

# Sparse Planning Graphs for Information Driven Exploration

Erik Nelson, Vishnu Desaraju, John Yao

The Robotics Institute  
Carnegie Mellon University  
Pittsburgh, PA 15217

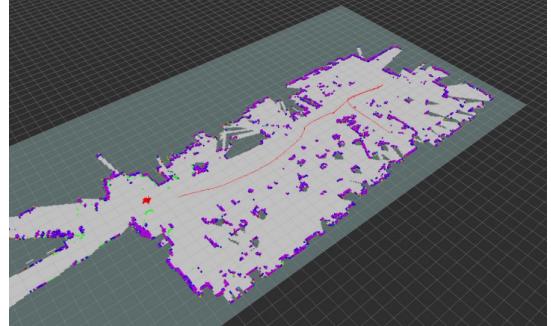
{enelson, rajeswar, johnyao}@cmu.edu

**Abstract**—This work presents a novel framework that enables online exploration of an unknown environment. We use the Cauchy-Schwarz Quadratic Mutual Information (CSQMI) metric to guide a sparse planning graph generated via the Closed-loop RRT (CL-RRT) algorithm. We discuss the formulation and implementation of the CSQMI metric, the CL-RRT planner, and an Unscented Kalman Filter (UKF) that were developed to enable state estimation and trajectory tracking. Finally, we demonstrate the performance of our approach using a high-fidelity simulation of a mobile ground robot exploring an unknown space using a laser scanner range sensor.

## I. INTRODUCTION

Exploration is a key capability that enables robotic vehicles to operate in unknown environments. The robot can be controlled to navigate the environment while building a map, leading to traditional simultaneous localization and mapping (SLAM) approaches, as illustrated in Fig. 1. However, this can be extended to also incorporate autonomous path planning, and in this project we develop an active perception policy for robotic exploration. Active perception exploration formulations choose control actions which optimize an information-theoretic objective function such as Shannon’s mutual information or entropy [1, 7] over the robot’s map, given a sensor measurement model. Other common exploration techniques, such as frontier exploration [9], use geometric reasoning to infer explorative paths. While these strategies work well in practice, they operate on a maximum likelihood estimate of the map, and apply heuristics to determine the most uncertain locations in the environment. In contrast, active perception strategies do not utilize geometric or maximum likelihood assumptions, and instead interpret the map as a binary random variable, choosing actions which directly minimize the random variable’s uncertainty. Additionally, information-based exploration methods easily extend to 3D configuration spaces, a benefit which is not shared by frontier methods. Julian et al. prove that maximizing mutual information between a robot’s map and expected future map naturally yields explorative behaviors [3].

Active perception formulations seek to optimize information-theoretic objectives. While this optimization is real-time for short planning horizons, these metrics are often expensive to compute online, requiring double integration over possible future robot states and measurements, or Monte Carlo sampling from the distribution of measurements. These expensive per-pose computations inhibit online dense evaluation over a configuration space. In this project, we



(a) Map and trajectory



(b) Ground robot

Figure 1: A ground robot mapping while being driven through a cluttered environment.

aim to develop an efficient active perception exploration strategy which evaluates the information-theoretic objective in a sparse, but well-chosen set of poses across the configuration space. This strategy evaluates the objective function a limited number of times, while still generating paths that sufficiently explore the space.

To achieve real-time active perception exploration, we use a Rapidly-Exploring Random Tree (RRT) to generate sets of dynamically feasible actions over a finite planning horizon [5]. RRT planners trade trajectory optimality for efficiency, allowing for evaluation of many potential future locations in the configuration space during a single planning step. In addition, RRT planners are anytime, and generate potential trajectories for a pre-specified amount of time before evaluating the most optimal sampled trajectory. Our strategy evaluates each RRT leaf-node using the information-theoretic objective function,

and stores the resulting reward in the tree. After planning for a specified amount of time, the maximum reward leaf-node is chosen as the optimal location to visit, and the RRT is traversed to generate a dynamically feasible trajectory to that location.

In addition to the efficiency gains from using an RRT, a recent work by Charrow et al. [2] has proposed the Cauchy-Schwarz Quadratic Mutual Information (CSQMI) as an efficient information-theoretic objective function. CSQMI is theoretically well-motivated: it is derived from Renyi's Quadratic Entropy, a generalization of Shannon's entropy. However, in contrast to Shannon's mutual information (which is derived directly from Shannon's entropy), CSQMI is shown to have superior computational efficiency.

