Autonomous Robotic Exploration Using Occupancy Grid Maps and Graph SLAM Based on Shannon and Rényi Entropy

Henry Carrillo, Philip Dames, Vijay Kumar, and José A. Castellanos

Abstract— **In this paper we examine the problem of autonomously exploring and mapping an environment using a mobile robot. The robot uses a graph-based SLAM system to perform mapping and represents the map as an occupancy grid. In this setting, the robot must trade-off between exploring new area to complete the task and exploiting the existing information to maintain good localization. Selecting actions that decrease the map uncertainty while not significantly increasing the robot's localization uncertainty is challenging. We present a novel information-theoretic utility function that uses both Shannon's and Rényi's definitions of entropy to jointly consider the uncertainty of the robot and the map. This allows us to fuse both uncertainties without the use of manual tuning. We present simulations and experiments comparing the proposed utility function to state-of-the-art utility functions, which only use Shannon's entropy. We show that by using the proposed utility function, the robot and map uncertainties are smaller than using other existing methods.**

I. Introduction

Autonomous exploration is a high level task that is required for many real-world robotic applications. The primary goal is to acquire the most complete and accurate map of an environment in a finite time. Constructing a high quality map requires the robot to maintain good localization, even while traversing unknown space where by definition the uncertainty of the robot grows. Considering a grid map representation of the environment, this autonomous exploration task can be divided into three general steps:

1) identification of possible locations to explore in the current estimate of the map
2) evaluation of the utility of possible actions
3) execution of the action with the highest utility.

In the first step, we would ideally evaluate every possible action in the robot and map space. However, this proves to be computationally intractable in real applications [1]. In practice, we select a small subset of locations in the map based on their local information, using techniques such as frontier-based exploration [2] where destinations are the boundaries between known and unknown space.

In the second step, the robot computes the utility of performing each of the candidate actions from the first step. We would ideally use the full joint distribution of the map $\mathbf{m}$ and robot poses $\mathbf{x}$ before $(P(\mathbf{x}, \mathbf{m}))$ and after $(P(\mathbf{x}, \mathbf{m}|\mathbf{u}, \mathbf{z}))$ taking the candidate action $\mathbf{u}$ and receiving measurements $\mathbf{z}$. We compute some measure of information gain for each action using these prior and posterior distributions. One major problem is that computing the aforementioned joint probability analytically is, in general, intractable [3], [4]. In practice, an approximation of the joint probability is used, such as assuming that the map and robot uncertainties are independent [5] or conditionally independent [3]. These approaches often rely on a heuristic linear combination of the robot and map uncertainties [6]–[9]. A caveat of the above is that the scale of the numerical values of the two uncertainties is not comparable, i.e. the map's uncertainty is often orders of magnitude larger than the robot's uncertainty, requiring the user to manually tune the weighting parameters.

In this paper, we assume that the map is represented by an occupancy grid, one of the most common approaches, and that a graph SLAM system is used, the state-of-the-art SLAM solution. We propose a novel utility function that drives the robot towards areas of high expected reduction of uncertainty within the map, as measured by Shannon's entropy. We discount the information value of visiting a region if we expect the robot to have high uncertainty in its pose. We do this using the Rényi entropy, linking the parameter of the Rényi entropy to the predicted uncertainty in the robot's localization.

II. State-of-the-Art Utility Functions for Autonomous Robotic Exploration

The literature on autonomous robotic exploration is diverse, dating back more than 30 years. We focus on recent approaches that use an information-theoretic framework, as these most closely relate to our proposed utility function. We discuss the underlying assumptions, and associated shortcomings, of these utility functions.

A. Overview

In this section, we focus on step 2 of the autonomous exploration task outlined in Sec. I. We focus on the uncertainty terms of the utility functions, leaving out other auxiliary costs such as energy consumption or heuristic travel costs.

The seminal work in autonomous exploration is from Yamauchi [2]. However, this initial work assumes perfect knowledge of the robot's pose, so the utility function depends only on the map. Bourgault, et al. [6] pioneer the use

HC and JC gratefully acknowledge funding from MINECO-FEDER project DPI2012-36070, research grants BES-2010-033116 and EEBB-2011-44287, and DGA Grupo(T04). PD and VK gratefully acknowledge funding from AFOSR Grant FA9550-10-1-0567, ONR Grants N00014-14-1-0510, N00014-09-1-1051, and N00014-09-1-103, and NSF Grant IIS-1426840. PD is the recipient of a National Defense Science and Engineering Graduate Fellowship.

PD and VK are with the GRASP Laboratory, University of Pennsylvania, Philadelphia, USA. {pdames, kumar}@seas.upenn.edu. JC is with the *Universidad de Zaragoza*, Zaragoza, Spain. jacaste@unizar.es. HC is with the *Depto. de Ingeniería Electrónica, Pontificia Universidad Javeriana*, Bogotá, Colombia. h.carrillo@javeriana.edu.co

of entropy-based utility functions for autonomous robotic exploration, using a convex combination of the map and the robot's pose entropies. They compute the posterior uncertainty of the map assuming that the robot has no error when executing actions, using (15) from [6]:

$$\mathbb{H}[P(\mathbf{m} \mid \mathbf{x}, \mathbf{d})] \approx -\sum_{i,j} P(m_{ij}) \log(P(m_{ij})) + (1 - P(m_{ij})) \log(1 - P(m_{ij})), \quad (1)$$

where m_{ij} is the Bernoulli random variable associated with cell ij of the occupancy grid map and $P(m_{ij})$ is the probability of that cell being occupied. However, a robot with high uncertainty in its pose may incorrectly clear a large number of cells in the map, yielding a less accurate map while reducing the map entropy according to (1). To account for this, Bourgault, et al. use a scalar function of the robot's pose covariance matrix given by the SLAM system proposed in [10]. One drawback of this approach is that the scale of the map's uncertainty is orders of magnitude larger than the scale of the robot's uncertainty, so a change in the robot's pose uncertainty has a negligible effect on the value of the utility function. To take the above into account, Bourgault, et al. propose a utility function that is a linear weighted combination of the robot's and map's uncertainty, but the weights are set in a heuristic fashion. Carlone, et al [4] and Stachniss [11, Ch. 7] show that such entropy-based strategies outperform the frontier-based method of Yamauchi [2].

