CrossMark

Autonomous robotic exploration using a utility function based on Rényi's general theory of entropy

Henry Carrillo[1] · Philip Dames[2] · Vijay Kumar[3] · José A. Castellanos[4]

Abstract In this paper we present a novel information-theoretic utility function for selecting actions in a robot-based autonomous exploration task. The robot's goal in an autonomous exploration task is to create a complete, high-quality map of an unknown environment as quickly as possible. This implicitly requires the robot to maintain an accurate estimate of its pose as it explores both unknown and previously observed terrain in order to correctly incorporate new information into the map. Our utility function simultaneously considers uncertainty in both the robot pose and the map in a novel way and is computed as the difference between the Shannon and the Rényi entropy of the current distribution over maps. Rényi's entropy is a family of functions parameterized by a scalar, with Shannon's entropy being the limit as this scalar approaches unity. We link the value of this scalar parameter to the predicted future uncertainty in the robot's pose after taking an exploratory action. This effectively decreases the expected information gain of the action, with higher uncertainty in the robot's pose leading to a smaller expected information gain. Our objective function allows the robot to automatically trade off between exploration and exploitation in a way that does not require manually tuning parameter values, a significant advantage over many competing methods that only use Shannon's definition of entropy. We use simulated experiments to compare the performance of our proposed utility function to these state-of-the-art utility functions. We show that robots that use our proposed utility function generate maps with less uncertainty and fewer visible artifacts and that the robots have less uncertainty in their pose during exploration. Finally, we demonstrate that a real-world robot using our proposed utility function is able to successfully create a high-quality map of an indoor office environment.

Keywords Autonomous exploration · Graph SLAM · Entropy

✉ Henry Carrillo
henry.carrillo@usa.edu.co

Philip Dames
pdames@temple.edu

Vijay Kumar
kumar@seas.upenn.edu

José A. Castellanos
jacaste@unizar.es

1 Escuela de Ciencias Exactas e Ingeniería, Universidad Sergio Arboleda, Bogotá, Colombia

2 Department of Mechanical Engineering, Temple University, Philadelphia, PA, USA

3 GRASP Laboratory, University of Pennsylvania Philadelphia, PA, USA

4 Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza, Zaragoza, Spain

1 Introduction

Autonomous exploration using mobile robots is a high-level task that is required for many real robotic applications such as search and rescue (Dames and Kumar 2013), environmental modeling (Hollinger et al. 2011), and archaeological reconstruction (Eustice et al. 2006). The main purpose of performing robot-based autonomous exploration is to acquire the most complete and accurate map of an environment in a finite time. This task can be divided into three general steps:

1. The robot identifies areas to visit in its current estimate of the map and plans actions to visit each location.
2. The robot computes the utility of each candidate action and selects the action with the highest utility.
3. The robot carries out the selected action.

After completing an action, the robot returns to the first step and the cycle repeats.

In the first step, we would ideally evaluate every possible action for the robot in every possible map that it could be exploring. Not surprisingly, this proves to be computationally intractable in real applications (Burgard et al. 2005; Martinez-Cantin et al. 2009) due to the high dimensionality of the joint map and action space. Recently Berg et al. (2012), Indelman et al. (2015) presented continuous approximations to select destinations and actions, but the computational complexity of these approaches is still prohibitively high for real-time implementation. In practice, a robot often selects a small subset of candidate locations based on the information from its local neighborhood in the maximum likelihood map. One of the most common techniques for selecting the candidate destinations is frontier-based exploration, first introduced by Yamauchi (1998), in which frontiers are defined as the boundaries between known and unknown space.

In the second step, the robot computes the utility of performing each of the candidate actions. Autonomous exploration requires the robot to maintain good localization so that it can accurately construct the map from the collected sensor data. However, the robot must do this while traversing unknown space, where by definition the localization uncertainty of the robot grows, in order to cover the full environment. Thus, any measure of utility in step 2 must carefully balance the exploration of new areas with the exploitation of previous data in order to maintain a sufficiently accurate estimate of the robot's pose. The most common approach is to compute the expected information gain for each action (Stachniss et al. 2005; Fairfield and Wettergreen 2010; Carlone et al. 2014). We would ideally use the full joint distribution of the map $\mathbf{m}$ and the robot's 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 $\mathbf{z}$ to compute

the information gain. However, in practice this is computationally intractable. Instead it is common to approximate the joint probability, for example by assuming that the map and robot uncertainties are independent (Valencia and Valls Miró 2012) or conditionally independent (Stachniss et al. 2005). These approaches often rely on a heuristic linear combination of the robot and map uncertainties (Bourgault et al. 2002; Makarenko et al. 2002; Blanco et al. 2008; Kim and Eustice 2013, 2015). One caveat is that the scale of the numerical values of the map and robot uncertainties are not comparable, i.e., the map's uncertainty is often orders of magnitude larger than the robot's state uncertainty, requiring the user to manually tune the weighting parameters. We describe this phenomenon in greater depth in Sect. 2.

Finally, in the third step the selected action is carried out. The three step cycle is then repeated until a stopping condition is met. However, determining the optimal stopping condition is still an open problem, as Cadena et al. (2016) note. Another open problem in active SLAM is the inherent trade-off between the length of the action and the accuracy of the robot's pose and map estimates. Roy et al. (1999) pioneered the research in this topic, proposing methods to generate robot trajectories that reduce the pose uncertainty and the map uncertainty (Sim et al. 2004). More recently, researchers have utilized the Theory of Optimal Experimental Design (TOED) (Pukelsheim 2006) to select actions that reduce the uncertainty in the robot pose during autonomous exploration. Specifically, Zhang et al. (2014, 2015) used the A-optimality metric and Carrillo et al. (2012a) used the D-optimality metric. We will discuss these optimality metrics and the TOED further in Sect. 4.2.

1.1 Paper contributions

In this paper we propose a novel utility function that is based on a novel definition of mutual information from Jumarie (1990, Ch. 2). This definition utilizes Rényi's general theory of entropy (Rényi 1960, 1970), of which Shannon's entropy and Hartley's entropy are special cases. Jumarie's framework allows us to avoid using any manually-tuned, environment-dependent parameters in the utility function, so that the robot is able to adaptively trade-off between exploring new area to complete the task and exploiting the existing information to maintain good localization in a variety of environments.

Removing the need to manually tune parameters in a SLAM system is one of the key open challenges to improve the robustness of SLAM solutions (Cadena et al. 2016). Indelman et al. (2015) recently devised an interesting framework to plan continuous motions when there is uncertainty in the state of the robot. Indelman et al. (2015) use this framework to show how incorrectly tuning a utility function can degrade the performance of their framework and explain in great detail heuristic methods to manually tune several

parameters in order to boost performance. Our work seeks to remove as many of these parameters as possible and to base any heuristics on fundamental mathematical properties of the objective function in a way that is independent of the environment.

Another challenge in autonomous exploration is considering both the localization and map uncertainties in a principled manner. Blanco et al. (2008) and Carlone et al. (2014) both demonstrate that correctly accounting for these uncertainties in the utility function has a major impact on the quality of the resulting map. One key issue is that the pose of a robot is a continuous quantity while an occupancy grid map, the most common map representation used in SLAM solutions, is discrete. This fundamental difference leads to issue of numerical scaling in most current objective functions, which we describe in greater detail in Sect. 2. Blanco et al. (2008) and Carlone et al. (2014) both proposed solutions to mitigate this scaling issue but they only work for particle-filter based SLAM back ends. Such SLAM back ends have been supplanted by graph-based solutions in recent years due to their superior scalability and accuracy (Grisetti et al. 2010). Our proposed utility function works for any SLAM back end and an occupancy grid representation of the map (or any other representation that consists of a finite collection of discrete random variables). However, the method that we propose to predict the future localization uncertainty of the robot assumes a graph-based SLAM back end. In the future we plan to expand this method to work for Kalman filter-based and particle filter-based SLAM systems.

The SLAM framework that we use in this paper is composed by well-know methods that work in practical scenarios and that, when combined, allow us to experimentally validate our approach, as we will show in Sect. 6. The components are all open source and outlined in detail in Sect. 4. To the best of our knowledge, this exact combination of approaches has not been used before in the literature so we consider their combination a minor contribution of this paper. More importantly, our code will be available at https://bitbucket.org/hcarrillo and on the authors' websites. This will allow other researchers to use this SLAM framework, to validate our results, and to use our utility function as a benchmark against their own systems.

This paper is an extension of the work presented by Carrillo et al. (2015a) at ICRA. This paper contains experimental results in two new, larger environments and more detailed analysis of the results. We have added appendices containing proofs of relevant properties of the Rényi entropy and a definition of a probability simplex. We have also extended several sections to make the description of the utility function easier to follow and provide a more complete literature review.

Previous utility functions for Graph SLAM-based autonomous robotic exploration that do not account correctly for the uncertainties or that require manual, heuristic tuning

can be argued to "explain no more than they assume, and tell us no more than the data does" (Averbeck 2015). In this paper we argue that tuning-free utility functions are key to enabling true robot autonomy, and that the proposed utility function is a step in this direction.

1.2 Paper structure

We start by defining an autonomous exploration task and providing a brief overview of the related work on using entropy-based utility functions in Sect. 2. Next, in Sect. 3, we detail our proposed utility function and its properties. In Sect. 4 we present a framework for performing robot-based autonomous exploration using the proposed utility function. Then, in Sects. 5 and 6, we present the results of the proposed framework in several simulated and hardware experiments and compare several different parameterizations. Finally, we present our concluding remarks in Sect. 7.

2 State of the art of utility functions for autonomous robotic exploration

In this section, we provide an overview of several existing utility functions for autonomous robotic exploration. The literature on this topic is diverse, dating back more than 30 years. We focus on recent approaches that use an information-theoretic framework, as these have been the most successful and are most closely relate to our proposed utility function. We discuss the underlying assumptions, and associated shortcomings, of these utility functions when used in an autonomous robotic exploration task. We focus on the uncertainty terms of the utility functions, including the map's representation and the robot's localization. We leave out other auxiliary costs such as energy consumption or heuristic travel costs, e.g., distance from a power outlet (for recharging a battery), as these have a second-order effect while we are proposing a fundamentally different first-order approach. Adding in these auxiliary costs to our utility function will be the focus of future work.

2.1 Map and robot uncertainty models

In order to evaluate the uncertainty in the map and in the pose of the robot, we must first model these quantities in a probabilistic manner. We can then use entropy to measure of the uncertainty of a random variable (Rényi 1960; Shannon and Weaver 1949). The most commonly accepted definition is from (Shannon and Weaver 1949),

$$\mathbb{H}[P(\mathbf{x})] = -\sum_{i=1}^{n} p_i \log p_i, \tag{1}$$

Table 1 Example of scale problems in Shannon's entropy-based utility functions: the table shows different values of the terms of Shannon's entropy-based utility functions using different parameters of the occupancy grid

Trial	Environment size (m)	Grid resolution (m)	State of cells			Map entropy (bits)	Robot's pose entropy (bits)
			$p = 0.33$ (%)	$p = 0.5$ (%)	$p = 1$ (%)		
1	10 m × 10 m	0.05	1	99	0	400	10.70
2	10 m × 10 m	0.05	3	89	8	4127.76	10.70
3	10 m × 10 m	0.05	2	82	15	6289.55	10.70
4	10 m × 10 m	0.76	1	99	0	1.73	10.70
5	10 m × 10 m	0.76	3	89	8	17.86	10.70
6	10 m × 10 m	0.76	2	82	15	27.22	10.70
7	7.5 m × 7.5 m	0.05	1	99	0	225	10.12
8	7.5 m × 7.5 m	0.05	3	89	8	2321.9	10.12
9	7.5 m × 7.5 m	0.05	2	82	15	3537.9	10.12

The map is an occupancy grid and its entropy is calculated using (3). The pose of the robot consists of a 2D position and orientation, and it is represented by a Gaussian distribution with a spherical covariance matrix with a standard deviation equal to the environment size [e.g., (10, 10 m, 2π rad)]. Its entropy is shown for reference, and it is calculated using (2)

where $p_i = P(\mathbf{x} = \mathbf{x}_i)$ is an element of the probability distribution $P(\mathbf{x})$ of a discrete random variable $\mathbf{x}$, i.e., $p_i \geq 0$, $\forall i$ and $\sum_{i=1}^{n} p_i = 1$. Shannon also defined the differential entropy for continuous random variables by replacing the summation in (1) with an integral.