The contribution of this work is an exploration framework to enable online exploration using CSQMI in sparse planning graphs, such as RRTs. We discuss the formulation and implementation of the CSQMI metric, RRT, and an Unscented Kalman Filter (UKF) that were developed to enable trajectory tracking and state estimation. Finally, we demonstrate the performance of our approach through a set of simulations in which a mobile ground robot must explore an unknown space using a laser scanner range sensor.

This paper is structured in the following manner: Section II gives a brief overview of occupancy grid mapping which is necessary for the CSQMI information metric. Section III details the CSQMI information-theoretic cost objective and its similarities to Shannon's mutual information. Section IV discusses the measurement model that was chosen to evaluate expected future measurements. Sections V and VI cover the RRT and UKF formulations used in our implementation. Finally, Sections VII and VIII give results and analysis of our implementation in simulation, and describe future work towards implementing our algorithms on a ground robot.

## II. OCCUPANCY GRID MAPPING

In order to develop an information-theoretic reward surface, we model the robot's map as a binary random variable. We therefore discretize the space, and represent the map as an occupancy grid - a common environmental representation for robotic mapping.

As a basis for the core formulation in the following sections, we provide a brief overview of occupancy grid mapping. Occupancy grids are a common and useful probabilistic model for representing an environment. We represent the map as an occupancy grid, which consists of a set of cells:  $m = \{m^i\}_{i=1}^N$ . The probability that an individual cell is occupied is given by  $p(m^i | x_{1:t}, z_{1:t})$ , where  $x_{1:t}$  denotes the history of states of the vehicle, and  $z_{1:t}$  denotes the history of range observations accumulated by the vehicle. We assume that cell occupancy probabilities are independent of one another:  $p(m | x_{1:t}, z_{1:t}) = \prod_i p(m^i | x_{1:t}, z_{1:t})$ . For notational simplicity we write the map conditioned on random variables  $x_{1:t}$  and  $z_{1:t}$  as  $p_t(m) := p(m | x_{1:t}, z_{1:t})$ . Additionally, unobserved grid cells are assigned a uniform prior of being occupied.

We represent the occupancy status of grid cell  $m^i$  at time

$t$  with a log odds expression

$$l_t := \log \frac{p(m^i | z_{1:t})}{p(\bar{m}^i | z_{1:t})} \quad (1)$$

where  $\bar{m}^i$  denotes the probability that  $m^i$  is unoccupied. When a new observation  $z_t$  is obtained, the log odds update is given by

$$l_t = l_{t-1} + \log \frac{p(m^i | z_t)}{p(m^i)} - \log \frac{p(\bar{m}^i | z_t)}{p(\bar{m}^i)} \quad (2)$$

where the last two terms represent the inverse sensor model.

## III. INFORMATION-THEORETIC OBJECTIVE

The goal of active perception exploration is to find a dynamically feasible sequential set of actions over a time interval,  $\tau := t+1, \dots, t+T$ , which enable the robot to explore its environment. We use the criteria that an explorative action is one that allows the robot to position itself in locations that generate observations that are informative to the robot's map. Under this criteria, choosing the optimal exploration action will maximally reduce the uncertainty in the robot's map. More precisely, an *action* can be defined as a discrete sequence of states,  $x_\tau = [x_{t+1}, \dots, x_{t+T}]$ . While executing an action, the robot will obtain a set of measurements  $z_\tau(x_\tau) = [z_{t+1}(x_{t+1}), \dots, z_{t+T}(x_{t+T})]$  by sensing from the states  $x_\tau$ .  $z_\tau(x_\tau)$  is modeled as a random variable whose distribution is parameterized by a deterministic action,  $x_\tau$ . In our formulation, we choose to select actions from a library of motion primitives,  $\mathcal{X}$ , which are generated by a planner. Under these notations, an explorative planner must determine  $x_\tau^*$ : the action that visits locations which allow the robot to obtain the set of measurements which maximally reduce uncertainty in its current map.

To determine  $x_\tau^*$ , we follow Charrow et al. [2] and maximize the Cauchy-Schwarz Quadratic Mutual Information (CSQMI) rate between the current map and the expectation of future measurements collected along an action,  $I_{CS}[m; z_\tau | x_\tau]$ .

$$x_\tau^* = \operatorname{argmax}_{x_\tau \in \mathcal{X}^T} \frac{I_{CS}[m; z_\tau(x_\tau)]}{R(x_\tau)} \quad (3)$$