Stachniss, et al. [3] propose a utility function for Rao-Blackwellized particle filter based SLAM systems. Based on several conditional independence assumptions, the utility function is a linear sum of the entropy of the robot's poses and the expected entropy of the possible maps associated with each particle:

$$\mathbb{H}[P(\mathbf{x}, \mathbf{m} \mid \mathbf{d})] \approx \mathbb{H}[P(\mathbf{x} \mid \mathbf{d})] + \sum_{i=1}^{\#P} \mathbf{w}^{[i]}\, \mathbb{H}[P(\mathbf{m}^{[i]} \mid \mathbf{x}^{[i]}, \mathbf{d})] \quad (2)$$

where $\#P$ is the number of particles, $\mathbf{w}^{[i]}$ is the likelihood of particle i, and $\mathbf{d} = (\mathbf{u}, \mathbf{z})$ is the history of data, i.e. the control inputs and received measurements. Using the standard assumption that the robot's pose is represented using a Gaussian distribution, the first term in (2) is:

$$\mathbb{H}[P(\mathbf{x} \mid \mathbf{d})] = \frac{n}{2}(1 + \log(2\pi)) + \frac{1}{2} \log \det \mathbf{\Sigma} \quad (3)$$

where n is the dimension of the robot's pose and $\mathbf{\Sigma}$ is the $n \times n$ covariance matrix. This approach suffers from the same issue in the relative scales of the entropy values, as discussed in [8] and [4].

Blanco, et al. [8] and Carlone et al. [4] present utility functions that make the entropy computations independent of the occupancy grid size and only consider cells seen by the robot. However, these approaches are restricted to particle filter based SLAM systems, which are known not to scale as well as graph-based approaches with the map size. Our approach most closely resembles that of Carlone, et al. [4], discounting the information gain of an action based on the probability of having good localization of the robot.

Valencia, et al. [5] and Kim, et al. [9] use graph-based SLAM systems [12], assuming that the mean of the map and the robot's poses is close to the maximum likelihood estimate [13, Ch. 11], [12]. Valencia, et al. represent the map as an occupancy grid and Kim, et al. as a set of features. Both assume that the uncertainty in the map and the robot's pose are independent, so the posterior entropy is the weighted sum of the individual entropies. These weights are chosen heuristically to trade-off between exploration and exploitation.

B. Remarks

The utility function from [5] takes the form:

$$\mathbb{I}_G[\mathbf{a}, \hat{\mathbf{z}}] = \underbrace{\mathbb{H}[P(\mathbf{x}, \mathbf{m} \mid \mathbf{d})]}_{\text{current entropy}} - \underbrace{\mathbb{H}[P(\mathbf{x}, \mathbf{m} \mid \mathbf{d}, \mathbf{a}, \hat{\mathbf{z}})]}_{\text{future/predicted entropy}} \quad (4)$$

where $\mathbf{a} = a_{1:T}$ is a candidate action with time horizon T and $\hat{\mathbf{z}} = z_{1:T}$ are the maximum likelihood sensor measurements received while executing action $\mathbf{a}$. These measurements are usually called "hallucinated" measurements. The set of future measurements $\hat{\mathbf{z}}$ is commonly computed via approximate ray-casting techniques in conjunction with a plausible sensor model, as in [1], [3]. This reduces the computational complexity by not considering all possible combinations of measurements for an action $\mathbf{a}$.

To select an action, the typical procedure is to greedily optimize (4) over the set of possible actions:

$$\mathbf{a}^* = \arg\max_{\mathbf{a}} \underbrace{\mathbb{H}[P(\mathbf{x}, \mathbf{m} \mid \mathbf{d})]}_{\text{current entropy}} - \underbrace{\mathbb{H}[P(\mathbf{x}, \mathbf{m} \mid \mathbf{d}, \mathbf{a}, \hat{\mathbf{z}})]}_{\text{future/predicted entropy}} \quad (5)$$

Valencia, et al. [5] noted that the starting position of the actions is the same, so the first term in (5) is equal for all actions. This simplifies the objective to:

$$\mathbf{a}^* = \arg\min_{\mathbf{a}} \underbrace{\mathbb{H}[P(\mathbf{x}, \mathbf{m} \mid \mathbf{d}, \mathbf{a}, \hat{\mathbf{z}})]}_{\text{predicted joint entropy}} \quad (6)$$

$$\approx \arg\min_{\mathbf{a}} \underbrace{\mathbb{H}[P(\mathbf{x} \mid \mathbf{d}, \mathbf{a}, \hat{\mathbf{z}})]}_{\text{pose entropy}} + \underbrace{\mathbb{H}[P(\mathbf{m} \mid \mathbf{x}, \mathbf{d}, \mathbf{a}, \hat{\mathbf{z}})]}_{\text{map entropy}} \quad (7)$$

The utility function (4) is usually computed using (1) and (3), as in [3], [5], [14]. As mentioned previously, the scales of the map and pose entropy values are very different, i.e. $\mathbb{H}[P(\mathbf{x} \mid \mathbf{d}, \mathbf{a}, \hat{\mathbf{z}})] \ll \mathbb{H}[P(\mathbf{m} \mid \mathbf{x}, \mathbf{d}, \mathbf{a}, \hat{\mathbf{z}})]$, effectively neglecting the effect of the robot's pose uncertainty. To illustrate this, consider a simple example scenario in which a robot explores a 10 × 10 m environment with an occupancy grid resolution of 0.05 m. Let us assume that only 1% of the cells are unknown, i.e. the probability of occupancy is 0.5, and the remaining cells are known perfectly, i.e. the probability of occupancy is 0 or 1. Using (1), the entropy of the map is 400 bits. Let the pose of the robot, which consists of a 2D position and orientation, be represented by a Gaussian distribution with a spherical covariance matrix. If the standard deviation of the robot is equal to the environment size (10 m), then, using (3), the entropy is only 16.1 bits. To have entropy equal to the map entropy requires the robot's pose to have a standard deviation of 3.32×10^{39}.