It is standard to assume that the robot's pose is represented using a multivariate Gaussian distribution. The (differential) Shannon entropy of the distribution is then given by the well known expression (Cover and Thomas 2012; Stachniss et al. 2005),

$$\mathbb{H}[P(\mathbf{x})] = \frac{n}{2}\left(1 + \log(2\pi)\right) + \frac{1}{2}\log\det\mathbf{\Sigma} \qquad (2)$$

where n is the dimension of the robot's pose and $\mathbf{\Sigma}$ is the $n \times n$ covariance matrix.

The map is represented by an occupancy grid, where each grid cell has associated with it an independent Bernoulli random variable m. $m = 0$ corresponds to the cell being empty and $m = 1$ corresponds to the cell being occupied. The Shannon entropy of the map distribution is then given by Stachniss et al. (2005)

$$\mathbb{H}[P(\mathbf{m})] \approx -\sum_{i,j} \big(P(m_{ij}) \log(P(m_{ij})) $$
$$ + \big(1 - P(m_{ij})\big) \log\big(1 - P(m_{ij})\big)\big), \qquad (3)$$

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.

It is worth noting that while the map and pose entropies are both measures of uncertainty about random variables, they are fundamentally different quantities. The differential

entropy of a continuous random variable, such as the pose of a robot, can take any real value while the entropy of a discrete random variable, such as a grid cell in a map, is strictly non-negative. In fact, as Jaynes (1957) notes, Shannon's differential entropy is not invariant under a change of variables or dimensionally correct. On a more practical note, in an autonomous exploration task these two entropies have vastly different numerical values. 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 (3), 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's pose is equal to the environment size (10, 10 m, 2π rad), then, using (2), the entropy is only 10.70 bits. To have entropy equal to the map entropy requires the robot's pose to have a standard deviation of 9.56×10^{57}. See Table 1 for further examples of this scaling issue.

2.2 Utility functions

A seminal work in autonomous exploration is from Yamauchi (1998), in which the author defines frontier-based exploration. However, since this initial work assumes the robot has perfect knowledge of its state, the utility function depends only on the map.

Bourgault et al. (2002) pioneered the use of entropy-based utility functions for autonomous robotic exploration. Such entropy-based methods generally evaluate the utility as:

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

where $\mathbf{a} = a_{1:T}$ is a candidate action with time horizon T, $\hat{\mathbf{z}} = z_{1:T}$ is the collection of sensor measurements received while executing action $\mathbf{a}$, and $\mathbf{d} = (\mathbf{u}, \mathbf{z})$ is the history of the data, i.e., the control inputs and received measurements. Carlone et al. (2014) and Stachniss (2009, Ch. 7) show that such entropy-based strategies outperform the frontier-based method of Yamauchi (1998).

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

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

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

where the second equality holds because, as Valencia and Valls Miró (2012) noted, the starting position for each action is the same so the current entropy is the same for every action.

However, computing the joint entropy of the posterior distribution is very difficult. In their pioneering work, Bourgault et al. (2002) assume that the map and robot pose are independent of one another and use a convex combination of the map and the robot's pose entropies. The posterior uncertainty of the map assumes that the robot has no error when executing actions. They compute this uncertainty of the map conditioned on the pose of the robot, $\mathbb{H}[P(\mathbf{m} \mid \mathbf{x}, \mathbf{d})]$, using (3), where $\mathbf{d} = (\mathbf{u}, \mathbf{z})$ is the history of the data, i.e., the control inputs and received measurements.

Stachniss et al. (2004, 2005) 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:

$$\mathbb{H}[P(\mathbf{x}, \mathbf{m} \mid \mathbf{d})] \approx$$

$$\mathbb{H}[P(\mathbf{x} \mid \mathbf{d})] + \sum_{i=1}^{n} \mathbf{w}^{[i]} \, \mathbb{H}[P(\mathbf{m}^{[i]} \mid \mathbf{x}^{[i]}, \mathbf{d})] \qquad (7)$$

where n 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. Valencia and Valls Miró (2012) and Kim and Eustice (2013, 2015) use graph-based SLAM systems, assuming that the mean of the map and the robot's poses is close to the maximum likelihood estimate (Thrun et al. 2005, Ch. 11). This simplifies the objective to:

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

Valencia and Valls Miró (2012) represent the map as an occupancy grid and Kim and Eustice (2013, 2015) 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. It is worth emphasizing that (8) is an approximation meant to reduce the computational complexity of the utility function, similar to that of Stachniss et al. (2005).

2.2.1 Scale problems in Shannon's entropy-based utility functions

The utility function (4), and the various approximations, are usually computed using (2) and (3) (Stachniss et al. 2005; Du et al. 2011; Valencia and Valls Miró 2012). As discussed in Sect. (2.1) and noted by Blanco et al. (2008) and Carlone et al. (2014), 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 negating the effect of the robot's pose uncertainty.

Given this, the optimization problems (7) and (8) are effectively equivalent to

$$\mathbf{a}^* \approx \arg \min_{\mathbf{a}} \underbrace{\mathbb{H}[P(\mathbf{m} \mid \mathbf{x}, \mathbf{d}, \mathbf{a}, \mathbf{z})]}_{\text{predicted map entropy}} \qquad (9)$$

Therefore, robots using the standard utility functions based on Shannon's entropy will neglect the effect of the robot's pose uncertainty. This will cause the robots to always prioritize exploring new areas while neglecting to exploit the current information about the map during exploration (Blanco et al. 2008; Carlone et al. 2014). This can have disastrous consequences when a robot estimate of its pose diverges, causing the robot to incorrectly clear a large number of cells in the map. This yields a less accurate map despite reducing the map entropy according to (9).

The heuristic weighting from Makarenko et al. (2002) or Bourgault et al. (2002) can overcome this scaling issue, but it requires careful manual tuning depending on the conditions of the scenario the robot is exploring. Even for the same environment, changing the number of cells in the occupancy grid (or equivalently the grid resolution) drastically changes the value of the map entropy in (3), as Table 1 shows. However, the robot's pose is independent of the grid. This means that the weighting parameters must be tuned again if the grid resolution changes, even if the physical environment is identical.

In order to illustrate the above critical cases for Shannon's entropy based utility functions, let us reconsider the example given in Sect. 2.1 with environments of different sizes and different grid resolutions. Table 1 shows different values of the terms of Shannon's entropy-based utility functions using different parameters of the occupancy grid. Each row shows the need for manual tuning to make the scale comparable

among the terms of the utility function. Moreover, there are some situations where the parameters of the occupancy grid change online (Blanco et al. 2008). Thus, even if a system is carefully tuned at the outset of exploration, it must be re-tuned to account for the new parameters. Examples of such situations are:

– The number of cells changes due to a variation in the size of the environment. This can happen if the bounds of the area to be explored are unknown. Trials 1 and 7 in Table 1 exemplify this situation, where the entropy values for each map are quite different, and are also significantly larger than the entropy of the robot's pose.
– The number of cells changes due to a variation in the resolution of the grid map. This can happen when using adaptive discretizations, such as OctoMap (Hornung et al. 2013), or to decrease the computational load. Trials 2 and 4 in Table 1 exemplify this situation.
– There is a rapid change of the state of the cells. This can happen due to a sensor malfunction. Trials 8 and 9 in Table 1 exemplify this situation.

All the above situations necessitate manually tuning parameters of a Shannon's entropy-based utility function in order to avoid scale problems in the entropy terms. The manual tuning is performed in practice by weighting the terms. This weighting needs to be set adaptively depending on the aforementioned conditions. In other words, for different scenarios or situations in the same scenario, the Shannon's entropy-based utility function may need to be manually tuned to provide an exploration-exploitation trade-off behavior.

Blanco et al. (2008) and Carlone et al. (2014) 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. (2014), discounting the information gain of an action based on the probability of having good localization of the robot.

2.2.2 A tip for speeding up the computation

Note that computing (9) requires a full update of the occupancy grid for each possible set of future measurements. This is computationally intractable in practice. One common approach to mitigate this is to use the maximum likelihood measurements for a given action. Such 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 sen-

sor model (Burgard et al. 2005; Stachniss et al. 2005). This reduces the computational complexity by not considering all possible combinations of measurements for an action $\mathbf{a}$.

Even with hallucinated measurements, updating the occupancy grid for each possible action can be a costly process. Early approaches to robotic exploration using information theory, such as Makarenko et al. (2002) or Bourgault et al. (2002), turn this into the equivalent problem of placing the sensor over regions of maximum entropy in the current map. This bypasses the need to update the map using unknown future measurements, and the objective becomes:

$$\mathbf{a}^* \approx \arg \max_{\mathbf{a}} \sum_{m \in \mathbf{m}(\mathbf{a})} \underbrace{\mathbb{H}[P(m \mid \mathbf{x}, \mathbf{d})]}_{\text{current entropy}} \qquad (10)$$

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 $\mathbf{a}$. The idea here is to select the action that causes the robot's sensor to cover as much of the uncertainty in the *current* map, as measured by entropy, as possible. Implicitly assumed here is that getting more observations about these maximum entropy regions of the map will allow the robot to reduce its uncertainty about the map. This agrees with the idea of Jaynes (1957, Sect. 6) who said that "the maximization of entropy is not an application of a law of physics, but merely a method of reasoning which ensures that no unconscious arbitrary assumptions have been introduced."

It is also possible to speed up the computation (9) by using a plausible sensor model and only updating the cells observed during the action, i.e., the cells in $\mathbf{m}(\mathbf{a})$ (Stachniss et al. 2005).

3 Utility function I: Rényi's entropy applied to autonomous robotic exploration

Rényi (1960, 1970) generalized Shannon's definition of entropy, given in (1), to be:

$$\mathbb{H}_\alpha[P(\mathbf{x})] = \frac{1}{1 - \alpha} \log_2 \left(\sum_{i=1}^{n} p_i^\alpha \right). \qquad (11)$$

Rényi's entropy is a parametric family of entropies which form the most general class of information measures that comply with Kolmogorov's probability axioms (Principe 2010, Sect. 2.2). In other words, Rényi's entropy is a mathematical formalization of an information metric that complies with the most commonly accepted axiomatic definitions of entropy (Aczél and Daróczy 1975; Rényi 1970; Xu 1998).

It is worth remarking that Rényi's entropy was not designed with a particular application in mind, but it is used
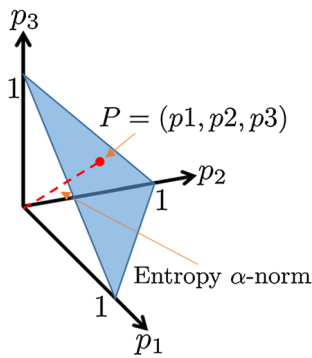
Fig. 1 The *blue* surface is the probability simplex in $\mathcal{R}^3$. The *red dot* is a particular distribution. The *dashed red line* shows the norm, or distance to the origin, of the distribution. In Rényi's entropy the α parameter defines the norm in the probability simplex (Color figure online)

in many fields, e.g., coding theory, statistical inference, quantum mechanics, and thermodynamics (Principe 2010). One of the reasons for its wide-spread use is the free parameter α, which yields a family of measures of uncertainty (or dissimilarity within a given distribution) Principe (2010, Sect. 2.2). For example, Shannon's entropy is a special case of Rényi entropy in the limit as $\alpha \rightarrow 1$ (Principe 2010; Rényi 1960) (Cover and Thomas 2012, Ch 17). See "Appendix A" for a proof of this fact and other properties.