Here,  $R : \mathbb{R}^{|x_\tau|} \rightarrow \mathbb{R}^+$  computes the estimated time required to complete action  $x_\tau$ . We choose to maximize CSQMI as opposed to more common metrics, such as Shannon's mutual information (MI), due to the property that CSQMI can be computed exactly in  $\mathcal{O}(N^2)$ , and closely in  $\mathcal{O}(N)$ , when  $N$  is the number of occupancy grid cells intersected by all rays in  $z_\tau$ . In contrast, MI requires averaging many  $\mathcal{O}(N^2)$  Monte Carlo samples. The full MI and CSQMI solutions are remarkably similar when computed over an occupancy grid map, further motivating CSQMI's use. CSQMI between the map and collection of measurements from  $\tau$  is given in (4).

$$\begin{aligned} I_{CS}[m; z_\tau] &= \\ &- \log \frac{(\sum \int p(m, z_\tau) p(m) p(z_\tau) dz_\tau)^2}{\sum \int p^2(m, z_\tau) dz_\tau \sum \int p^2(m) m^2(z_\tau) dz_\tau} \end{aligned} \quad (4)$$

where sums are over possible enumerations of  $m$ , and integrals are over the space of measurements that can be observed during  $\tau$ . CSQMI is non-negative, and maps two random variables to a real valued quantity which represents the information that

one learns about each variable by learning the other. CSQMI is equal to zero if and only if its arguments are independent.

We represent measurements as  $B$ -tuple random variables, such that a measurement  $z_k^b$  captured at time  $k \in \tau$  contains  $B$  beams. Most occupancy grid measurement models assume that cells not intersected by a laser beam are not updated in the map. Let  $c$  be the set of cells intersected by laser beam  $z_k^b$ , and let  $C = |c|$ . Then  $I_{\text{CS}}[m; z_k^b] = I_{\text{CS}}[c; z_k^b]$ . Charrow et al. [2] derive a closed form solution to the CSQMI between a single laser beam and the robot's map.

$$\begin{aligned} I_{\text{CS}}[c; z_k^b] &= \log \sum_{l=0}^C w_l \mathcal{N}(0, 2\sigma^2) \\ &+ \left( \log \prod_{i=1}^C (o_i^2 + (1 - o_i)^2) \right. \\ &\quad \left. \sum_{j=0}^C \sum_{l=0}^C p(e_j) p(e_l) \mathcal{N}(\mu_l - \mu_j, 2\sigma^2) \right) \\ &- 2 \log \sum_{j=0}^C \sum_{l=0}^C p(e_j) w_l \mathcal{N}(\mu_l - \mu_j, 2\sigma^2) \end{aligned} \quad (5)$$

Here,  $p(e_j)$  is the probability that the  $j$ th cell in the raycast  $c$  is the first occupied cell, and  $o_i = p(c^i = 1)$ , the probability that  $i$ th cell in the raycast is occupied. Weights  $w_{l \in [1, C]}$  can be pre-computed over the raycast to avoid duplication, and are evaluated as

$$w_l = p^2(e_l) \prod_{j=l+1}^C (o_j^2 + (1 - o_j)^2) \quad (6)$$

By assuming that the resolution of the occupancy grid is greater than the variance of the range sensor, which is typically the case in occupancy grid mapping scenarios, and by assuming that Gaussians approach zero with growth in  $|\mu_l - \mu_j|$ , one may assert that the inner sum in the double sum terms is only non-zero when  $l$  is close to  $j$ . In other words, the  $\mathcal{O}(C^2)$  double sum terms can be reduced to  $\mathcal{O}(C)$  complexity by assuming

$$\begin{aligned} \sum_{j=0}^C \sum_{l=0}^C \alpha_{j,l} &\approx \sum_{j=0}^C \sum_{l=j-\delta}^{j+\delta} \alpha_{j,l} \\ \sum_{j=0}^C \sum_{l=0}^C \beta_{j,l} &\approx \sum_{j=0}^C \sum_{l=j-\delta}^{j+\delta} \beta_{j,l} \end{aligned} \quad (7)$$

where  $\delta \ll C$ , and  $\alpha_{j,l}$  and  $\beta_{j,l}$  are the weighted Gaussian terms inside of the double sums in (5). Our simulator and robot's laser scanners have  $\sigma \approx 0.1$  cm, and use occupancy grid sizes of 10.0 cm, allowing us to choose  $\delta = 1$  with minimal loss of information.