Given this, all the above optimization problems from (5) to (7) are equivalent in practice and effectively neglect the uncertainty in the robot's localization. The heuristic weighting from [6], [7] can overcome this, but requires careful manual tuning. Given the aforementioned scale problem, in practice the utility (7) reduces to:

$$\mathbf{a}^* \approx \arg\min_{\mathbf{a}} \underbrace{\mathbb{H}[P(\mathbf{m} \mid \mathbf{x}, \mathbf{d}, \mathbf{a}, \hat{\mathbf{z}})]}_{\text{predicted map entropy}} \tag{8}$$

Note that computing (8) requires a full update of the occupancy grid, a time consuming operation. Early approaches to robotic exploration using information theory, such as [6], [7], turn this into the equivalent problem of placing the sensor over regions of maximum entropy in the current map. This avoids the need to update the map using unknown future measurements, and the objective becomes:

$$\mathbf{a}^* \approx \arg\max_{\mathbf{a}} \sum_{m \in \mathbf{m}(\mathbf{a})} \underbrace{\mathbb{H}[P(m \mid \mathbf{x}, \mathbf{d})]}_{\text{current entropy}} \tag{9}$$

where $\mathbb{H}[P(m \mid \mathbf{x}, \mathbf{d})]$ is the current entropy of cell m and $\mathbf{m}(\mathbf{a})$ is the set of cells that the robot may see by taking action $\mathbf{a}$. It is also possible to speed up the computation of future measurements by using a plausible sensor model and only considering the cells in $\mathbf{m}(\mathbf{a})$ as in [3].

III. Proposed Utility Function

Entropy is a measure of the uncertainty of a random variable [15], [16]. The most commonly accepted definition is from Shannon [15]. Rényi generalized Shannon's definition of entropy in [16], to be:

$$\mathbb{H}_\alpha[p(\mathbf{x})] = \frac{1}{1-\alpha} \log_2 \left(\sum_{i=1}^{n} P_i^\alpha \right) \tag{10}$$

where $\mathbf{x}$ is a discrete variable with possible outcomes $\mathbf{x}_1, \ldots, \mathbf{x}_r$ and P_i is the probability associated with the outcome $\mathbf{x}_i$; in our framework the random variable is the occupancy of each cell $P(m_{ij})$, which can have two possible outcomes: 0 (free space) and 1 (obstacles). The variable $\alpha \in [0, 1) \cup (1, \infty)$ is a free parameter, though in this work we restrict our attention to the range $(1, \infty)$.

The α parameter in (10) has an intuitive, geometrical interpretation. Consider the simplex formed by all possible discrete probability distribution functions over a set of random variables. The Rényi entropy with parameter α is related to the α norm of a point in that vector space, i.e. a probability distribution. See [17, Ch. 2] for a more complete description. Shannon entropy is a special case of Rényi entropy, in the limit as $\alpha \to 1$ [17].

A. A Shannon and Rényi Based Utility Function

Our proposed utility function is:

$$\mathbf{a}^* = \arg\max_{\mathbf{a}} \sum_{m \in \mathbf{m}(\mathbf{a})} \underbrace{\mathbb{H}[p(m \mid \mathbf{x}, \mathbf{d})]}_{\text{Shannon entropy}} - \underbrace{\mathbb{H}_{\alpha(\mathbf{a})}[P(m \mid \mathbf{x}, \mathbf{d})]}_{\text{Rényi entropy}} \tag{11}$$

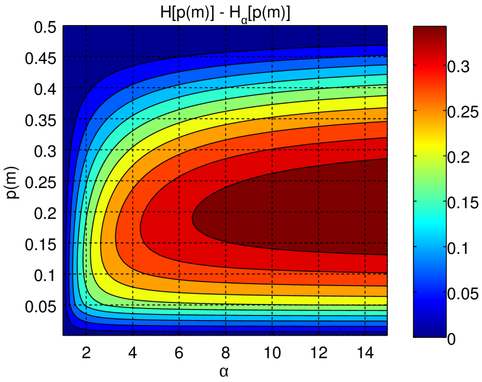


Fig. 1. Value of the utility function for a Bernoulli random variable m. The figure is mirrored for $P(m) \in [0.5, 1]$.

where $\alpha = \alpha(\mathbf{a})$ depends on the uncertainty in the robot's pose after taking action $\mathbf{a}$. As in (9), the entropy is computed using only the cells that the robot will "see" by carrying out the action. The key difference between our utility function and (9) is the Rényi entropy, which is used to discount the information value of the robot visiting a cell depending on its localization uncertainty.

Intuitively, the Shannon entropy term in (11) is an optimistic measure of the map uncertainty, assuming the robot will not experience localization errors when carrying out action $\mathbf{a}$. For example, imagine a scenario where a robot has significant drift in its odometry and it is not possible to perform a loop closure to correct for this. If the robot continues to explore, the SLAM system may yield poor localization and map estimates. However, the map uncertainty computed using (1) will likely decrease because the robot will continue to visit unexplored cells. Conversely, (11) combines the two uncertainties in a unified manner without any heuristic tuning, linking the uncertainty in the map and the robot's pose.

Given the above interpretation, and the relationship [17]:

$$\mathbb{H}[P(x)] > \mathbb{H}_\alpha[P(x)] \geq \mathbb{H}_{\alpha'}[P(x)] \geq 0, \quad 1 < \alpha \leq \alpha' \tag{12}$$

the proposed utility function (11):

1) is non-negative;
2) is bounded from above by the Shannon entropy and from below by zero;
3) monotonically decreases with α.

Note that the first property complies with the Shannonian belief that information does not hurt, i.e. adding more measurements does not increase entropy in expectation.

