes the depth of leaf q. We employ a depth-aware uncertainty threshold of the form $\tau_{\text{det}}(d_q) = \tau^0_{\text{det}}\, 4^{-d_q}$, where τ^0_{det} is the base threshold. The term $P_{\text{v}}(\mathbf{v}_q) \in [0,1]$ denotes the occupancy probability at the leaf centre.

A refinement (leaf-splitting) termination criterion is also enforced. A leaf $\mathbf{v}_q$ may be split only if both

$$h_q > \frac{r_0}{2} \qquad \text{and} \qquad d_q < d_{\max}, \tag{6}$$

that is, splitting is permitted only when the half leaf size h_q is larger than the minimum admissible resolution $\frac{r_0}{2}$ and the quadtree depth limit $d_{\max}$ has not been reached. Equivalently, refinement is terminated once either

$$h_q \leq \frac{r_0}{2} \qquad \text{or} \qquad d_q \geq d_{\max}. \tag{7}$$

The parameters τ^0_{det}, τ_p, and $d_{\max}$ are predefined constants. When a split occurs, four children are created at the centres of the subcells and are initialised according to

$$\mathbf{\Sigma}_{\text{child}} = \mathbf{\Sigma}_0, \qquad P_{\text{v}}(\mathbf{v}_{\text{child}}) = P_{\text{v}}(\mathbf{v}_q). \tag{8}$$

The full refinement procedure is summarised in Alg. 1.

Leaves whose centres fall in high occupancy-probability cells are locked by setting

$$\mathbf{\Sigma}_q = \mathbf{\Sigma}_{\text{fix}} = \begin{bmatrix} \sigma^2_{\text{fix}} & 0 \\ 0 & \sigma^2_{\text{fix},} \end{bmatrix}, \tag{9}$$

and excluding them from subsequent updates and from the map uncertainty term. Locking is applied after visible-set

updates in each cycle, as shown in Alg. 2. Let N_{split} denote the number of splits performed in a cycle. The computational complexity of one refinement cycle is

$$O_{\text{refine}} = O(N_{\text{split}}) + O(|\mathcal{V}(\mathbf{x}_k)|)\,, \tag{10}$$

and is typically dominated by the size of the visible set.

B. Area-Weighted Map Valuation

The mapping accuracy is quantified by the mapping utility $U_{\text{map}}(\pi)$, where the uncertainty evaluation function $\phi(\mathbf{v}_q)$ is defined as a log-determinant metric. For the uniform virtual map, the virtual landmarks are evenly distributed, and each one contributes with equal weight. In contrast, for the variable-resolution representation, directly summing $\log\det(\mathbf{\Sigma}_q)$ over leaves is split-sensitive: dividing a coarse leaf into several finer leaves with similar covariances increases the total simply because the number of terms increases, which can bias decisions at earlier stages. We therefore seek a valuation that is less dependent on the current split structure while still promoting viewpoints that reduce uncertainty in high-entropy regions.

For a set of virtual landmarks $\mathcal{V}$, we define the split-invariant weight of a leaf $\mathbf{v}_q \in \mathcal{V}$ as

$$w_{\text{area}}(\mathbf{v}_q, \mathcal{V}) \;=\; \frac{A_q}{\sum_{\mathbf{v}_r \in \mathcal{V}} A_r}, \tag{11}$$

where $A_q = 4h_q^2$ denotes the area associated with leaf $\mathbf{v}_q$. By construction, these weights are normalised such that

$$\sum_{\mathbf{v}_r \in \mathcal{V}} w_{\text{area}}(\mathbf{v}_r, \mathcal{V}) = 1. \tag{12}$$

The area-weighted map uncertainty of the virtual map is then given by

$$J_{\text{area}}(\mathcal{V}) \;=\; \sum_{\mathbf{v}_q \in \mathcal{V}} w_{\text{area}}(\mathbf{v}_q, \mathcal{V})\, \phi(\mathbf{v}_q). \tag{13}$$

If a parent leaf is split into children whose covariances are locally similar to the parent covariance, the weighted sum in Eq.(13) remains approximately unchanged, so $J_{\text{area}}(\mathcal{X}_{0:t})$ is locally invariant to the current quadtree split pattern.