#### IV. MEASUREMENT MODEL

The CSQMI of a single beam (5) depends on the ability to estimate  $p(e_j)$ , the probability that the  $j$ th cell in a raycast is the first occupied grid cell. Rather than building a measurement model using the maximum likelihood estimate of  $z_j$  [2, 3, 8], we compute a discrete distribution over the raycast, where each cell value directly approximates the probability that cell  $c^i$  is

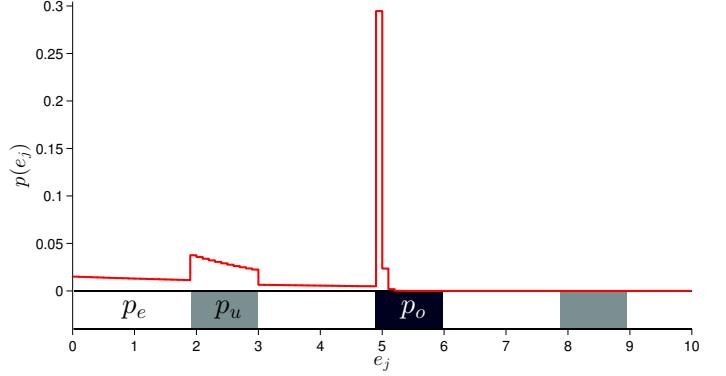


Figure 2: The distribution  $p(e_j)$  over a depicted 1-dimensional map, where  $p_e = 0.01$ ,  $p_o = 0.92$ ,  $p_u = 0.05$ .

the first occupied cell in  $c$ ,  $p(e_j = c^i) \forall c^i \in c$  using the continuous occupancy values in  $c$ .

To calculate this distribution, we use a generalization of the geometric distribution, which is commonly used to determine the number of Bernoulli trials necessary to obtain one success, supported on the set  $\mathbb{N}^+$ . Since each occupancy grid cell in a raycast contains continuous probabilities  $\in [0, 1]$  of the cell's occupancy, we build on the geometric distribution to support independent, but not identically distributed (i.n.i.d.) random variables. Using this generalization, the probability that a cell  $c^i$  terminates the raycast is computed by assuming that the ray passed through all previous cells.

$$p(e_j = c^i) = o^i \prod_{j=1}^{i-1} (1 - o^j) \quad (8)$$

A straightforward implementation can compute (8) in  $\mathcal{O}(C^2)$  operations for all  $c^i \in c$ . However, we were able to derive an efficient recursive formula that runs in  $\mathcal{O}(C)$ .

$$\begin{aligned} p(e_j = c^i) &= o^i \prod_{j=1}^{i-1} (1 - o^j) \\ &= o^i \left( \frac{o^{i-1}}{o^{i-1}} \right) (1 - o^{i-1}) \prod_{j=1}^{i-2} (1 - o^j) \\ &= o^i \frac{1 - o^{i-1}}{o^{i-1}} p(e_j = c^{i-1}) \\ &= o^i \left( \frac{1}{o^{i-1}} - 1 \right) p(e_j = c^{i-1}) \end{aligned} \quad (9)$$

The distribution  $p(e_j)$  is depicted over a one-dimensional raycast in Fig. 2.

#### V. CLOSED-LOOP RRT

To make use of this information cost function to guide exploration, we consider a sampling-based planning approach that can evaluate the predicted information gain throughout the environment. In addition, we wish to use the occupancy grid that is being updated online (as described in Sect. II) to guide the vehicle around obstacles in the environment. The RRT algorithm is well suited to planning paths through these types of large environments, and is described below.

The planner starts from the vehicle's current state and samples a point  $x_s$  in the environment. Using the occupancy grid, we can reject samples that lie in cells with a sufficiently high probability of being occupied. If the sample is valid, we find the best node in the tree of paths (initially just the vehicle state), as determined by the cost function. We then add a new edge to the tree connecting the sample point to the nearest node. Then a new sample is drawn and the process repeats to grow a tree of path segments through the environment. This tree-growing process terminates after a specified time, making this an anytime approach, and the minimum cost path is returned. Specifically, we consider a variant of the anytime RRT algorithm known as Closed-loop RRT (CL-RRT) [5], in which the edges are generated by forward simulating the closed-loop vehicle dynamics toward the sample point. This approach is traditionally used to ensure dynamic feasibility and dense collision checking. However, the forward simulation also yields full state information for the system at the end of each segment. This allows us to evaluate the predicted information gain (4) at that point and assign a corresponding cost to the candidate trajectory.

However, CL-RRT is traditionally a goal-directed planner. Since we are primarily interested in exploring the environment, we modify CL-RRT to function as an exploration-driven planner. We first define the sampling distribution to be a Gaussian centered about the root of the tree, with no bias toward any direction (unlike standard sampling-based planners that will sample the goal some small probability). We also do not include any distance information in the node cost, in order to favor pure exploration. Finally, since there is no goal to guide the selection of the best branch from the tree, we simply select the branch with the minimum cost endpoint in the entire tree. This enables the planner to compute paths that aim to maximize the predicted information gain while keeping the robot in the free space,  $\mathcal{X}_{\text{free}}(t)$  (defined by the current occupancy map). The planner continuously replans at a set rate and selects the best path from the tree if it has a lower cost than the previously selected path (or if the old path is now infeasible due to map updates). Before this check, the cost of the old path is updated using the current map to capture changes in the expected information gain. Planning terminates when a user-defined condition is met (e.g., the entropy rate of the map falls below a threshold). The resulting modified CL-RRT algorithm is presented in Alg. 1.