B. The Parameter α

We relate the parameter $\alpha(\mathbf{a})$ in (11) to the predicted uncertainty of the robot after taking action $\mathbf{a}$. When the robot has perfect localization, i.e. minimal uncertainty, then the information gain should be maximal and when the robot is completely lost, i.e. maximal uncertainty, then the information gain should be zero. In other words, we want $\alpha \to 1$ as the uncertainty becomes infinite since the two entropies in (11) cancel out, and $\alpha \to \infty$ as the uncertainty approaches zero, since this minimizes the Rényi entropy.

Fig. 1 shows the value of (11) for a Bernoulli random variable m representing the probability of occupancy of a single cell. In the case that $P(m) \approx 0$, meaning we have (nearly) perfect information about the cell, then a single action will have little effect on the estimate, and uncertainty of the robot should not matter. This is reflected in the cost function, where the information gain is approximately zero for all α.

In the case that $P(m) \approx 0.5$ we have (nearly) no information about the cell. Since we have an uninformative prior, any measurement may be considered a good measurement, and the information gain again does not depend upon the robot uncertainty.

When we have a little bit of information about a cell, i.e. $0.05 \lesssim P(m) \lesssim 0.45$, then the robot uncertainty is most important, as an incorrect measurement (due to poor localization of the robot, not sensor noise) will contradict prior information, increasing the uncertainty about the map. In other words, this case should have the largest dependence on the uncertainty of the robot, which we see is true from Fig. 1.

We want the scalar parameter α to have a monotone dependence on the uncertainty of the robot's localization. A simple candidate relationship is:

$$\alpha = 1 + \frac{1}{\sigma} \tag{13}$$

where σ is a scalar related to the predicted uncertainty of the robot's pose after taking action $\mathbf{a}$. More complex relationships should be the focus of further study, but we will show in Sec. V that (13) is sufficient to demonstrate the efficacy of our proposed utility function. The computation of σ is left until Sec. IV as it has an indirect connection to the proposed utility function.

IV. Autonomous Exploration Framework

In this section we detail a framework for robot-based autonomous exploration. Our goal is to acquire a map of an unknown environment in a finite time, which implicitly requires the robot to maintain a good estimate of its pose.

We assume that the robot has a SLAM system running, simultaneously estimating the robot's pose and generating an occupancy grid of the environment from the collected sensor data. Our SLAM front-end is an Iterative Closest Point (ICP)-based laser scan matcher [18]. Our SLAM back-end is the Incremental Smoothing and Mapping (iSAM) library [19], which builds a pose graph using the laser odometry to constrain consecutive pose nodes. In our framework, each node in the pose graph also has an associated set of features extracted from the laser scan taken at that location. These features are computed using the Fast Laser Interest Region Transform (FLIRT) [20]. We compare the feature sets from different nodes using a modified RANSAC algorithm, adding loop closure constraints if the match is sufficiently strong.

We also assume that the robot has a navigation system capable of driving it to any accessible point in the environment. We use the algorithm proposed in [21], which takes the most recent laser scan, inflates all obstacles by a specified margin, and then drives the robot towards the point in free space that is closest to the goal location. While this method works well to avoid local obstacles, the robot often gets stuck when the goal location is far away in the map, e.g. when driving down a series of corridors with multiple turns. To avoid this issue, the robot plans a path in the current map and intermediate waypoints are passed to the navigation routine, replanning if any of the waypoints are found to be inaccessible.

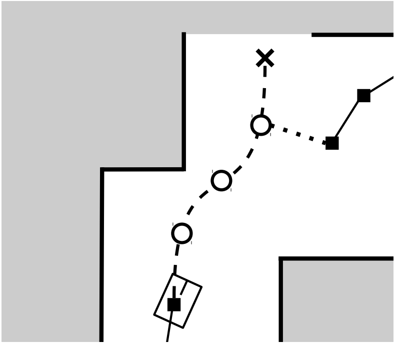


Fig. 2. Candidate path through the environment. The occupancy grid map is shown in the background, with white representing free space, black is occupied space, and grey is unknown. The robot is at the bottom and the dashed line is a path through free space to the frontier goal location, denoted with the X at the top. Black squares indicate existing iSAM nodes while the hollow circles indicate potential new nodes along the path. The dotted line is a potential loop closure between the candidate path and a previous node in the pose graph.

Our framework is divided in three high-level steps, as outlined in Sec. I. We describe our approach to these tasks in the remainder of this section.

A. Identification of Candidate Destinations

In order to generate goal locations, the robot computes exploration frontiers [2] from the latest occupancy grid map. A cell of the occupancy grid is labeled as being at a frontier if it is unoccupied, adjacent to an unknown cell, and not adjacent to an occupied cell. Such cells are clustered in the map to generate frontier regions, with the goal locations being the mean positions of the cells within each cluster.

We create an action plan $\mathbf{a}$ for each of these goal locations. An action plan is a set of waypoints, in free space, that lead the robot from its current location to a goal: $\mathbf{a} = \{(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)\}$. We use the AD^* algorithm from the SBPL library [22] to create the action plans, using the current occupancy grid.

B. Evaluation of Action Utilities

The robot then computes the utility of each candidate action according to (11), in which every term can be computed using standard ray-casting techniques such as the one described in [3]. The main characteristic of our utility function is the parameter α, which depends on the future uncertainty of an action plan. As discussed in Sec. III-B the parameter α is related to the predicted uncertainty of the robot during an action $\mathbf{a}$ through (13). This raises two questions: how to get a good approximation of the robot's

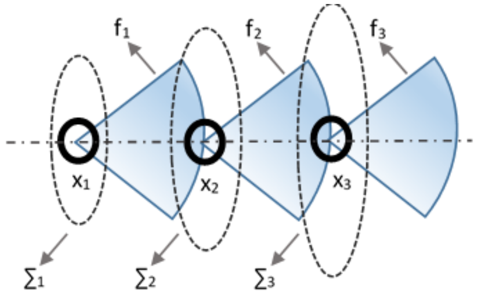


Fig. 3. The figure depicts an action plan with three steps. The mean estimated pose ($\mathbf{x}_k$) of the robot, the footprint of a laser sensor attached to the robot (f_k), and the covariance matrix ellipsoid of the robot's pose ($\mathbf{\Sigma}_k$) are shown at each step.

localization uncertainty during an action $\mathbf{a}$ and then how to extract a meaningful uncertainty scalar σ from it.