The α parameter in (11) has an intuitive, geometrical interpretation. Consider the simplex formed by all possible discrete probability distribution functions over a set of random variables. Rényi's entropy with parameter α is related to the α norm of a point in that vector space, i.e., a probability distribution. Figure 1 illustrates this. The reader is referred to "Appendix B" and Principe (2010, Ch. 2) for a more complete description.

3.1 A Shannon and Rényi-based utility function

Using the definitions of entropy from Shannon and Rényi, Jumarie (1990, Ch. 2) presents a definition of mutual information (also known as transinformation):

$$\mathbb{I}_c[P(\mathbf{x})] = \underbrace{\mathbb{H}_{\alpha=1}[P(\mathbf{x})]}_{\text{Shannon's entropy}} - \underbrace{\mathbb{H}_{\alpha=c}[P(\mathbf{x})]}_{\text{Rényi's entropy with}} \tag{12}$$

which is the difference between the Shannon entropy of the probability distribution $P(\mathbf{x})$ and Rényi's entropy of the same distribution with $\alpha = c$. Jumarie (1990) goes on to describe the parameter α as a gain coefficient which measures the efficiency of an observer who is considering the distribution $P(\mathbf{x})$ and that (11) represents an information value for $\alpha > 1$ (as opposed to an uncertainty value for $\alpha < 1$).

Based on this interpretation, we define a new utility function for autonomous exploration:

$$\mathbf{a}^* = \arg\max_{\mathbf{a}} \mathbb{I}_{c(\mathbf{a})}[P(\mathbf{m} \mid \mathbf{x}, \mathbf{d})] \tag{13}$$

where $P(\mathbf{m} \mid \mathbf{x}, \mathbf{d})$ is the *current* distribution over possible maps and the value of the parameter $\alpha = c(\mathbf{a})$ depends on the possible future actions of the robot, $\mathbf{a}$. The key insight is that we tie the value of α, which, according to Jumarie (1990), is an efficiency measure of the observer, to the actions of the robot.

Note that computing (13) does not require updating the map using possible future measurements, which in turn would require propagating both the sensor and localization uncertainties forward in time. Correctly modeling and propagating these uncertainties is both difficult and computationally expensive because they are conditionally dependent given the map. Avoiding this forward propagation of uncertainty is a key advantage regarding computational speed of our proposed utility function.

To make the proposed utility function even more computationally tractable we can take the same approach as (Stachniss et al. 2005), as described in Sect. 2.2.2, and compute the utility functions only over the regions of the map that will be visited during the action $\mathbf{a}$. This leads to an approximate form of the utility function:

$$\mathbf{a}^* =$$
$$\arg\max_{\mathbf{a}} \sum_{m \in \mathbf{m}(\mathbf{a})} \underbrace{\mathbb{H}[P(m \mid \mathbf{x}, \mathbf{d})]}_{\text{Shannon's entropy}} - \underbrace{\mathbb{H}_{\alpha(\mathbf{a})}[P(m \mid \mathbf{x}, \mathbf{d})]}_{\text{Rényi's entropy with}}$$
$$\tag{14}$$

where $\mathbf{m}(\mathbf{a})$ is the subset of the *current* map that may be visible to the robot while executing action $\mathbf{a}$. $\mathbf{m}(\mathbf{a})$ may be computed using standard ray tracing techniques. The key difference between our utility function and (10) is the Rényi entropy.

One advantage of our proposed approach is that both entropy terms in the right hand side of (12) are of the same type, i.e., they can be added and subtracted in a meaningful way, unlike the entropy terms in (7) and (8). Both terms in (12) stem from Rényi's general theory of entropy (Principe 2010; Rényi 1960) and both aim to quantify the uncertainty of the same underlying probability distribution using different "norms" in probability space. Just like Rényi's entropy generalizes the concept of Shannon's entropy, our proposed objective function generalizes previous objective functions, such as (10).

Note that the proposed utility function makes no assumption about the SLAM back end and can be used with any system that provides a distribution over maps and the robot pose. The solitary restriction on (14) is that it is defined only for discrete random variables and thus requires the map to be represented by a finite number of discrete random variables, e.g., an occupancy grid.

3.2 Key properties of the utility function

Rényi's entropy has a key mathematical property:

$$\mathbb{H}[P(x)] > \mathbb{H}_\alpha[P(x)] \geq \mathbb{H}_{\alpha'}[P(x)] \geq 0, \quad 1 < \alpha \leq \alpha'. \tag{15}$$

A proof for this can be found in "Appendix A". Using this property, we see that when $\alpha \geq 1$ then the proposed utility function (13):

1. is non-negative.
2. is bounded from above by Shannon's entropy and from below by zero.
3. monotonically increases with α.

Note that the first property complies with the Shannonian belief that, in expectation, information is never unhelpful, i.e., the value of information is always non-negative. Thus, while the parameter $\alpha \in [0, 1) \cup (1, \infty)$ is a free parameter, we restrict our attention to the range $(1, \infty)$ so that the objective function is non-negative.

In an autonomous exploration task we want the expected information gain to decrease when the robot has either a high level of certainty in the map or a high level of uncertainty in its pose. 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 inconsistent map estimates. However, the map uncertainty computed using (3) will likely decrease because the robot will continue to visit unexplored cells. Intuitively, the Shannon entropy term in (14) represents an optimistic measure of the map uncertainty assuming that the robot will not experience localization errors when carrying out action **a**. Thus, the Rényi entropy term will represent a measure of map uncertainty that takes into account the uncertainty in the pose of the robot.

Figure 2 shows the value of (14) for a single Bernoulli random variable m representing the probability of occupancy of a single cell.

In the case that $P(m) \approx 0$ (or $P(m) \approx 1$), meaning we have (nearly) perfect information about the cell, then a single action will have little effect on the estimate, and the 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.
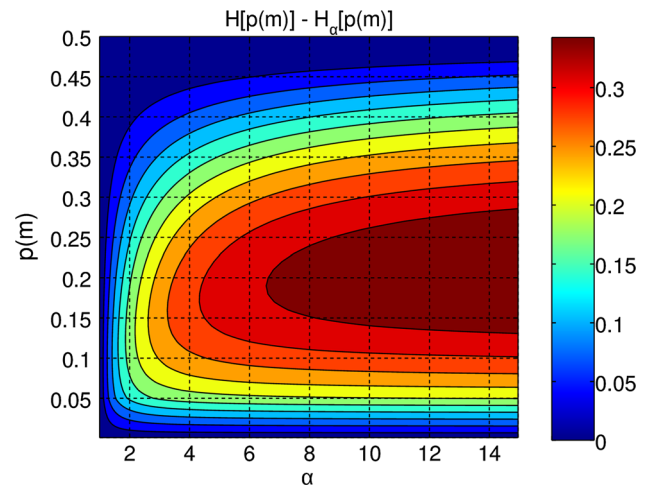

Fig. 2 Value of the utility function for a Bernoulli random variable m. The figure is mirrored for $P(m) \in [0.5, 1]$

When we have a little bit of information about a cell, i.e., $0.05 \lesssim P(m) \lesssim 0.45$ (or $0.55 \lesssim P(m) \lesssim 0.95$), then the robot uncertainty is most important, as an incorrect measurement due to poor localization of the robot 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. 2. We feel that this is a very useful property of the objective function since the robot needs good measurements in order to have full confidence about its pose and the surrounding environment.

Overall the objective function will select actions that will most improve the estimate of the partially explored sections of the map, i.e., where $P(m) \neq \{0, 0.5, 1\}$. Such situations arise when the robot has poor localization during its first pass through an area or when the robot only sees an area in passing, e.g., if the robot gets a brief glimpse into a room while driving past the doorway. These cases will often occur near frontiers in the map.

3.3 The parameter α

We relate the parameter $\alpha(\mathbf{a})$ in (14) to the predicted uncertainty in the pose of the robot after taking action **a** in a way that will decrease the information gain when the pose uncertainty is high. We do this in a way that, while based on heuristic decisions, is independent of the environment, sensor, and robot; avoids the need for any manual tuning; and is based on the mathematical properties of Rényi's entropy and the proposed objective function discussed above.

First, let us briefly discuss the proposed objective function (14). The utility value will increase when an action causes the robot's sensor footprint to cover a larger number of uncertain cells ($P(m) \neq \{0, 0.5, 1\}$). This effect naturally causes the

robot to explore. Conversely, the utility value must decrease when an action causes the robot's localization uncertainty to increase, i.e., when the robot cannot exploit existing information. This allows the robot to autonomously switch between exploration and exploitation, a necessity in any autonomous exploration task. The remainder of this section suggests one method for doing this and details the principled reasoning behind our choice.

When the robot has perfect localization, i.e., minimal uncertainty, then the information gain should be maximal. Similarly, when the robot is completely lost, i.e., maximal uncertainty, the the information gain should be zero. In other words, we want $\alpha \to 1$ as the uncertainty becomes infinite since the two entropies in (14) cancel out, and $\alpha \to \infty$ as the uncertainty approaches zero, since this minimizes Rényi's entropy. We also want α to monotonically decrease with increasing pose uncertainty so that (14) will monotonically increase with improved localization of the robot.

A simple candidate relationship that meets all of these requirements is:

$$\alpha = 1 + \frac{1}{\sigma} \qquad (16)$$

where σ is a scalar related to the predicted uncertainty of the robot's pose while taking action $\mathbf{a}$. More complex relationships will be the focus of further study, but we will show in Sect. 5 that (16) is sufficient to demonstrate the efficacy of our proposed utility function in several different environments and in both simulated and hardware experiments without needing to be adjusted in any way. The computation of σ is left until Sect. 4 as it has an indirect connection to the proposed utility function.

4 Utility unction II: autonomous exploration framework

In this section we describe our test framework for robot-based autonomous exploration using the utility function from (14). We also provide a detailed description of several methods to compute the uncertainty scalar σ that was defined in (16), and which we will test out in Sects. 5 and 6.

We require a framework for robot-based autonomous exploration in order to test the proposed utility function. The goal of the test framework is to create an accurate map of an unknown environment in a finite time, which implicitly requires the robot to maintain a good estimate of its pose.

Our test framework is composed of well-known, open-source methods that work in practical scenarios and that, when combined, allow us to experimentally validate our approach, as we will show in Sect. 6. We assume that the robot has a SLAM system running, simultaneously esti-

mating the robot's pose and generating an occupancy grid representation of the environment from the collected sensor data. Our SLAM front-end is an Iterative Closest Point (ICP)-based laser scan matcher (Pomerleau et al. 2013). Our SLAM back-end is the Incremental Smoothing and Mapping (iSAM) library (Kaess et al. 2008), 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) (Tipaldi and Arras 2010). We compare the FLIRT feature sets from different nodes using a modified RANSAC algorithm (Tipaldi and Arras 2010), adding loop closure constraints if the match is sufficiently strong.

Our test framework is divided into three high-level steps, as outlined in Sect. 1: identification of the candidate actions, computation of the utility of each action, and execution of the selected action. We describe our approach to each of these tasks in the remainder of this section.

4.1 Identification of candidate actions

In order to generate candidate goal locations, the robot first identifies frontiers (Yamauchi 1998) in the latest occupancy grid map. A cell of the occupancy grid is labeled as belonging to a frontier if it is unoccupied, adjacent to an unknown cell, and not adjacent to an occupied cell. The robot then clusters these cells to generate frontier regions, with the goal locations being the mean position of the cells within each cluster. This yields a discrete, finite set of candidate actions, and we cannot guarantee that we select the global optimal exploration action (Indelman et al. 2015). However, using a finite set is computationally tractable and allows for real-time exploration.

The robot next creates an action plan $\mathbf{a}$ for each goal location. An action plan is a set of waypoints, in free space, that lead the robot from its current location to a goal: $\mathbf{a} = \{(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)\}$. We use the AD^* algorithm from the SBPL library (Likhachev 2015) to create the action plans using the current occupancy grid.