For a candidate control sequence π with an end-of-horizon trajectory $\mathcal{X}_{0:t+h}$, we propagate our variable-resolution virtual map along the discretised trajectory. The mapping utility $U_{\text{map}}(\pi)$ is then formulated using area-weighted map valuation criteria.

The mapping utility $U_{\text{map}}(\pi)$ consists of two components: the end-of-horizon virtual map uncertainty $J_{\text{area}}(\mathcal{V}_{t+h})$, and the mapping accuracy gain achieved by executing the control policy $\Delta J_{\text{gain}}(\pi)$:

$$U_{\text{map}}(\pi) = J_{\text{area}}(\mathcal{V}_{t+h}) + \Delta J_{\text{gain}}(\pi). \tag{14}$$

To emphasise uncertainty reduction along the executed trajectory, we define the split-invariant gain $\Delta J_{\text{gain}}(\pi)$ as:

$$\Delta J_{\text{gain}}(\pi) = \sum_{k=0}^{t+h} \left[\gamma_k\, J_{\text{area}}\big(\mathcal{V}_{t+h}(\mathbf{x}_k)\big) \;-\; J_{\text{area}}\big(\mathcal{V}_t(\mathbf{x}_k)\big)\right]. \tag{15}$$

Algorithm 3: Split-Invariant Mapping Accuracy Gain

Input: Initial trajectory guess $\bar{\mathcal{X}}_{0:t+h}$; optimised trajectory $\mathcal{X}^{\star}_{0:t+h}$; virtual maps before and after executing policy π, $\mathcal{V}_t$ and $\mathcal{V}_{t+h}$; weights $\{\gamma_k\}_{k=0}^{t+h}$.

Output: Split-invariant gain $\Delta J_{\text{gain}}(\pi)$.