1) Uncertainty Prediction: To address the first question we utilize iSAM. An initial factor is placed at the robot's current estimated location, with the covariance matrix taken from the most recent node in the graph. We interpolate the action plan $\mathbf{a}$ with some fixed step size and add pose nodes along the length of the path, adding odometry constraints between them.

If the action plan takes the robot near other existing nodes in the graph, then the robot has the potential to close a loop. To take this into account, the robot adds additional factors for any nearby nodes in the pose graph that have a sufficiently high number of FLIRT features, i.e. the areas that have a great deal of structure which the robot can use to localize itself. The position and covariance matrix of these factors are taken from the iSAM graph of the full map, and we add constraints based on the transformation between the existing and potential nodes.

We use iSAM to optimize the miniature pose graph created from the action plan $\mathbf{a}$. The computational overhead is minimal since the actions consist of only 10's of nodes. Fig. 2 illustrates this process.

2) Uncertainty Scalar: We compute the uncertainty scalar σ using the covariance matrix estimates from the action pose graph. From the Theory of Optimal Experimental Design (TOED) [23]–[25], there are several standard optimality criteria that map a covariance matrix to a scalar while retaining useful statistical properties. The three most widely used criteria are:

- A-optimality (*A-opt*) minimizes the average variance,

$$\frac{1}{n}\mathrm{trace}(\mathbf{\Sigma}) = \frac{1}{n}\sum_{k=1}^{n}\lambda_k \tag{14}$$

  where n is the dimension of the covariance matrix $\mathbf{\Sigma}$ and λ_k is its kth eigenvalue.
- D-optimality (*D-opt*) minimizes the volume of the covariance matrix ellipsoid,

$$\det(\mathbf{\Sigma})^{1/n} = \exp\left(\frac{1}{n}\sum_{k=1}^{n}\log(\lambda_k)\right). \tag{15}$$

- E-optimality (*E-opt*) minimizes the maximum eigenvalue of the covariance matrix, $\mathbf{\Sigma}$,

$$\max(\lambda_k). \tag{16}$$

These criteria can be applied to the covariance matrix estimate from the iSAM solution of the action plan graph, using either the full covariance matrix or the marginal covariance matrices from each node in the graph. If the full covariance matrix is used, there is a single α value along the entire path. Then (11) is applied to the subset of the map visited by the robot when executing action $\mathbf{a}$. The use of marginal covariance matrices is more subtle, as different nodes may have different α values. For each cell $m \in \mathbf{m}(\mathbf{a})$, the robot finds the last node j from which the cell was visible and uses that α_j to compute the information gain (11) in that cell. In this way, the information gain is discounted using the uncertainty of the robot when viewing each individual cell. This is illustrated in Fig. 3.

Note that actions may consist of a variable number of waypoints, depending on the distance through the map to the frontier. Longer paths allow the robot to explore more area at the expense of higher uncertainty, unless the robot is able to close a loop. *The proposed approach implicitly penalizes long paths due to the expected increase in the pose uncertainty, balancing this with the potential to gain more information by viewing a larger area.*

C. Execution of Actions

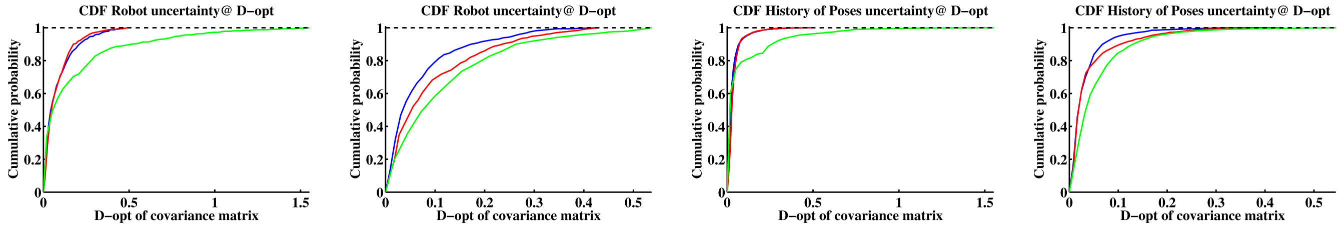
The robot carries out the action with the maximum information gain, using the navigation algorithm from [21]. It is possible to experience a failure in the navigation or SLAM system while executing an action. If the failure occurs within a predetermined time of the beginning of the action $\mathbf{a}$, the next best action $\mathbf{a}'$ is executed, without recomputing the action set. In this case, a small neighborhood around the final destination of the faulty action $\mathbf{a}$ is blacklisted until it is clear that the goal is accessible to the robot. This prevents the robot from repeatedly attempting an impossible action.

V. Simulated Experiments

We perform a series of experiments in the cave-like and Autolab 2D environments in Fig. 5, both of which are available in the Radish repository [26]. The code is written in C++ using ROS and run on a computer with an Intel i7 processor, 8GB of RAM, and running 64-bit Ubuntu 12.04. Let `SH` denote the standard utility function (4) based on Shannon entropy. For our proposed utility function, we test two optimality criteria for the σ computation: *A-opt* (`A`) and *D-opt* (`D`). Both use the full covariance matrix for each action plan.