## VI. STATE ESTIMATION

State estimation addresses the problem of determining the robot's pose in the environment given noisy sensor observations. The state estimation pipeline's pose output is a superior alternative to directly feeding in exteroceptive sensor observations to the controller and planner. In this section, we present an Unscented Kalman Filter for fusing inertial measurement unit (IMU) and localization observations for a 2D robot.

### A. Process Model

The vehicle state  $\mathbf{x}$  consists of the global position  $\mathbf{p} = [p_x \ p_y]^T$ , global heading angle  $\theta$ , global velocity  $\mathbf{v} = [v_x \ v_y]^T$ , IMU angular velocity bias in the  $z$ -direction  $b_\omega$

---

### Algorithm 1 CL-RRT

---

```

1:  $t \leftarrow 0, x(t) \leftarrow x_{\text{initial}}$ 
2: Initialize tree with node at  $x_{\text{initial}}$ 
3: while explore == true do
4:    $t_{\text{start}} \leftarrow t$ 
5:   for  $t - t_{\text{start}} <$  planning time do
6:     Sample point  $x_s$  from the environment
7:     Select min-cost node from  $n$  nearest in tree
8:      $k \leftarrow 0$ 
9:      $\hat{x}(t+k) \leftarrow$  last state at  $n$ 
10:    while  $\hat{x}(t+k) \in \mathcal{X}_{\text{free}}(t), \hat{x}(t+k) \neq x_s$  do
11:      Compute command to drive system to  $x_s$ 
12:      Simulate system, get end state  $\hat{x}(t+k+1)$ 
13:      Compute cost based on (4)
14:       $k \leftarrow k + 1$ 
15:    end while
16:    Store final  $x_{\text{hat}}$  as new node in tree
17:  end for
18:  Get min-cost path from tree
19:  Re-evaluate old path cost on current map
20:  if new path cost < old path cost then
21:    Set new path
22:  end if
23:   $t \leftarrow t +$  planning time
24: end while

```

---

and IMU linear acceleration biases in the  $x$ - and  $y$ -directions  $\mathbf{b}_a = [b_{ax} \ b_{ay}]^T$ .

$$\dot{\mathbf{p}} = \mathbf{v} \quad (10)$$

$$\dot{\theta} = \omega - b_\omega - n_\omega \quad (11)$$

$$\dot{\mathbf{v}} = \mathbf{C}(\theta) (\mathbf{a} - \mathbf{b}_a - \mathbf{n}_a) \quad (12)$$

$$\dot{b}_\omega = n_{b\omega} \quad (13)$$

$$\dot{\mathbf{b}}_a = \mathbf{n}_{ba} \quad (14)$$

The IMU measurements are comprised of the  $z$ -direction angular velocity  $\omega$  as well as the body frame  $x$ - and  $y$ -direction linear accelerations  $\mathbf{a} = [a_x \ a_y]^T$ . Both measurements are modelled as being corrupted by additive Gaussian white noise and a random walk bias driven by Gaussian white noise (13), (14).

$$\mathbf{n} = \begin{bmatrix} \mathbf{n}_a \\ n_\omega \\ \mathbf{n}_{ba} \\ n_{b\omega} \end{bmatrix} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}) \quad (15)$$

$$\mathbf{Q} = \text{diag}\{\sigma_a^2, \sigma_a^2, \sigma_\omega^2, \sigma_{ba}^2, \sigma_{ba}^2, \sigma_{b\omega}^2\} \quad (16)$$

Since the IMU measurements are in the body frame, we use a  $2 \times 2$  rotation matrix  $\mathbf{C}(\theta)$  to rotate them into the global reference frame (12). Each component of the noise vector in (15) is assumed to be independent, and their corresponding covariance sigma values are chosen by offline sensor characterization tests (16).