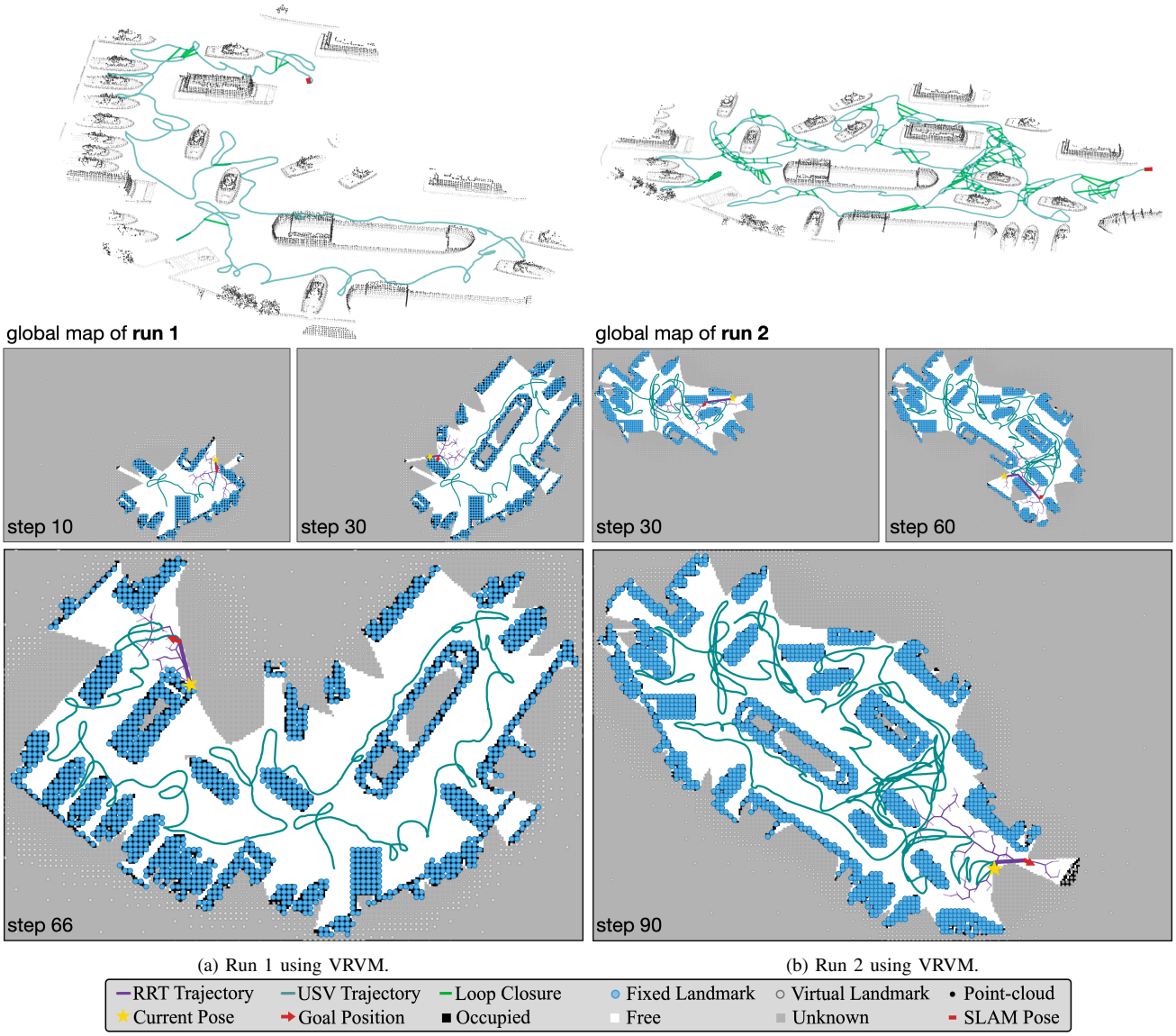
$\Delta J_{\text{gain}} \leftarrow 0$;
for $k \leftarrow 0$ **to** $t+h$ **do**
  $\mathcal{V}_t(\mathbf{x}_k) \leftarrow \text{Visible}(\mathcal{V}_t, \mathbf{x}_k)$,;
  $\mathcal{V}_{t+h}(\mathbf{x}_k) \leftarrow \text{Visible}(\mathcal{V}_{t+h}, \mathbf{x}_k)$;
  ; // Extract visible sets
  $S_t \leftarrow \sum_{\mathbf{v}_q \in \mathcal{V}_t(\mathbf{x}_k)} A_q$,;
  $S_{t+h} \leftarrow \sum_{\mathbf{v}_q \in \mathcal{V}_{t+h}(\mathbf{x}_k)} A_q$;
  ; // Area-normalisation terms
  $J_t(\mathbf{x}_k) \leftarrow -\sum_{\mathbf{v}_q \in \mathcal{V}_t(\mathbf{x}_k)} \frac{A_q}{S_t} \operatorname{logdet}(\Sigma_q)$,;
  $J_{t+h}(\mathbf{x}_k) \leftarrow -\sum_{\mathbf{v}_q \in \mathcal{V}_{t+h}(\mathbf{x}_k)} \frac{A_q}{S_{t+h}} \operatorname{logdet}(\Sigma_q)$;
  ; // Are-weighted uncertainties
  $\Delta J_{\text{gain}} \leftarrow \Delta J_{\text{gain}} + \gamma_k\, J_{t+h}(\mathbf{x}_k) - J_t(\mathbf{x}_k)$;
return ΔJ_{gain};

where $\mathcal{V}_t(\mathbf{x}_k)$ and $\mathcal{V}_{t+h}(\mathbf{x}_k)$ denote the sets of virtual landmarks visible from $\mathbf{x}_k$ in the virtual maps constructed before and after executing the control sequence π, respectively. $\gamma_k \in (0, 1]$ is a discount factor to downweigh samples that require longer travel distance or have higher predicted pose uncertainty. We summarise our calculation of $\Delta J_{\text{gain}}(\pi)$ in Alg. 3. In our implementation, we simplify the formulation by using a fixed γ_k. The proposed area-weighted map valuation is used only to rank candidate control sequences during utility calculation.