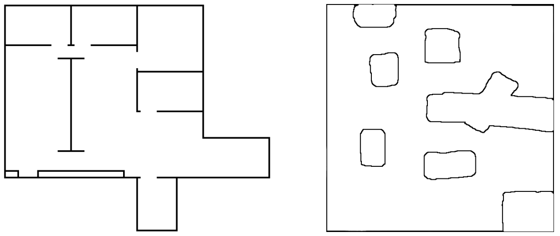
A. Experimental Procedure

We start the robot from five different locations in each environment and give the robot a 3 minute time budget to complete the exploration task. With more time the robot would explore the whole environment, but it is more interesting to compare the different utility functions under the same



(a) Autolab (b) Cave (c) Autolab (d) Cave

Fig. 4. CDF of the uncertainty in the robot pose at every time step of every simulation (a-b) and CDF of the running average uncertainty of the robot pose over each simulation (c-d). In this figure SH strategy is colored in green, A strategy in blue, and D in red.



(a) Autolab (b) Cave

Fig. 5. Maps used in the simulation experiments.

conditions. In the accompanying multimedia material there are snapshots of the resulting occupancy grids for some of the simulation and hardware experiments.

The robot is a differential drive platform with a maximum velocity of 0.4 m/s. The robot is equipped with a laser scanner with a maximum laser range of 10 m, field of view of 270°, and 1080 beams per scan. We use our own kinematic and laser simulators. The odometry noise is Gaussian with a standard deviation of 5 cm for every m in the (x, y) position and 1° for every 45° in the orientation. The laser noise is also Gaussian, with a standard deviation of 1 cm.

The robot begins each experiment with no information about the environment. For each trial, we record the uncertainties of the current pose and the history of poses and the percentage of area correctly explored at every time step. We measure the uncertainty of a covariance matrix using *D-opt* , i.e. based on the determinant of the covariance matrix, as suggested in [24].

B. Presentation of the Results

One problem with evaluating robot-based autonomous exploration experiments is that trials may have radically different trajectories, making pairwise comparisons difficult. Moreover, presenting just the mean or the median of the trials can be misleading. Inspired by the solution of a similar problem [18], we summarize the results of the experiments using the cumulative distribution function (CDF) of the metrics of interest. The CDF provides a richer representation of the variability of the results than the mean or median, while avoiding misleading anecdotal evidence due to noise.

For each parameter of interest, e.g. uncertainty in the robot's pose, we compute the CDF from a histogram of the available data at each time step. The bin size is automatically set using the procedure described in [27]. To more clearly quantify the differences between CDFs, we extract three point estimates: the 50^{th}, 75^{th} and 95^{th} percentiles. For metrics in which a high value implies worse performance, e.g. uncertainty or translational error, we would like a CDF that reaches 1 as quickly as possible.

TABLE I

50^{th}, 75^{th} AND 95^{th} PERCENTILES OF THE CDFS IN FIG. 4.

		Autolab			Cave		
		$\mathbb{P}50$	$\mathbb{P}75$	$\mathbb{P}95$	$\mathbb{P}50$	$\mathbb{P}75$	$\mathbb{P}95$
Robot pose	A	0.048	0.121	0.302	0.040	0.092	0.262
	D	0.051	0.121	0.257	0.057	0.139	0.308
	SH	0.072	0.252	0.808	0.087	0.175	0.380
Pose history	A	0.023	0.039	0.116	0.022	0.043	0.106
	D	0.031	0.048	0.111	0.022	0.043	0.169
	SH	0.026	0.066	0.411	0.039	0.077	0.181

C. Simulation Results

1) CDFs of Uncertainty: Fig. 4 shows the CDFs of the pose uncertainty in each test environment. Fig. 4a and Fig. 4b show the uncertainty in the robot's pose at each individual node in the pose graph as computed by iSAM, measuring the worst-case performance of the exploration strategies. Fig. 4c and Fig. 4d show the running average uncertainty over the history of robot's poses, measuring the average performance over a trial. Tab. I shows the percentiles of the CDF. Overall, our proposed utility function has a lower uncertainty in both environments. The robot using our utility function with *A-opt* results in 49.70% less uncertainty in the pose at the 75^{th} percentile than the robot using Shannon entropy. This difference is still large (46.84%) at the 95^{th} percentile.

2) Percentage of Explored Area: Evaluating the percentage of a map correctly identified by a robot during exploration is inherently a classification problem: we want to determine if each cell is correctly identified as being free or occupied. However, the number of free cells is typically much greater than the number of occupied cells. So a robot may return a large number of matched free cells even if there are very few matches of the occupied cells.

To avoid this bias towards free space, we measure the map accuracy by independently estimating free and occupied cells. The percentage of the map correctly explored is computed as the balanced accuracy (BAC), a concept borrowed from the machine learning community [28]. The BAC equally weights the accuracy of the estimates of the free and occupied cells:

$$\text{BAC} = \frac{1}{2}\left(\frac{\text{\# correct free cells}}{\text{\# total free cells}} + \frac{\text{\# correct occupied cells}}{\text{\# total occupied cells}}\right) \tag{17}$$

TABLE II

MEAN AREA CORRECTLY EXPLORED BY EACH EXPLORATION STRATEGY

	Autolab					Cave				
	Free cells		Occupied Cells		BAC	Free cells		Occupied Cells		BAC
	Accuracy	$\pm\sigma$	Accuracy	$\pm\sigma$	Accuracy	Accuracy	$\pm\sigma$	Accuracy	$\pm\sigma$	Accuracy
A	44.66 %	6.09 %	16.27 %	4.86 %	30.46 %	37.73 %	6.27 %	10.89 %	1.96 %	24.31 %
D	38.28 %	5.60 %	20.24 %	5.43 %	29.26 %	40.47 %	5.50 %	9.41 %	1.01 %	24.94 %
SH	40.49 %	7.95 %	8.77 %	5.41 %	24.63 %	40.68 %	5.50 %	2.61 %	0.61 %	21.64 %

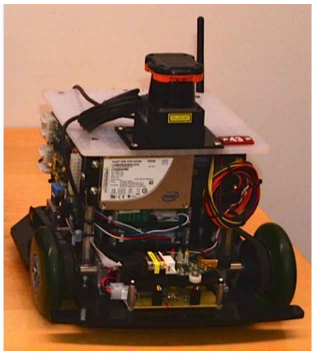


Fig. 6. Scarab platform used to perform the autonomous exploration task.