4.2 Evaluation of action utilities

The robot then computes the utility of each candidate action according to (14). We use standard ray-casting techniques, such as the one described in Stachniss et al. (2005), to determine the set of cells, $\mathbf{m}(\mathbf{a})$, that will be visible to the robot while executing action $\mathbf{a}$. The free parameter of our utility function is α, which depends on the predicted future uncertainty of an action plan via (16). This raises two questions: 1. how to get a good approximation of the robot's localiza-

tion uncertainty during an action **a**, and 2. how to extract a meaningful uncertainty scalar σ.

4.2.1 Uncertainty prediction

Localization uncertainty propagation has been the focus of a great deal of scholarly work. Early work from Smith et al. (1990) provides a method for estimating the uncertainty associated with rigid body transformations. Lu and Milios (1997) extend this work for pose estimation using graph-based representations of trajectories and landmarks. More recently, Censi (2007) provided a computationally tractable method to estimate the covariance of the ICP method, which we employ as our SLAM front-end. Also, Kaess and Dellaert (2009) provide a method for covariance extraction in the iSAM back end that we utilize. However, none of these methods account for the possibility of loop closures that may occur while executing an action plan. A loop closure event, where the robot re-enters a previously explored region of the environment, allows the robot to significantly reduce its localization uncertainty. However, as Cadena (2016, Sect. VIII.B.1) note, there are no efficient and principled methods that exist for predicting future uncertainty with uncertain loop closure events.

We propose a solution to this problem that utilizes iSAM with a series of heuristic rules to predict the robot's future localization uncertainty while executing an action plan. To do this we create a miniature pose graph representing the action plan, which we call the *action pose graph*. We place an initial factor at the robot's current estimated location, with the covariance matrix taken from the most recent node in the graph, as computed by Kaess and Dellaert (2009). We interpolate the action plan **a** with some fixed step size and add pose nodes along the length of the path, adding odometry constraints between them. These constraints have a fixed covariance value associated with them so that longer actions will lead to larger increases in uncertainty. Note that this covariance value is independent of the future measurements, the map, the robot's current pose, and the candidate action. Future work will aim to correctly account for the dependence of the odometry covariance on these factors, without significantly increasing the computational time, and to examine their effects on the overall performance of the system.

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 of the environment 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. Future work will aim to develop
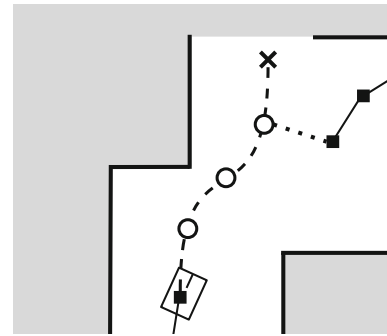

Fig. 3 Candidate path through the environment, denoted by the *dashed line*. The occupancy grid map is shown in the background, with *white* representing free space, *black* is occupied space, and *grey* is unknown. The robot is at the *bottom* and the *dashed line* is a path through free space to the 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

a data-driven model for the probability of a loop closure that uses a measure of the local structure of the environment (e.g., the number of FLIRT features).

We use the iSAM library to optimize the action pose graph for the action plan **a**. These action pose graphs typically consist of only 10's of nodes, so the computational load is minimal. Figure 3 illustrates this process.

Note that this method requires the use of a graph-based SLAM back end. Future work will aim to expand this method to work with Kalman filter-based and particle filter-based SLAM back ends.

4.2.2 Uncertainty scalar

We compute the uncertainty scalar σ from (16) using the covariance matrix estimates from the action pose graph. From the Theory of Optimal Experimental Design (TOED) (Carrillo et al. 2012a, 2015b; Pukelsheim 2006), there are several standard optimality criteria that map a covariance matrix to a scalar value while retaining useful statistical properties. The three most commonly used criteria are:

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

$$\frac{1}{n}\text{trace}(\mathbf{\Sigma}) = \frac{1}{n}\sum_{k=1}^{n}\lambda_k \tag{17}$$

where n is the dimension of the covariance matrix $\mathbf{\Sigma}$ and λ_k is its kth eigenvalue.
– D-optimality (*D-opt*), which minimizes the volume of the covariance matrix ellipsoid,
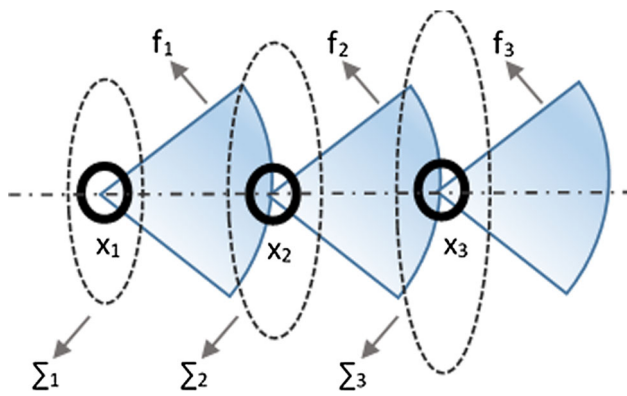
Fig. 4 The figure depicts an action plan with three steps. The mean estimated pose ($\mathbf{x}_k$) of the robot, the footprint of a laser sensor attached to the robot (f_k), and the covariance matrix ellipsoid of the robot's pose ($\mathbf{\Sigma}_k$) are shown at each step

$$\det(\mathbf{\Sigma})^{1/n} = \exp\left(\frac{1}{n}\sum_{k=1}^{n}\log(\lambda_k)\right). \qquad (18)$$

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

$$\max(\lambda_k). \qquad (19)$$

These criteria can be applied to either the full covariance matrix or to the marginal covariance matrices from each node in the action pose graph. If the full covariance matrix is used, there is a single α value along the entire path. Then (14) is applied to the full subset of the map visited by the robot when executing action $\mathbf{a}$. Using the 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 (14) in that cell. In this way, the information gain is calculated using the uncertainty of the robot when it last viewed each individual cell. Figure 4 illustrates this process.

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.*

4.3 Execution of actions

We assume that the robot has a navigation system capable of driving it to any accessible point in the environment. For our experiments we use the algorithm proposed in Guzzi et al. (2013), 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 current goal. 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 passes a sequence of intermediate waypoints to the low-level controller from Guzzi et al. (2013).

It is possible to experience a failure in the navigation or SLAM system while executing an action. If this occurs, 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. If the failure occurs within a short time of the beginning of the action $\mathbf{a}$, the next best action $\mathbf{a}'$ is executed, without recomputing the action set.

5 Simulated experiments

We perform a series of experiments in the four simulated environments from Fig. 5, all of which are available in the Radish repository (Howard and Roy 2009). The code for the experiments was written in C++ using ROS and run on a computer with an Intel i7 processor, 8 GB of RAM, and running 64-bit Ubuntu 12.04.

The objective of the experiments is to compare the performance of our proposed utility function with a state-of-the-art utility function (4) based only on Shannon's entropy. Let SH denote the standard utility function based on Shannon entropy. For our proposed utility function, we test two
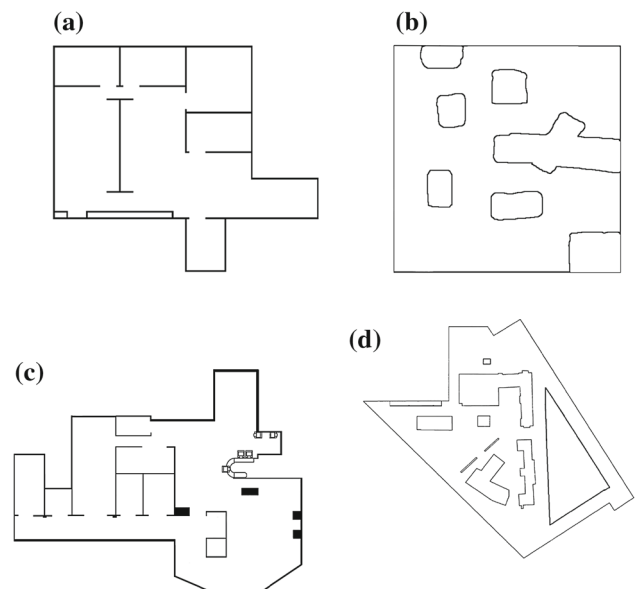

Fig. 5 Maps used in the simulation experiments. **a** Autolab. **b** Cave. **c** Cumberland. **d** Freiburg

optimality criteria for the σ computation: *A-opt* (`A`) and *D-opt* (`D`). Both use the full covariance matrix over the action pose graph.

5.1 Experimental procedure

In each trial the robot starts at one of five different locations in each environment. The robot has a 3 min time budget to complete the exploration task. It is worth remarking that with more time the robot would explore the whole environment, but it is more interesting to compare the different utility functions under the same conditions. Figure 8 shows an example of a snapshot of the resulting occupancy grids for a complete run of a simulated experiment without time constraints.

The robot is a differential drive platform equipped with a planar laser scanner. The maximum velocity of the robot is 0.4 m/s, the maximum range of the laser is 10 m, the field of view of the laser is 270°, and there are 1080 beams in each scan.

We use our own kinematic and laser simulators based on open-source initiatives (Charrow and Dames 2016; MRPT 2016) and the work of Fernandez (2012, Ch. 5) whose models include noise. The main reason behind using our own simulators is computational complexity. Open-source and ROS compatible mobile robotic simulators such as Gazebo, MORSE, or V-REP are designed to create visually appealing, dynamic simulations and they tend to be processor, memory, and GPU hungry. For our simulations we did not need elaborate 3D graphics or robot/environment dynamics so we used our own kinematic simulators.

The kinematic simulator keeps track of the robot's ground-truth position in the global frame and the noise-corrupted pose in the odometry frame, the latter of which is used as the input to laser odometry. The commanded linear and angular velocities are integrated at a fixed rate of 50 Hz in the simulator, and Gaussian noise is added to the estimated pose, with standard deviations of 5 cm for every meter in the (x, y) position and 1° for every 45° in the orientation. The laser simulator uses the ground-truth pose in the global frame and a ground-truth map to compute the scan, adding Gaussian noise with a standard deviation of 1 cm to each beam. These noise values are comparable to those seen on the actual platform described in Sect. 6.

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 the standard definition given by the theory of optimal experiment design (Pukelsheim 2006), using D-optimality (i.e., the determinant of the covariance matrix) as suggested in Carrillo et al. (2012a). The percentage of area correctly explored

is measured using the balanced accuracy, which we introduce in Sect. 5.3.2.

5.2 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 to a similar problem from Pomerleau et al. (2013), 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. It also avoids misleading anecdotal evidence due to noise or outliers.

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 Shimazaki and Shinomoto (2007), summarized as follows:

1. Split the data range into N bins of width Δ, and count the number of events k in each bin i.
2. Compute the mean and variance of the events as follows:

$$\bar{k} = \frac{1}{N} \sum_{i=1}^{N} k_i \tag{20}$$

$$v = \frac{1}{N} \sum_{i=1}^{N} (k_i - \bar{k})^2 \tag{21}$$

3. Find the Δ that minimizes:

$$C(\Delta) = \frac{2\bar{k} - v}{\Delta^2} \tag{22}$$

Note that $\bar{k}$ and v both depend on the choice Δ so computing the value of Δ that minimizes $C(\Delta)$ is non-trivial. Shimazaki and Shinomoto (2007) provide a detailed set of procedures to efficiently compute this Δ.

To more clearly quantify the differences between CDFs, we extract three point estimates: the 50th, 75th and 95th 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.