C. Planning and Utility Calculation

In this section, we summarise the overall planning pipeline of the VRVM. At each control cycle a finite candidate set $\Pi = \{\pi\}$ is assembled by sampling exploring frontiers located at the boundary between known and unknown regions of the occupancy grid, together with exploit frontiers near previously observed structures to encourage loop closures and improve localisation. For each frontier, a kinodynamically feasible path is generated using a goal-biased RRT alogrithm [22] followed by trajectory short-cutting. Candidates that violate collision constraints or curvature limits are discarded. Fig. 4 illustrates the set of end-of-horizon trajectories generated by the RRT planner, shown in purple.

The pose covariance of each end-of-horizon trajectory is propagated using a factor graph optimiser based on the latest occupancy map; in our implementation, we use iSAM2 [23]. The VRVM is then updated accordingly using a visible-set update along the predicted trajectory. We evaluate a scalar utility function that balances three objectives: pose uncertainty reduction, area-weighted map uncertainty reduction, and path cost. The planner selects the feasible control sequence with the highest utility value. The trajectory and



(a) Run 1 using VRVM.

(b) Run 2 using VRVM.

Fig. 5: Representative exploration runs. Run 1 snapshots are shown at steps 10, 30, and 66 with the final SLAM point-cloud map; Run 2 snapshots are shown at steps 30, 60, and 90 with the final SLAM point-cloud map.

energy utility terms are defined as:

$$U_{\text{traj}}(\pi) = -\log\det\big(\mathbf{\Sigma}_{\mathbf{p},t+H}(\pi)\big), \tag{16}$$

$$U_{\text{length}}(\pi) = -\alpha\text{length}(\pi), \tag{17}$$

where $\mathbf{\Sigma}_{\mathbf{p},t+H}(\pi)$ denotes the predicted pose covariance at the end of the trajectory, and α is a scalar weight that penalises longer paths.

We execute $\pi^\star$ in a receding-horizon fashion: after traversing a short segment, the SLAM smoother assimilates new odometry and loop-closure factors, the VRVM fuses visible-set information, the candidate frontiers are refreshed, and the VRVM-based EM planner evaluates the candidate control sequences. The computation required to evaluate the utilities of all candidates scales with the number of sampled frontiers. The prediction cost is $O\Big(\sum_{\pi\in\Pi}\sum_k \big|\mathcal{V}(\mathbf{x}_k)\big|\Big)$, with only 2×2 operations on VRVM leaves, plus the incremental smoothing updates provided by the SLAM back-end. By introducing the area-weighted map valuation and the visible-set update, we reduce the computational cost relative to EM planners whose operations scale with the map size. Example results are illustrated in Fig. 5.

IV. Conclusion

Motivated by long-horizon USV exploration in GNSS-degraded near-shore waters, this work addressed a practical limitation of uncertainty-aware planning: the inefficiency and instability that arise when uncertainty must be evaluated over large, geometrically uneven workspaces under constrained onboard computation. We introduced a variable-resolution virtual map (VRVM) framework that confines uncertainty updates to the sensor-visible set and refines representation adaptively, coupled with split-stable, area-weighted valuation within a receding-horizon planner. This design preserves compatibility with factor-graph SLAM while shifting computation from global bookkeeping to locally relevant updates.

Acknowledgements

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) under Grants EP/Y000862/1 and EP/X034909/1, and by The Royal Society Kan Tong Po Fellowship (KTP/R1/251117). This work was also supported in part by NSF grant 2144624, and by ONR grant N00014-24-1-2522.

References

[1] A. Makar, "Limitations of multi-GNSS positioning of USV in area with high harbour infrastructure," *Electronics*, vol. 12, no. 3, 2023.

[2] A. Pandele, A. Croitoru, A. Hulea, C. Cherciu, A. Radutu, I. Stefanescu, K. Urbanska, D. Andrescu, C. Dragasanu, M. Trusculescu, *et al.*, "Maritime environment gnss multipath analysis in the framework of the margot project," in *Proceedings of the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2020)*, 2020, pp. 757–778.