### B. Correction Model

The laser scans are used to construct a map of the environment, and a grid-based localization algorithm is used to compute the global pose (position and heading) of the vehicle

with respect to the map. The sensor model for the localization algorithm is given by

$$\mathbf{z} = \begin{bmatrix} \mathbf{p} \\ \theta \end{bmatrix} + \begin{bmatrix} \mathbf{C}(\theta) & \mathbf{0} \\ \mathbf{0}^T & 1 \end{bmatrix} \mathbf{w} \quad (17)$$

$$\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}) \quad (18)$$

The observation noise covariance matrix  $\mathbf{R}$  (18) is computed by fitting a multivariate Gaussian to the posterior probability grid of the localization algorithm. Because this covariance is expressed in the scanner frame (assumed to be coincident with the body frame), we rotate its associated noise vector  $\mathbf{w}$  into the world frame in (17).

### C. Unscented Kalman Filter

The nonlinearities introduced by the rotation in the process and correction models motivated the choice of the Unscented Kalman Filter in this project. For brevity, we omit the complete presentation of all the steps involved in the UKF (we refer the reader to [4]). We apply the process model and correction model equations when the relevant measurement is received by the state estimator. Outlier exteroceptive observations are rejected by a chi-squared innovation gate.

## VII. RESULTS

To validate the proposed exploration strategy, we implemented the formulations and algorithms discussed in Sections III-VI in a high-fidelity C++ simulation environment. The simulator assumes a ground robot constrained to a 2D horizontal plane, which is equipped with a noisy laser scanner (range = 30 m,  $\sigma = 0.1$  cm), and IMU. The robot's motion is governed by a 13-state nonlinear model of skid-steer dynamics controlled by velocity and yaw rate commands. The velocity command is computed using a proportional-derivative (PD) feedback law on the position error between the robot's current position and current reference point. The yaw rate command is computed based on proportional feedback on the error between the robot's current heading angle and the angle to the reference point. Our simulator generates horizontal laser scans from a meshed point cloud input, and generates IMU observations according to the true state of the robot. Prior to evaluating CSQMI or building an RRT, we populate a map using recent laser scans and localize with respect to it using a custom SLAM implementation that was developed prior to this project. Our SLAM implementation leverages ICP for laser odometry [6], a histogram filter for localization [8], and a custom 3D mapping framework.

To evaluate performance of the proposed exploration framework, we consider two simulation environments, shown in Fig. 3. Figure 4 shows a series of snapshots of the robot navigating the smaller map. The robot is initially unaware of the map layout and must construct it while driving. The occupancy grid is updated at 10 Hz, and the planner replans online at 1 Hz to use the updated map information. This enables it to successfully plan around obstacles as they are detected. Moreover, the tree of candidate trajectories that are considered at each planning iteration accurately identifies regions of the map that have not yet been explored and therefore would result in large information gains (indicated by the green nodes in the tree). Regions of the map that have

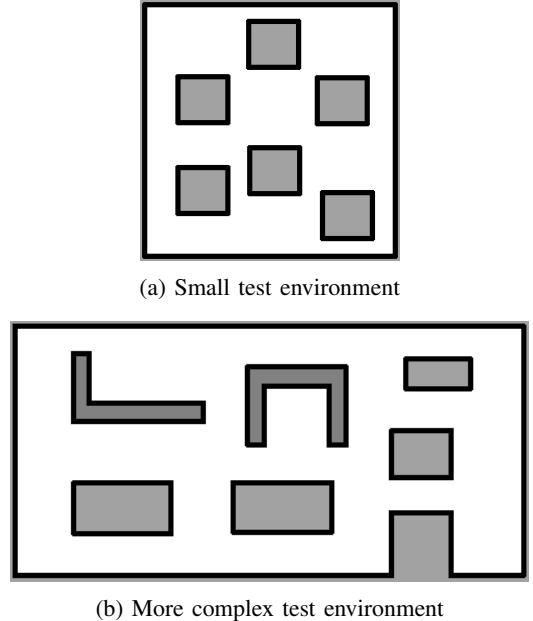


Figure 3: Layouts of the environments used to test the proposed exploration technique.

already been observed will yield significantly less information, and this approach accurately captures this effect (indicated by the red nodes in the tree). We observe similar performance on the larger environment, as illustrated in Fig. 5. Due to the speed of the RRT-based approach, the planner is able to consider candidate paths extending into a large portion of the environment at each step, allowing it to automatically identify and explore the most salient regions even in an expansive and more complex environment.

## VIII. CONCLUSION AND FUTURE WORK

In this work, we have presented a novel approach for information-based exploration and validated performance of this approach using a high-fidelity simulation environment of our ground robot platform. These results demonstrate that this information-based planning strategy is able to accurately and automatically identify regions of the environment that will yield the most information, plan feasible paths to these regions, and successfully explore the environment.

While our approach can be used to successfully produce maps of large environments, it can have difficulty escaping local minima where the RRT's paths do not extend far enough into new unknown territories. In future work, we will adaptively modify the RRT's parameters and planning time to enable consideration of longer trajectories when the map's entropy rate is reduced.