5.3 Simulation results

5.3.1 CDFs of uncertainty

Figures 6 and 7 show the CDFs of the position uncertainty for all the test environments. Figure 6 shows the uncertainty in
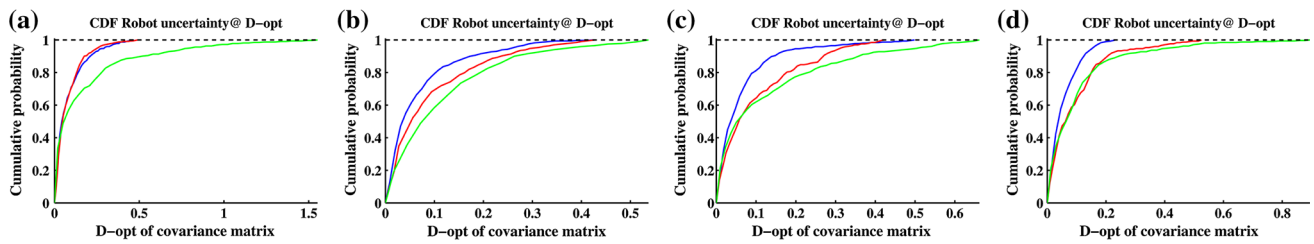
Fig. 6 CDF of the uncertainty in the robot pose at every time step of every simulation. Results using the SH strategy are *green*, the A strategy are *blue*, and the D strategy are *red*. **a** Autolab. **b** Cave. **c** Cumberland. **d** Freiburg (Color figure online)

Fig. 7 CDF of the running average uncertainty of the robot pose over each simulation. Results using the SH strategy are *green*, the A strategy are *blue*, and the D strategy are *red*. **a** Autolab. **b** Cave. **c** Cumberland. **d** Freiburg (Color figure online)
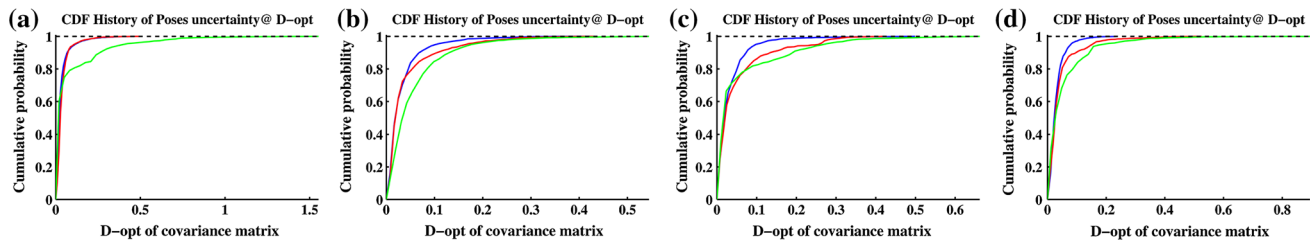
Table 2 50th, 75th and 95th percentile of the CDFs in Figs. 6 and 7

	Autolab			Cave			Cumberland			Freiburg		
	P50	P75	P95	P50	P75	P95	P50	P75	P95	P50	P75	P95
Robot's pose												
A	0.048	0.121	0.302	0.040	0.092	0.262	0.040	0.082	0.222	0.045	0.088	0.159
D	0.051	0.121	0.257	0.057	0.139	0.308	0.061	0.158	0.315	0.067	0.144	0.332
SH	0.072	0.252	0.808	0.087	0.175	0.380	0.057	0.187	0.515	0.072	0.136	0.426
Pose history												
A	0.023	0.039	0.116	0.022	0.043	0.106	0.022	0.047	0.101	0.025	0.040	0.081
D	0.031	0.048	0.111	0.022	0.043	0.169	0.024	0.060	0.258	0.030	0.049	0.155
SH	0.026	0.066	0.411	0.039	0.077	0.181	0.022	0.057	0.274	0.035	0.073	0.186

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. Figure 7 shows the running average uncertainty over the history of robot's poses, measuring the average performance over a trial.

Table 2 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 75th percentile than the robot using Shannon entropy. This difference is still large (46.84%) at the 95th percentile.

5.3.2 Percentage of explored area

Evaluating the percentage of a map that has been correctly labeled as free or occupied by a robot during exploration is inherently a classification problem. However, the number of free cells is typically much greater than the number of occupied cells, and knowing the correct locations of the obstacles

is typically more important than the free space. A robot may return a large number of matched free cells even if there are very few matches of the occupied cells, indicating that the obstacles are poorly localized.

To avoid this bias towards free space, we measure the map accuracy by independently counting the number of free and occupied cells that are correctly labeled. The percentage of the map that has been correctly explored is then computed using the balanced accuracy (BAC), a concept borrowed from the machine learning community (Brodersen et al. 2010a). The BAC equally weights the accuracy of the estimates of the free and occupied cells:

$$\text{BAC} = \frac{1}{2} \left(\frac{\text{\# correct free cells}}{\text{\# total free cells}} + \frac{\text{\# correct occupied cells}}{\text{\# total occupied cells}} \right) \tag{23}$$

Table 3 shows the percentage of each map correctly explored by a robot using each utility function. The pro-

Table 3 Mean and standard deviation (σ) of the 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
	Cumberland					Freiburg				
	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	32.11	7.03	6.89	2.43	19.50	44.90	6.79	14.47	3.89	29.68
D	31.65	5.98	8.51	2.21	20.08	42.60	5.01	18.68	3.30	30.64
SH	30.27	6.83	3.72	1.84	16.99	43.31	6.38	8.83	2.36	26.07
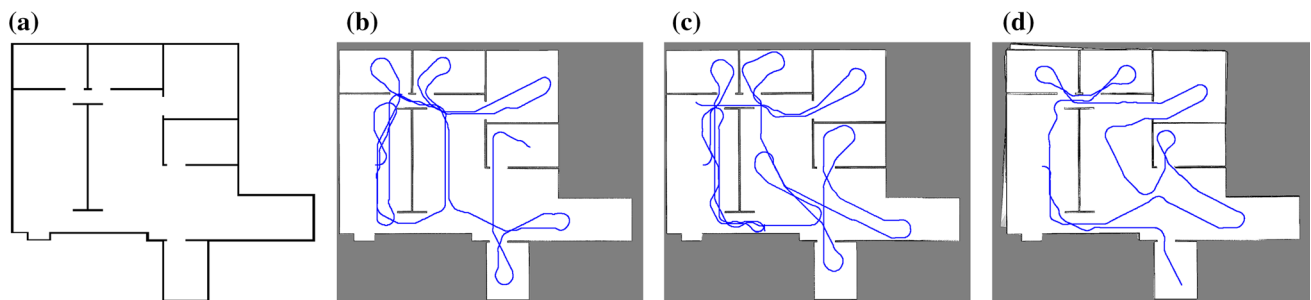
Fig. 8 Examples of complete exploration in the Autolab environment. The visualization is done in MATLAB from the data gathered in ROS. **a** Ground truth of the Autolab environment. **b** Resulting occupancy grid and estimated trajectory in *blue* using A. **c** using D. **d** using SH (Color figure online)
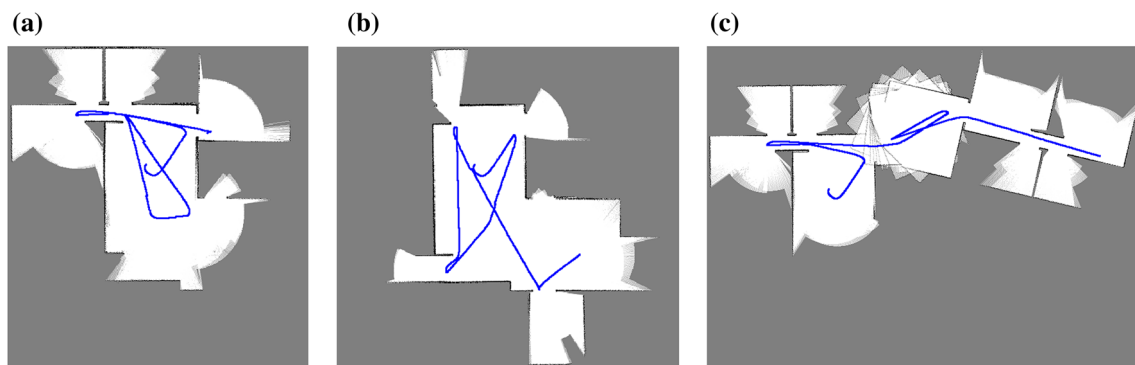
Fig. 9 Examples of 3 min exploration in the Autolab environment, starting from a position in the center. The resulting occupancy grid and estimated trajectory in *blue* using **a** A, **b** D, and **c** SH (Color figure online)

posed utility function, while more conservative in its choice of actions, still explores a similar fraction of the free space compared to the standard approach (SH) while correctly identifying a significantly higher fraction of the occupied cells.

5.4 Exploration examples

Figure 8 shows an example of an exhaustive exploration of the Autolab environment. Although the topology of the envi-ronment constrains the possible trajectories of the robot, it can be seen in that robots using the strategies A and D tend to re-traverse known terrain more frequently. On the other hand, robots using SH tend to venture into new terrain more quickly. In situations with high noise in the odometry and in the sensors this sometimes leads to an incorrect loop closure, which yields an inaccurate map. In the Autolab examples, it can be seen that the robot using the SH strategy has notice-able artifacts in the top-left and bottom-left corners, such as
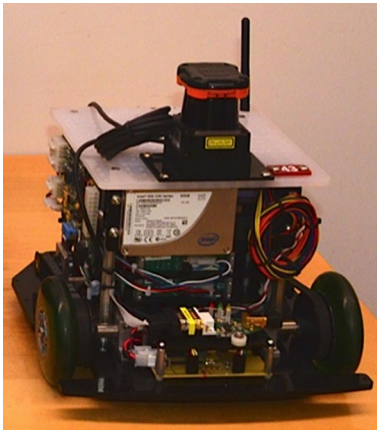
Fig. 10 Robotic platform used to perform the autonomous exploration task

showing the same wall twice and showing free space outside of a wall. The result from robots using the other two strategies are better, but they still contain minor artifacts in the walls in the upper and bottom part with A and in the upper part with D.

Figure 9 shows examples of maps created by robots at the 3 min mark in the Autolab test environment. Note that the estimate of the robot using SH has diverged and the map is inconsistent, as Fig. 9 shows. The estimates of robots using either A or D never diverged in our trials. This is likely due to the fact that the proposed strategy is more conservative

than the SH strategy due to the novel way that we account for localization uncertainty during exploration.

6 Hardware experiments

As Brooks et al. (1993) famously quoted, "simulations are doomed to succeed." To validate the real-world efficacy of our proposed utility function we also performed a series of experiments using the robot platform in Fig. 10 to explore the office environment at the University of Pennsylvania shown in Fig. 11. The hallways are not equipped with a motion capture system, or any other method of obtaining ground truth localization data. Because of this we cannot provide an insightful comparison of the proposed strategy against the SH strategy. Nevertheless, the hardware experiments will allow us to compare different parameterizations of our framework and check how they behave with real data in a new environment.