Tab. II shows the percentage of each map correctly explored by a robot using each utility function, with the proposed utility function performing favorably, particularly with respect to occupied cells.

VI. Hardware Experiments

We perform a series of experiments using the Scarab platform, shown in Fig. 6. The robot is equipped with a Hokuyo UTM-30LX laser scanner (30 m range and 270° field of view) and an on-board computer with an Intel i5 processor and 4GB of RAM running Ubuntu and ROS. The robot explores the office environment at the University of Pennsylvania shown in Fig. 7.

A. *Experimental Procedure*

The robot begins the autonomous exploration task at different starting locations within the environment and runs until it has explored the entire environment. The maximum velocity of the robot is set to 0.45 m/s. The robot runs the autonomous controller using its on-board computer.

The robot starts each experiment with no information about the environment. Since it was not possible to obtain ground truth for the trial, we cannot provide an insightful comparison against the Shannon based utility function. Nevertheless, the hardware experiments allow us to compare different parameterizations of our framework and check how they behave with real data.

B. *Results*

We compare the results of a robot using the A and D methods to a teleoperated exploration. Fig. 7 depicts example occupancy grids of the environment, after 5 minutes and after the robot has completed the exploration task.

Overall, the hardware experiments show the conservative behavior of our utility functions, with the robot retraversing known areas of the map in order to maintain good localization. The experiments with *A-opt* show that it is more conservative than *D-opt*, which agrees with the findings in [24]. The experiments also reveal the fact that the utility function is not robust to failure in the SLAM or navigation system. In others words, the utility function is not fault-tolerant to failures of the laser-based loop closure system, or to diffuse reflections of the laser scan that produce "phantom" exploration frontiers. This will be addressed in future work as it is necessary for truly autonomous robots.

VII. Conclusions

In this paper we present a novel information-theoretic utility function to select actions in an autonomous exploration task. The proposed utility function uses both the Shannon and Rényi definitions of entropy to automatically balance exploration and exploitation by considering the uncertainties in both the robot pose and the map. Our utility function has several key properties: it is non-negative, it is bounded from above by the Shannon entropy (i.e. the existing approaches), and monotonically decreases with the uncertainty in the robot's localization. These properties stem from the mathematical relationship between the Shannon and Rényi definitions of entropy.

Unlike previous attempts to define utility functions using a convex combination of the map and robot pose uncertainty, our utility function does not require any manual per case parameter tuning. Instead we rely on Rényi's definition of entropy to discount the expected information gain and directly relate the α parameter in (11) to the predicted uncertainty. This reduces to standard approaches when the localization uncertainty is eliminated.

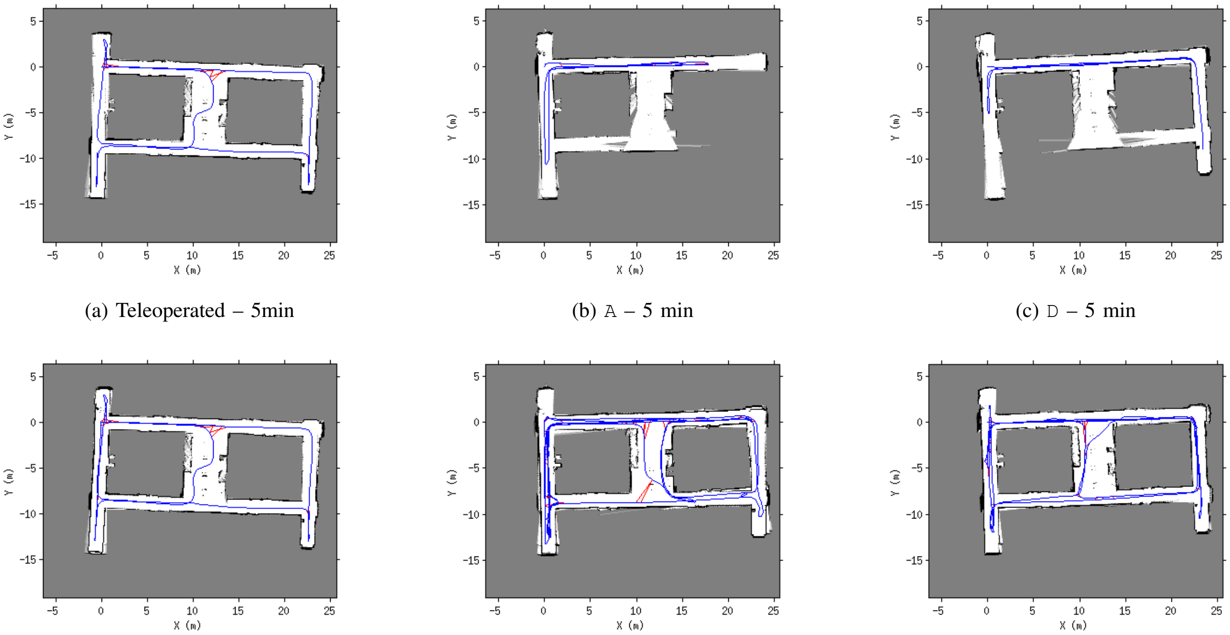
Our simulation and experimental results show a substantial reduction of the robot pose and map uncertainties when using our proposed utility function compared to the state of the art utility functions. This decrease in uncertainty is due to the exploitation of previous map information, resulting in more loop closures. However, exploration is more conservative and the predicted pose uncertainty used to compute α assumes that loops can be closed reliably. Clearly characterizing the reliability of the loop closure system, and the subsequent effects on the information gain, is an important direction for future work.

References

[1] W. Burgard, M. Moors, C. Stachniss, and F. Schneider, "Coordinated Multi-robot Exploration," *IEEE Transactions on Robotics (TRO)*, vol. 21, no. 3, pp. 376–386, Jun. 2005.