Having verified the exploration strategy in simulation, we would also like to experiment with it using a ground robot (Fig. 1b). We currently have the UKF and SLAM running on the ground robot (see map and state estimate in Fig. 1a), so the next steps would be to transition and integrate the planning and control software. In addition, it would be useful to implement the strategy on aerial robots to see if it easily extends to 6D configuration spaces.

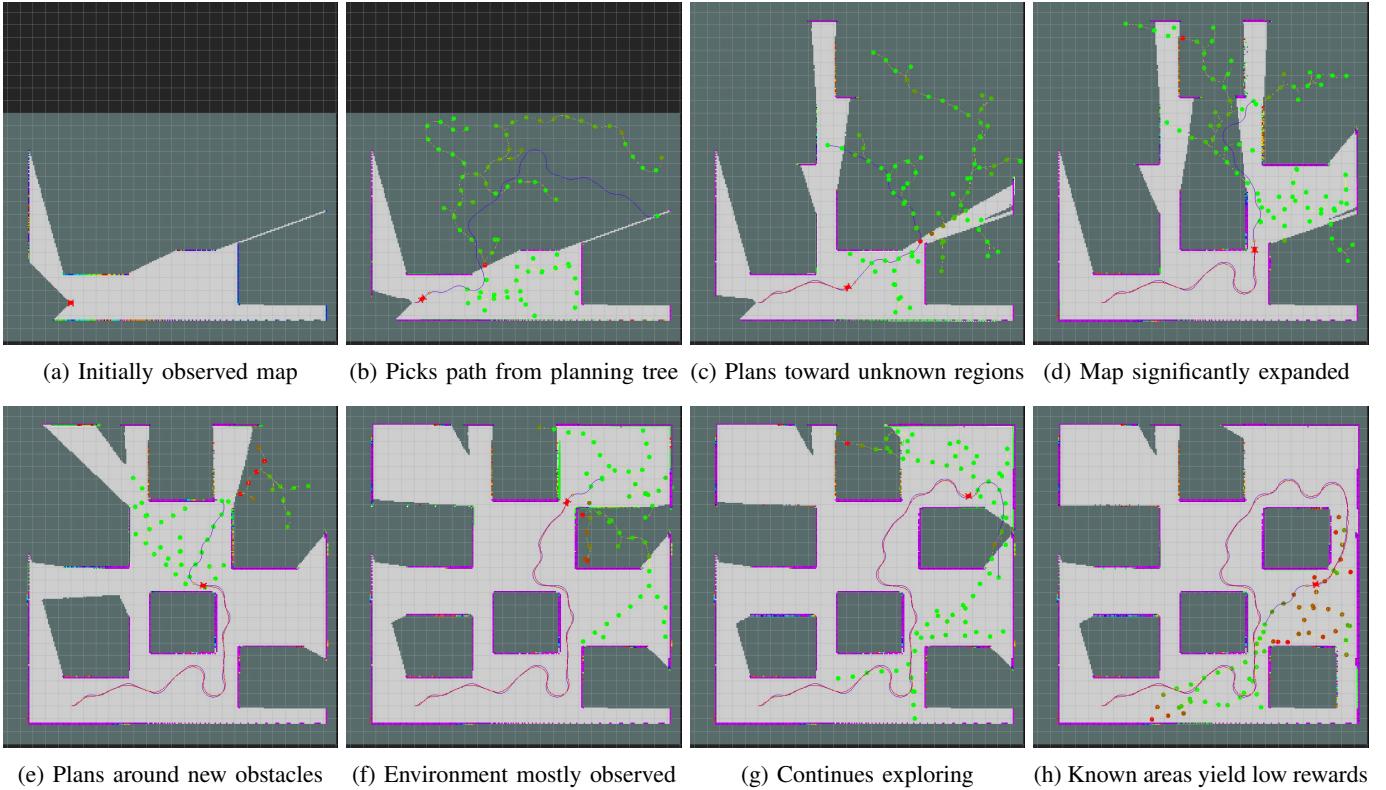
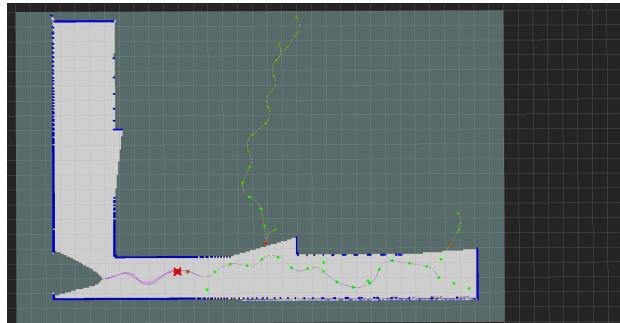