Instead of the SH strategy, we teleoperated the robot as a benchmark. The teleoperated robot was running the same SLAM system, the only difference was that the control input was coming from a human instead of the autonomous navigation system (which gets actions from the utility function).
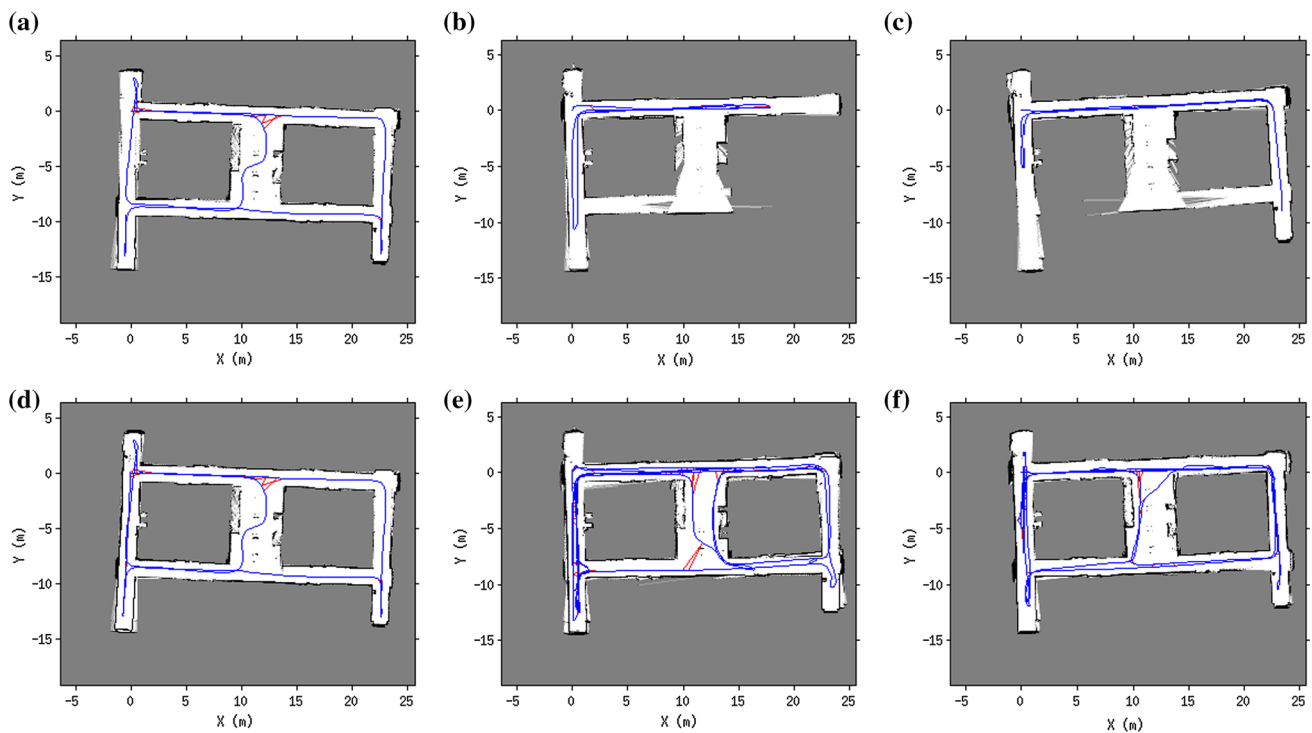

Fig. 11 Examples of the resulting maps and pose graphs from hardware experiments. The robot is either teleoperated, or autonomously explores using the A and D methods from Sect. 5. The *blue* edges indicate odometry constraints in the pose graph while *red* edges indicate loop closure events. **a–c)** The occupancy grids and pose graphs built by the robot after 5 min. **d–f** The final occupancy grids and pose graphs. **a** Teleoperated—5 min. **b** A—5 min. **c** D—5 min. **d** Teleoperated—5 min 21 s. **e** A—30 min 57 s. **f** D—16 min 41 s (Color figure online)

The differential drive robot has a maximum velocity of 0.45 m/s and is equipped with a Hokuyo UTM-30LX laser scanner (30 m range and 270° field of view with 1080 beams per scan) that is uses both for laser-based odometry (Olson 2009) and map building. The robot has an on-board computer with an Intel i5 processor and 8 GB of RAM running Ubuntu 14.04 and ROS Indigo. All of the computations, including sensor processing, SLAM, and utility computations, are performed directly on the robot. Michael et al. (2008) provide further details on the robot.

6.1 Experimental procedure

The robot begins each trial at one of five different starting locations within the environment with no prior information about the map. Unlike the simulated trials, where the robot had a fixed time budget, in the hardware trials the robot explores until it no longer has any available actions. Note that this stopping criterion is sensitive to artifacts in the map as phantom walls and blurred obstacles (caused by poor localization) may create false frontiers in the map.

We recorded all of the data from each trial locally on the robot. We then used these files to generate the resulting maps on a separate computer after the experimental trials were completed.

6.2 Results

We compare the results of a robot using the A and D methods, as described in Sect. 4, to a teleoperated exploration. The test environment is a typical office environment, with long and short corridors that intersect. Figure 11 depicts an occupancy grid representation of the environment, both after 5 min and when the robot has completed the exploration task. Figure 11 presents the results of two exploration runs, with videos of some experiments in the accompanying multimedia material.

Figure 12 quantifies the exploration rates that Fig. 11 qualitatively shows. Specifically, Fig. 12a shows the number of cells explored over time. A cell m is labeled as explored if, in the occupancy grid map, the probability of occupancy $P(m) \neq 0.5$. As can be seen, the robots quickly explore a large fraction of the map and spend the remaining time visiting the rest of the environment and re-exploring areas where the map is incomplete or inconsistent. Note that while in reality the explored area monotonically increases, the number of explored cells does not. The occasional drops in the number of explored cells are due to loop closure events causing the robot's past locations, and thus the resulting estimate of the map, to shift.

Overall, the hardware experiments show the conservative behavior of our utility functions, with the robot re-traversing known areas of the map in order to maintain good localization and correct artifacts in the map. The experiments with *A-opt*
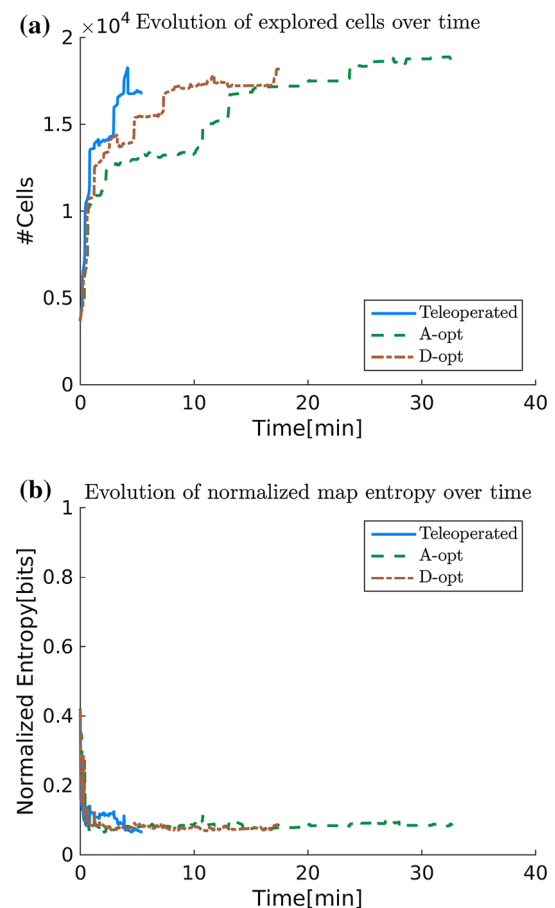

Fig. 12 **a** Evolution of the explored cells over time for the hardware experiments. **b** Evolution of map normalize entropy over time for the hardware experiments. Each strategy quickly plateaus to a normalized entropy less than 0.1, which corresponds to a probability of occupancy $P(m) < 0.02$ (or $P(m) > 0.98$)). The teleoperated robot does it faster given the human in the loop, but the robots using the A and D strategies are able to achieve comparable results. **a** Cells explored versus time. **b** Entropy versus time

show that it is more conservative than *D-opt*, which agrees with the findings of Carrillo et al. (2012a, 2013, 2012b). The experiments also reveal the fact that the utility function is not robust to failure in the SLAM or navigation system. In other 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.

Figure 12b shows the normalized entropy of the explored cells. This is computed by first identifying the portion of the map that has been explored, computing the entropy of those cells, and dividing by the total number of cells. We see that all three hardware experiments have a similar, and low, average entropy per explored cell. Each strategy quickly plateaus to a normalized entropy less than 0.1, which corresponds to a probability of occupancy $P(m) < 0.02$ (or $P(m) > 0.98$)).

Robots using strategies A and D are slower than the teleoperated robot, by a factor of 5.79 and 3.14, respectively. However, in the case of the teleoperated robot the human user had prior knowledge of the environment topology and thus was able to move in near optimal direction at any given time. Also, the user did not account for any uncertainty in the pose of the robot since the human knew where they were in the environment at all times. Lastly, the human user stopped driving the robot after the map looked topologically correct, even if there were small artifacts in the map that would have caused the autonomous robot to continue exploring. Despite these advantages for the teleoperated robot, the robots using our proposed utility function were able to autonomously create maps that are qualitatively and quantitatively (in terms of area and uncertainty) similar to that of the teleoperated robot.

7 Conclusions

In this paper we presented a novel information-theoretic utility function to select actions for a robot in an autonomous exploration task. To effectively explore an unknown environment a robot must be capable of automatically balancing the conflicting goals of exploring new area and exploiting prior information. The robot must do this by considering the uncertainties in both the robot's pose and the map as well as predicting the effects of future actions. Most existing utility functions use a convex combination of the map and the robot's pose uncertainty. This requires the user to manually tune the weighting parameters, which depend on the robot, sensor, and environment.

Instead we utilize a novel definition of mutual information from Jumarie (1990), which computes the difference between the Shannon and Rényi entropies of the map. We tie the α parameter in Rényi's entropy to the predicted uncertainty in the robot's pose after taking a candidate action in a heuristic, but principled, manner. This gives our utility function several key properties: it is non-negative, it is bounded from above by the Shannon entropy (i.e., the existing approaches), and it 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.

Our approach reduces to the standard approaches that use only Shannon's entropy when the localization uncertainty is eliminated. Carlone et al. (2010, 2014) also discount the information gain of each action based on the probability of having good localization of the robot. At a high level, Carlone's proposal and the utility function presented in this paper are similar. However, our work uses a different mathematical approach and can be adapted for use in different SLAM paradigms while Carlone's proposal is tailored for particle filter SLAM.

A side contribution of our work is a new method for predicting the future localization uncertainty of a robot. As Cadena et al. (2016) note, no existing methods properly account for the effects of uncertain loop closure events in a tractable manner. Our proposed solution creates an action pose graph for each candidate action. This pose graph consists of uniformly spaced points along a planned trajectory that are connected with odometry constraints. The nominal values come from the planned trajectory and each constraint is given a fixed covariance value, independent of the environment, robot, and action. We then add additional nodes from the full pose graph, and edges connecting these nodes to the action pose graph, if the action bring the robot near a previously explored region with sufficient local structure to close a loop, as measured by the number of FLIRT features. This method creates a small pose graph with 10's of nodes for each action and can be optimized using the iSAM library in a fraction of a second, allowing us to compute σ values for candidate actions in real time. The only drawback is that it restricts the SLAM back end to be graph-based, which is minor as these are the current state of the art.

Another side contribution of the experimental sections is the evaluation of autonomous exploration tasks using the CDF rather than the mean or median value of the trials. The use of CDFs provides a richer representation of the variability of the results than the mean or median, while avoiding misleading anecdotal evidence due to noise or chance. Moreover, given the nature of experiments in autonomous exploration, where each trial can be thought as the result of a Monte–Carlo simulation, the use of CDFs are justified as they allow an easier comparison between trials with radically different trajectories.

Finally, the last contribution is the release of the code used in the experimental section. The code will be available at https://bitbucket.org/hcarrillo and on the authors' websites.

A possible avenue for future work with the proposed utility function is the study of different functions for mapping the α parameter of Rényi's entropy to the uncertainty of the robot. In this paper we use the Theory of Optimal Experiment Design to do this mapping. In particular, we map the estimated covariance matrix for an action to a scalar σ and then link this scalar σ to the α parameter using (16). The study of the effect of different monotonic functions, such as an exponential or logarithmic relationship, to map the scalar σ is of future interest. Additionally, different techniques from those in the theory of optimal experiment design can be used to map the uncertainty of the robot to a scalar value, such as kernel estimation techniques from Principe (2010) or Bayesian optimization.

Another direction for future work is to improve upon our method for predicting future localization uncertainty. First, we could create a data-driven model for the likelihood of

closing a loop as a function of the local structure. Second, we could connect the covariance of each odometry constraint to the robot, environment, and action, for example using the work of Censi (2007). We believe that these steps will lead to more accurate estimates of future uncertainty. We also hope to generalize this method to other SLAM back ends.