[2] B. Yamauchi, "Frontier-based Exploration Using Multiple Robots," in *Proceedings of the Second International Conference on Autonomous Agents*, ser. AGENTS '98. ACM, 1998, pp. 47–53.

[3] C. Stachniss, G. Grisetti, and W. Burgard, "Information Gain-based Exploration Using Rao-Blackwellized Particle Filters," in *Proceedings of Robotics: Science and Systems Conference (RSS)*, Cambridge, MA, USA, Jun. 2005.



(a) Teleoperated – 5min

(b) A – 5 min

(c) D – 5 min

(d) Teleoperated – 5 min 21 sec

(e) A – 30 min 57 sec

(f) D – 16 min 41 sec

Fig. 7. Examples of the resulting maps and pose graphs from hardware experiments. The robot is either teleoperated, or autonomously explores using the A and D methods from Sec. V. The blue edges indicate odometry constraints in the pose graph while red edges indicate loop closure events. (a–c) show the occupancy grids and pose graphs built by the robot after 5 min. (d–f) show the final occupancy grids and pose graphs.

[4] L. Carlone, J. Du, M. Kaouk, B. Bona, and M. Indri, "Active SLAM and Exploration with Particle Filters Using Kullback-Leibler Divergence," *Journal of Intelligent & Robotic Systems*, vol. 75, no. 2, pp. 291–311, Oct. 2014.

[5] R. Valencia, J. Vallve, G. Dissanayake, and J. Andrade-Cetto, "Active Pose SLAM," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2012, pp. 1885–1891.

[6] F. Bourgault, A. Makarenko, S. Williams, B. Grocholsky, and H. Durrant-Whyte, "Information Based Adaptive Robotic Exploration," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2002, pp. 540–545.

[7] A. Makarenko, S. B. Williams, F. Bourgault, and H. F. Durrant-Whyte, "An Experiment in Integrated Exploration," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2002, pp. 534–539.

[8] J. Blanco, J. Fernndez-Madrigal, and J. Gonzalez, "A Novel Measure of Uncertainty for Mobile Robot SLAM with RaoBlackwellized Particle Filters," *The International Journal of Robotics Research (IJRR)*, vol. 27, no. 1, pp. 73–89, Jan. 2008.

[9] A. Kim and R. M. Eustice, "Perception-driven Navigation: Active visual SLAM for Robotic Area Coverage," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2013, pp. 3196–3203.

[10] H. J. S. Feder, J. J. Leonard, and C. M. Smith, "Adaptive Mobile Robot Navigation and Mapping," *The International Journal of Robotics Research (IJRR)*, vol. 18, no. 7, pp. 650–668, Jul. 1999.

[11] C. Stachniss, *Robotic Mapping and Exploration*. Berlin, Germany: Springer, 2009, vol. 55.

[12] G. Grisetti, R. Kuemmerle, C. Stachniss, and W. Burgard, "A Tutorial on Graph-based SLAM," *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31–43, Jan. 2010.

[13] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Boston, MA, USA: MIT Press, 2005.

[14] J. Du, L. Carlone, M. Kaouk, B. Bona, and M. Indri, "A Comparative Study on Active SLAM and Autonomous Exploration with Particle Filters," in *Proceedings of IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Jul. 2011, pp. 916–923.

[15] C. Shannon and W. Weaver, *The Mathematical Theory of Communication*, ser. Illinois Books. Champaign, IL, USA: University of Illinois Press, 1949.

[16] A. Rényi, "On Measures Of Entropy And Information," in *Proceedings of the 4th Berkeley Symposium on Mathematics, Statistics and Probability*, 1960, pp. 547–561.

[17] J. Principe, *Information Theoretic Learning: Rényi's Entropy and Kernel Perspectives*, ser. Information Science and Statistics. Berlin, Germany: Springer, 2010.

[18] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat, "Comparing ICP Variants on Real-world Data Sets," *Autonomous Robots (AR)*, vol. 34, no. 3, pp. 133–148, Apr. 2013.

[19] M. Kaess, A. Ranganathan, and F. Dellaert, "iSAM: Incremental Smoothing and Mapping," *IEEE Transactions on Robotics (TRO)*, vol. 24, no. 6, pp. 1365–1378, Dec. 2008.

[20] G. D. Tipaldi and K. O. Arras, "FLIRT-interest regions for 2D range data," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2010, pp. 3616–3622.

[21] J. Guzzi, A. Giusti, L. M. Gambardella, G. Theraulaz, and G. A. Di Caro, "Human-friendly Robot Navigation in Dynamic Environments," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, May 2013, pp. 423–430.

[22] M. Likhachev, "Search-Based Planning Library," https://github.com/sbpl/sbpl, accessed: 2014-10-15.

[23] F. Pukelsheim, *Optimal Design of Experiments*, ser. Classics in Applied Mathematics. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics (SIAM), 2006.

[24] H. Carrillo, I. Reid, and J. A. Castellanos, "On the Comparison of Uncertainty Criteria for Active SLAM," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, St. Paul, MN, USA, May 2012, pp. 2080–2087.

[25] H. Carrillo, Y. Latif, J. Neira, and J. A. Castellanos, "Fast Minimum Uncertainty Search on a Graph Map Representation," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vilamoura, Portugal, Oct. 2012, pp. 2504–2511.

[26] A. Howard and N. Roy, "Radish: The Robotics Data Set Repository," http://radish.sourceforge.net/, accessed: 2014-10-15.

[27] H. Shimazaki and S. Shinomoto, "A Method for Selecting the Bin Size of a Time Histogram," *Neural Computation*, vol. 19, no. 6, pp. 1503–1527, Jun. 2007.

[28] K. Brodersen, C. S. Ong, K. Stephan, and J. Buhmann, "The Balanced Accuracy and its Posterior Distribution," in *Proceedings of the International Conference on Pattern Recognition (ICPR)*, Aug. 2010, pp. 3121–3124.