[3] Y. Wang, C. Liu, J. Liu, J. Wang, J. Liu, K. Zheng, and R. Zheng, "A laser-based slam algorithm of the unmanned surface vehicle for accurate localization and mapping in an inland waterway scenario," *Journal of Marine Science and Engineering*, vol. 12, no. 12, 2024.

[4] Ł. Marchel, K. Naus, and M. Specht, "Optimisation of the position of navigational aids for the purposes of SLAM technology for accuracy of vessel positioning," *Journal of Navigation*, vol. 73, no. 2, pp. 282–295, 2020.

[5] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97. 'Towards New Computational Principles for Robotics and Automation'*, 1997, pp. 146–151.

[6] C. Stachniss, D. Hahnel, and W. Burgard, "Exploration with active loop-closing for fastslam," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, vol. 2, 2004, pp. 1505–1510 vol.2.

[7] C. Leung, S. Huang, and G. Dissanayake, "Active SLAM using model predictive control and attractor based exploration," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2006, pp. 5026–5031.

[8] S. Kriegel, C. Rink, T. Bodenmüller, and M. Suppa, "Efficient next-best-scan planning for autonomous 3d surface reconstruction of unknown objects," *Journal of Real-Time Image Processing*, vol. 10, no. 4, pp. 611–631, 2015.

[9] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon "next-best-view" planner for 3d exploration," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1462–1468.

[10] F. Bourgault, A. A. Makarenko, S. B. Williams, B. Grocholsky, and H. F. Durrant-Whyte, "Information based adaptive robotic exploration," in *2002 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 1. IEEE, 2002, pp. 540–545.

[11] L. Carlone, J. Du, M. K. Ng, B. Bona, and M. Indri, "An application of kullback-leibler divergence to active slam and exploration with particle filters," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 287–293.

[12] B. Charrow, S. Liu, V. Kumar, and N. Michael, "Information-theoretic mapping using cauchy-schwarz quadratic mutual information," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 4791–4798.

[13] Z. Zhang, T. Henderson, V. Sze, and S. Karaman, "FSMI: Fast computation of shannon mutual information for information-theoretic mapping," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 6912–6918.

[14] A. Asgharivaskasi, F. Girke, and N. Atanasov, "Riemannian optimization for active mapping with robot teams," *IEEE Transactions on Robotics*, pp. 1077–1097, 2025.

[15] J. Wang and B. Englot, "Autonomous exploration with expectation-maximization," in *Robotics Research: The 18th International Symposium ISRR*. Springer, 2017, pp. 759–774.

[16] J. Wang, T. Shan, and B. Englot, "Virtual maps for autonomous exploration with pose SLAM," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 4899–4906.

[17] J. Wang, F. Chen, Y. Huang, J. McConnell, T. Shan, and B. Englot, "Virtual maps for autonomous exploration of cluttered underwater environments," *IEEE Journal of Oceanic Engineering*, vol. 47, no. 4, pp. 916–935, 2022.

[18] B. Saldarriaga-Mesa, J. Montesdeoca, D. Báez, F. Roberti, and J. M. Toibero, "Open-access simulation platform and motion control design for a surface robotic vehicle in the VRX environment," *Robotics*, vol. 14, no. 10, 2025.

[19] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "LIO-SAM: Tightly-coupled lidar inertial odometry via smoothing and mapping," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 5135–5142.

[20] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual–inertial odometry," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 1–21, 2016.

[21] D. Brugali, L. Muratore, and A. De Luca, "Mobile robots exploration strategies and requirements: A systematic mapping study," *The International Journal of Robotics Research*, vol. 44, no. 9, pp. 1461–1506, 2025.

[22] C. Urmson and R. Simmons, "Approaches for heuristically biasing RRT growth," in *2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(Cat. No. 03CH37453)*, vol. 2. IEEE, 2003, pp. 1178–1183.

[23] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "iSAM2: Incremental smoothing and mapping using the Bayes tree," *The International Journal of Robotics Research*, vol. 31, no. 2, pp. 216–235, Feb. 2012.