Our simulations and experimental results showed a substantial reduction of the robot's 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 the current map information, resulting in more loop closures. However, the resulting 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.

Appendix

A.1 Properties of Rényi's Entropy

Entropy is a measure of the uncertainty of a random variable (Shannon and Weaver 1949; Rényi 1960; Jumarie 1990). A proper definition of entropy should comply with a set of axioms that guarantee a coherent way of accounting for uncertainty. A widely accepted set of axioms was developed by Aczél and Daróczy (1975). Feinstein (1958) also developed an earlier and more succinct set.

The first attempt to mathematically define entropy was from Hartley (1928). The second definition was developed by Shannon and Weaver (1949), and is the most widely-known and commonly-used definition. Finally Rényi (1960, 1970) created a family of entropy functions, of which the entropies of Hartley and Shannon are special cases. This family of functions, parameterized by α, is defined as:

$$\mathbb{H}_\alpha[P(\mathbf{x})] = \frac{1}{1-\alpha} \log_2 \left(\sum_{i=1}^{n} p_i^\alpha \right) \tag{A.1}$$

where $p_i = P(\mathbf{x} = \mathbf{x}_i)$ is an element of the probability distribution of a discrete random variable $\mathbf{x}$, so that $p_i \geq 0$, $\forall i$ and $\sum_{i=1}^{n} p_i = 1$. The variable α is a free parameter in the range $[0, 1) \cup (1, \infty)$.

A.2 $\mathbb{H}_\alpha$ at $\alpha = 0$

Plugging $\alpha = 0$ in to (A.1) yields $\mathbb{H}_0[P(\mathbf{x})] = \log_2 n$, which is the Hartley entropy.

A.3 $\mathbb{H}_\alpha$ as $\alpha \to 1$

Note from the definition of Rényi's entropy in (A.1) that it is undefined at $\alpha = 1$. Thus to define $\mathbb{H}_1[P(\mathbf{x})]$ we must look at the limit as $\alpha \to 1$:

$$\mathbb{H}_1[P(\mathbf{x})] = \lim_{\alpha \to 1} \mathbb{H}_\alpha[P(\mathbf{x})]. \tag{A.2}$$

Applying the limit directly, we obtain a canonical indeterminate form $\frac{0}{0}$. Applying l'Hôpital's rule we see that:

$$\begin{aligned} \mathbb{H}_1[P(\mathbf{x})] &= \lim_{\alpha \to 1} \frac{\left(\sum_{i=1}^{n} p_i^\alpha\right)^{-1} \left(\sum_{i=1}^{n} p_i^\alpha \log_2(p_i)\right)}{-1} \\ &= -\sum_{i=1}^{n} p_i \log_2(p_i) \\ &= \mathbb{H}[P(\mathbf{x})]. \end{aligned} \tag{A.3}$$

In other words, in the limit as $\alpha \to 1$ Rényi's entropy becomes equal to Shannon's entropy.

A.4 $\mathbb{H}_\alpha$ as $\alpha \to \infty$

Attempting to compute the limit at infinity of $\mathbb{H}_\alpha[P(\mathbf{x})]$ directly yields infinity as a result. However, we can obtain the true value using the squeeze theorem. We start by defining $p_{i'} = \max(p_i)$ and recall that $i \in \{1 \ldots n\}$, $0 \leq p_i \leq 1$, and $\sum_{i=1}^{n} p_i = 1$. Hence, for $1 < \alpha < \infty$, the following inequality stands:

$$p_{i'}^\alpha \leq \sum_{i=1}^{n} p_i^\alpha \leq n \, p_{i'}^\alpha \tag{A.4}$$

If we take the binary logarithm of (A.4), divide by $1 - \alpha$, and rearrange terms, we obtain a more familiar inequality:

$$\log_2(p_{i'}^\alpha) \leq \log_2 \left(\sum_{i=1}^{n} p_i^\alpha \right) \leq \log_2(n \, p_{i'}^\alpha)$$

$$\alpha \log_2(p_{i'}) \leq \log_2 \left(\sum_{i=1}^{n} p_i^\alpha \right) \leq \log_2(n \, p_{i'}^\alpha)$$

$$\frac{\alpha}{1-\alpha} \log_2(p_{i'}) \geq \mathbb{H}_\alpha[P(\mathbf{x})] \geq \frac{\log_2(n)}{1-\alpha} + \frac{\alpha}{1-\alpha} \log_2(p_{i'}) \tag{A.5}$$

Computing the limit as $\alpha \to \infty$ with l'Hôpital's rule, we see that both sides yield the same value of $-\log_2(p_{i'})$. Hence, according to the squeeze theorem, we can compute the desired limit:

$$\mathbb{H}_\infty[P(\mathbf{x})] = \lim_{\alpha \to \infty} \mathbb{H}_\alpha[P(\mathbf{x})] = -\log_2(\max_i p_i) \tag{A.6}$$

A.5 Monotonicity with respect to α

We seek to show that Rényi's entropy monotonically decreases with increasing α. To do this, we take the derivative with respect to α and show that it is non-positive. Let $q_i = p_i^\alpha / \sum_j p_j^\alpha$, and note that this defines a probability distribution $Q(\mathbf{x})$.

Taking the derivative of (A.1) with respect to α yields:

$$
\begin{aligned}
\frac{d}{d\alpha} \mathbb{H}_\alpha[P(\mathbf{x})] &= \frac{d}{d\alpha} \frac{1}{1-\alpha} \log_2\left(\sum_{i=1}^n p_i^\alpha\right) \\
&= \frac{(1-\alpha)(\sum_j p_j^\alpha)^{-1}(\sum_i p_i^\alpha \log p_i) + \log(\sum_j p_j^\alpha)}{(1-\alpha)^2} \\
&= \frac{(1-\alpha)(\sum_i q_i \log p_i) + \sum_i (q_i \log(\sum_j p_j^\alpha))}{(1-\alpha)^2} \\
&= \frac{\sum_i q_i \log p_i - q_i \log p_i^\alpha + q_i \log(\sum_j p_j^\alpha)}{(1-\alpha)^2} \\
&= \frac{\sum_i q_i \log p_i - q_i \log q_i}{(1-\alpha)^2} \\
&= -\frac{\sum_i q_i \log \frac{q_i}{p_i}}{(1-\alpha)^2} \\
&= -\frac{\mathrm{KL}[Q(\mathbf{x}) \,||\, P(\mathbf{x})]}{(1-\alpha)^2}.
\end{aligned}
\tag{A.7}
$$

Since both $(1-\alpha)^2$ and the Kullback–Leibler divergence $\mathrm{KL}[Q(\mathbf{x}) \,||\, P(\mathbf{x})]$ are non-negative (Cover and Thomas 2012) the derivative is non-positive. Thus we conclude that Rényi's entropy monotonically decreases in α for $\alpha \in (1, \infty)$.

A.6 Useful inequalities of $\mathbb{H}_\alpha$

Let us consider two values for the free parameter of the Rényi entropy, α and α', such that $1 \le \alpha \le \alpha'$. For these two values of the free parameter, we can show that:

1. $\mathbb{H}[P(\mathbf{x})] \ge \mathbb{H}_\alpha[P(\mathbf{x})], \ \forall \alpha \ge 1$
2. $\mathbb{H}[P(\mathbf{x})] \ge \mathbb{H}_\alpha[P(\mathbf{x})] \ge \mathbb{H}_{\alpha'}[P(\mathbf{x})], \ 1 \le \alpha \le \alpha'$.

A.6.1 $\mathbb{H}[P(\mathbf{x})] \ge \mathbb{H}_\alpha[P(\mathbf{x})]$

The function $-\log_2(z)$ is convex and a non-negative weighted sum operation does not affect the convexity of a function Boyd and Vandenberghe (2004, Ch. 3), hence using the p_i as weights, the function $-\sum_{i=1}^n p_i \log_2(z_i)$ is still convex. Applying Jensen's inequality we see that:

$$
-\log_2\left(\sum_{i=1}^n p_i z_i\right) \le -\sum_{i=1}^n p_i \log_2(z_i)
\tag{A.8}
$$

where it is understood that $\sum_{i=1}^n p_i = 1$. Letting $z_i = p_i^{\alpha-1}$, (A.8) becomes:

$$
-\log_2\left(\sum_{i=1}^n p_i^\alpha\right) \le -(\alpha-1)\sum_{i=1}^n p_i \log_2(p_i)
\tag{A.9}
$$

$$
\frac{1}{1-\alpha} \log_2\left(\sum_{i=1}^n p_i^\alpha\right) \le -\sum_{i=1}^n p_i \log_2(p_i)
\tag{A.10}
$$

$$
\mathbb{H}_\alpha[P(\mathbf{x})] \le \mathbb{H}[P(\mathbf{x})].
\tag{A.11}
$$

A.6.2 $\mathbb{H}[P(\mathbf{x})] \ge \mathbb{H}_\alpha[P(\mathbf{x})] \ge \mathbb{H}_{\alpha'}[P(\mathbf{x})]$

It is known Hardy et al. (1952, Theorem 16) that

$$
\left(\sum_{i=1}^n w_i x_i^\beta\right)^{\frac{1}{\beta}}
\tag{A.12}
$$

is a monotone increasing function of β. Rewriting (A.1) as

$$
\mathbb{H}_\alpha[P(\mathbf{x})] = \log_2\left(\sum_{i=1}^n p_i \left(\frac{1}{p_i}\right)^{1-\alpha}\right)^{\frac{1}{1-\alpha}}
\tag{A.13}
$$

and letting $\beta = 1 - \alpha$, we may apply (A.12) to conclude that (A.1) is a monotone decreasing function of α and the inequality holds.

B Probability simplex

Let $P(\mathbf{x})$ be a probability distribution of a discrete random variable $\mathbf{x}$, where $p_i = P(\mathbf{x} = \mathbf{x}_i)$ is an element of the distribution, $p_i \ge 0, \forall i$, and $\sum_{i=1}^N p_i = 1$. The probability simplex Δ_N is a $N-1$ dimensional manifold in a N-dimensional space where all the possible probabilities distributions of a N multidimensional random variable $\mathbf{x}$ lives (Principe 2010). That is:

$$
\Delta_N = \left\{ p = (p_1, \ldots, p_N) \in \mathcal{R}^N, \, p_i \ge 0, \sum_{i=1}^N p_i = 1, \ \ \forall i \right\}
\tag{A.14}
$$

Where any point p in the probability simplex has a natural interpretation as a discrete probability distribution (Calafiore and Ghaoui 2014).

As an example consider the probability simplex in $\mathcal{R}^3$ for three variables p_1, p_2, p_3 where all possible distributions lives inside the equilateral triangle with vertices at $(1, 0, 0)$, $(0, 1, 0)$ and $(0, 0, 1)$. Figure 13 shows an illustration of the above.

Fig. 13 The probability simplex in $\mathcal{R}^3$ is shown as the *blue triangle* with vertices at $(1, 0, 0)$, $(0, 1, 0)$ and $(0, 0, 1)$. Any point on the simplex represents a probability distribution over three variables p_1, p_2, p_3 (Color figure online)
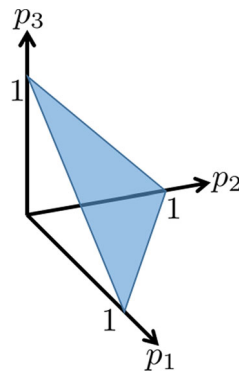
References

Aczél, J., & Daróczy, Z. (1975). On measures of information and their characterizations. In *Mathematics in science and engineering* (Vol. 115). New York, NY Academic Press/Harcourt Brace Jovanovich Publishers.

Averbeck, B. B. (2015). Theory of choice in bandit, information sampling and foraging tasks. *PLoS Computational Biology*, *11*, 1–28. doi:10.1371/journal.pcbi.1004164.

Blanco, J., Fernández-Madrigal, J., & Gonzalez, J. (2008). A novel measure of uncertainty for mobile robot slam with rao-blackwellized particle filters. *The International Journal of Robotics Research (IJRR)*, *27*(1), 73–89. doi:10.1177/0278364907082610.