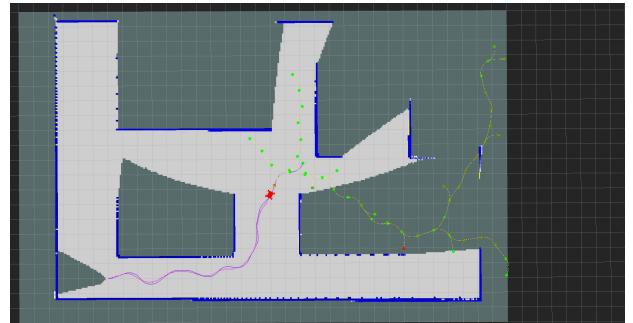
Figure 4: Snapshots of the ground robot exploring an initially unknown environment using information gain to drive planning. The nodes in the tree are colored according to information gain (green = high, red = low).

## REFERENCES

- [1] F. Bourgault, A. A. Makarenko, S. B. Williams, B. Grocholsky, and H. F. Durrant-Whyte. Information based adaptive robotic exploration. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 1, pages 540–545. IEEE, 2002.
- [2] B. Charrow, S. Liu, V. Kumar, and N. Michael. Information-theoretic mapping using cauchy-schwarz quadratic mutual information. technical report. *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.
- [3] B. Julian, S. Karaman, and D. Rus. On mutual information-based control of range sensing robots for mapping applications. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5156–5163. IEEE, 2013.
- [4] S. J. Julier and J. K. Uhlmann. A new extension of the kalman filter to nonlinear systems. In *Proc. SPIE*, volume 3068, pages 182–193, July 1997.
- [5] Y. Kuwata, S. Karaman, J. Teo, E. Frazzoli, J. P. How, and G. Fiore. Real-time motion planning with applications to autonomous urban driving. *IEEE Transactions on Control Systems Technology*, 17(5):1105–1118, Sepetember 2009.
- [6] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat. Comparing icp variants on real-world data sets. *Autonomous Robots*, 34(3):133–148, 2013.
- [7] C. Stachniss, G. Grisetti, and W. Burgard. Information gain-based exploration using rao-blackwellized particle filters. In *Robotics: Science and Systems*, volume 2, 2005.
- [8] S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics*. MIT press, 2005.
- [9] Brian Yamauchi. A frontier-based approach for autonomous exploration. In *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pages 146–151. IEEE, 1997.



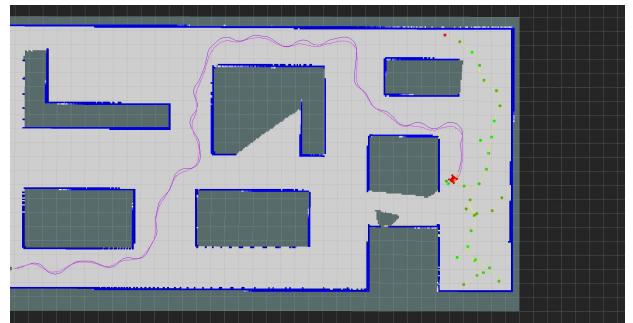
(a) Initial exploration guides robot down hallway



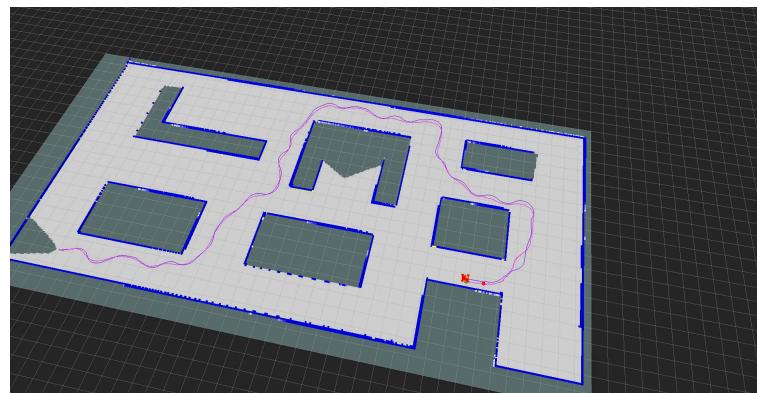
(b) High information-gain nodes added near unknown regions



(c) Planner expands into new regions as the map expands



(d) Robot has autonomously mapped most of the environment



(e) Final map

Figure 5: Snapshots of the ground robot exploring a large unknown environment.