Bourgault, F., Makarenko, A., Williams, S., Grocholsky, B., & Durrant-Whyte, H. (2002) Information based adaptive robotic exploration. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 540–545). doi:10.1109/IRDS.2002.1041446.

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. New York, NY: Cambridge University Press.

Brodersen, K., Ong, C. S., Stephan, K., & Buhmann, J. (2010). The balanced accuracy and its posterior distribution. In *Proceedings of the international conference on pattern recognition (ICPR)* (pp. 3121–3124). doi:10.1109/ICPR.2010.764.

Brooks, R. A., & Mataric, M. J. (1993). *Real robots, real learning problems*. Berlin: Springer.

Burgard, W., Moors, M., Stachniss, C., & Schneider, F. (2005). Coordinated multi-robot exploration. *IEEE Transactions on Robotics (TRO)*, *21*(3), 376–386. doi:10.1109/TRO.2004.839232.

Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., Neira, J., et al. (2016). Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, *32*(6), 1309–1332.

Calafiore, G., & Ghaoui, L. (2014). *Optimization models*. Cambridge: Cambridge University Press.

Carlone, L., Du, J., Kaouk, M., Bona, B., & Indri, M. (2010) An application of Kullback–Leibler divergence to active slam and exploration with particle filters. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 287–293). doi:10.1109/IROS.2010.5652164.

Carlone, L., Du, J., Kaouk, M., Bona, B., & Indri, M. (2014). Active SLAM and exploration with particle filters using Kullback-Leibler divergence. *Journal of Intelligent & Robotic Systems*, *75*(2), 291–311. doi:10.1007/s10846-013-9981-9.

Carrillo, H., Latif, Y., Neira, J., & Castellanos, J. A. (2012a) 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 (pp. 2504–2511). doi:10.1109/IROS.2012.6385927.

Carrillo, H., Reid, I., & Castellanos, J. A. (2012b) 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 (pp. 2080–2087). doi:10.1109/ICRA.2012.6224890.

Carrillo, H., Birbach, O., Taubig, H., Bauml, B., Frese, U., & Castellanos, J. A. (2013) On task-oriented criteria for configurations selection in robot calibration. In *Proceedings of the IEEE international conference on robotics and automation (ICRA)*, Karlsruhe, Germany (pp. 3653–3659). doi:10.1109/ICRA.2013.6631090.

Carrillo, H., Dames, P., Kumar, K., & Castellanos, J. A. (2015a) Autonomous robotic exploration using occupancy grid maps and graph SLAM based on Shannon and rényi entropy. In *Proceedings of the IEEE international conference on robotics and automation (ICRA)*, Seattle, WA, USA.

Carrillo, H., Latif, Y., Rodríguez, M. L., Neira, J., & Castellanos, J. A. (2015b) On the monotonicity of optimality criteria during exploration in active SLAM. In *Proceedings of the IEEE International conference on robotics and automation (ICRA)*, Seattle, WA, USA.

Censi, A. (2007). An accurate closed-form estimate of ICP's covariance. In *Proceedings 2007 IEEE international conference on robotics and automation* (pp. 3167–3172). IEEE.

Charrow, B., & Dames, P. (2016). ROS code for UPenn's SCARAB robot. https://github.com/bcharrow/scarab.

Cover, T. M., & Thomas, J. A. (2012). *Elements of information theory*. Hoboken, NJ: Wiley.

Dames, P., & Kumar, V. (2013). Cooperative multi-target localization with noisy sensors. In *Proceedings of the IEEE international conference on robotics and automation (ICRA)*, Karlsruhe, Germany.

Du, J., Carlone, L., Kaouk, M., Bona, B., & Indri, M. (2011) A comparative study on active slam and autonomous exploration with particle filters. In *Proceedings of IEEE/ASME international conference on advanced intelligent mechatronics* (pp. 916–923). doi:10.1109/AIM.2011.6027142.

Eustice, R. M., Singh, H., Leonard, J. J., & Walter, M. R. (2006). Visually mapping the RMS titanic: Conservative covariance estimates for SLAM information filters. *The International Journal of Robotics Research (IJRR)*, *25*(12), 1223–1242.

Fairfield, N., & Wettergreen, D. (2010). Active SLAM and Loop prediction with the segmented map using simplified models. In A. Howard, K. Iagnemma, & A. Kelly (Eds.), *Field and service robotics, Springer Tracts in Advanced Robotics* (Vol. 62, pp. 173–182). New York: Springer. doi:10.1007/978-3-642-13408-11_6.

Feinstein, A. (1958). *Foundations of information theory*. New York City, NY: McGraw-Hill.

Fernández-Madrigal, J. A., & Blanco, J. L. (2012). *Simultaneous localization and mapping for mobile robots: Introduction and methods* (1st ed.). Hershey, PA: IGI Global.

Grisetti, G., Kuemmerle, R., Stachniss, C., & Burgard, W. (2010). A tutorial on graph-based SLAM. *IEEE Intelligent Transportation Systems Magazine*, *2*(4), 31–43. doi:10.1109/MITS.2010.939925.

Guzzi, J., Giusti, A., Gambardella, L. M., Theraulaz, G., & Di Caro, G. A. (2013) Human-friendly robot navigation in dynamic environments. In *Proceedings of the IEEE international conference on robotics and automation (ICRA)* (pp. 423–430).

Hardy, G., Littlewood, J., & Pólya, G. (1952). *Inequalities. Cambridge Mathematical Library*. Cambridge: Cambridge University Press.

Hartley, R. V. L. (1928). Transmission of Information. *Bell System Technical Journal*, *7*, 535–563.

Hollinger, G. A., Mitra, U., & Sukhatme, G. S. (2011) Autonomous data collection from underwater sensor networks using acoustic communication. In: *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 3564–3570). IEEE.

Hornung, A., Wurm, K. M., Bennewitz, M., Stachniss, C., & Burgard, W. (2013). OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots (AR)*, *34*(3), 189–206. doi:10.1007/s10514-012-9321-0.

Howard, A., Roy, N. (2009) Radish: The robotics data set repository. http://radish.sourceforge.net/. Accessed October 15, 2014.

Indelman, V., Carlone, L., & Dellaert, F. (2015). Planning in the continuous domain: A generalized belief space approach for autonomous navigation in unknown environments. *The International Journal of Robotics Research*, *34*(7), 849–882. doi:10.1177/0278364914561102.

Jaynes, E. T. (1957). Information theory and statistical mechanics. *Physical Review*, *106*(4), 620.

Jumarie, G. (1990). *Relative information: Theories and applications*. New York, NY: Springer.

Kaess, M., & Dellaert, F. (2009). Covariance recovery from a square root information matrix for data association. *Robotics and Autonomous Systems (RAS)*, *57*, 1198–1210. doi:10.1016/j.robot.2009.06.008.

Kaess, M., Ranganathan, A., & Dellaert, F. (2008). iSAM: Incremental Smoothing and Mapping. *IEEE Transactions on Robotics (TRO)*, *24*(6), 1365–1378.

Kim, A., & Eustice, R. M. (2013) Perception-driven Navigation: Active visual SLAM for robotic area coverage. In *Proceedings of the IEEE international conference on robotics and automation (ICRA)* (pp. 3196–3203). doi:10.1109/ICRA.2013.6631022.

Kim, A., & Eustice, R. M. (2015). Active visual SLAM for robotic area coverage: Theory and experiment. *The International Journal of Robotics Research (IJRR)*, *34*(4–5), 457–475. doi:10.1177/0278364914547893.

Likhachev, M. (2015) Search-based planning library. https://github.com/sbpl/sbpl. Accessed October 15, 2015.

Lu, F., & Milios, E. (1997). Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, *4*(4), 333–349.

Makarenko, A., Williams, S. B., Bourgault, F., & Durrant-Whyte, H. F. (2002) An experiment in integrated exploration. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 534–539). doi:10.1109/IRDS.2002.1041445.

Martinez-Cantin, R., de Freitas, N., Brochu, E., Castellanos, J., & Doucet, A. (2009). A bayesian exploration-exploitation approach for optimal online sensing and planning with a visually guided mobile robot. *Autonomous Robots (AR)*, *27*(2), 93–103. doi:10.1007/s10514-009-9130-2.

Michael, N., Fink, J., & Kumar, V. (2008). Experimental testbed for large multirobot teams. *Robotics and Autonomous Systems (RAS)*, *15*(1), 53–61.

MRPT. (2016). Mobile robot programming toolkit. http://www.mrpt.org/.

Olson, E. (2009) Real-time correlative scan matching. In *Proceedings of the IEEE international conference on robotics and automation (ICRA)* (pp. 1233–1239).

Pomerleau, F., Colas, F., Siegwart, R., & Magnenat, S. (2013). Comparing ICP variants on real-world data sets. *Autonomous Robots (AR)*, *34*(3), 133–148. doi:10.1007/s10514-013-9327-2.

Principe, J. (2010). *Information theoretic learning: Rényi's entropy and kernel perspectives. Information Science and Statistics*. Berlin: Springer.

Pukelsheim, F. (2006). *Optimal design of experiments. Classics in Applied Mathematics*. Philadelphia, PA: Society for Industrial and Applied Mathematics (SIAM).

Rényi, A. (1960) On measures of entropy and information. In: *Proceedings of the 4th Berkeley symposium on mathematics, statistics and probability* (pp. 547–561).

Rényi, A. (1970). *Probability theory. North-Holland series in applied mathematics and mechanics*. Amsterdam: Elsevier.

Roy, N., Burgard, W., Fox, D., Thrun, S. (1999) Coastal navigation—Mobile Robot navigation with uncertainty in dynamic environments. In *Proceedings of the IEEE international conference on robotics and automation (ICRA)*.

Shannon, C., & Weaver, W. (1949). *The mathematical theory of communication*. Champaign, IL: Illinois Books, University of Illinois Press.

Shimazaki, H., & Shinomoto, S. (2007). A method for selecting the bin size of a time histogram. *Neural Computation*, *19*(6), 1503–1527. doi:10.1162/neco.2007.19.6.1503.

Sim, R., Dudek, G., & Roy, N. (2004) Online control policy optimization for minimizing map uncertainty during exploration. In *Proceedings of the IEEE international conference on robotics and automation (ICRA)* (Vol. 2, pp. 1758–1763). doi:10.1109/ROBOT.2004.1308078.

Smith, R., Self, M., & Cheeseman, P. (1990). Estimating uncertain spatial relationships in robotics. In I. J. Cox & G. T. Wilfong (Eds.), *Autonomous robot vehicles* (pp. 167–193). New York, NY: Springer.

Stachniss, C. (2009). *Robotic mapping and exploration* (Vol. 55). Berlin: Springer.

Stachniss, C., Hahnel, D., & Burgard, W. (2004) Exploration with active loop-closing for FastSLAM. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS)* (Vol. 2, pp. 1505–1510). doi:10.1109/IROS.2004.1389609.

Stachniss, C., Grisetti, G., & Burgard, W. (2005) Information gain-based exploration using RAO-Blackwellized particle filters. In *Proceedings of robotics: Science and systems conference (RSS)*, Cambridge, MA, USA.

Thrun, S., Burgard, W., & Fox, D. (2005). *Probabilistic robotics*. Boston, MA: MIT Press.

van den Berg, J., Patil, S., Alterovitz, R., et al. (2012). Motion planning under uncertainty using iterative local optimization in belief space. *The International Journal of Robotics Research (IJRR)*, *31*(11), 1263–1278. doi:10.1177/0278364912456319.

Tipaldi, G. D., & Arras, K. O. (2010) FLIRT-interest regions for 2D range data. In *Proceedings of the IEEE international conference on robotics and automation (ICRA)* (pp. 3616–3622).

Valencia, R., & Valls Miró, J., Dissanayake, G., Andrade-Cetto, J. (2012) Active pose SLAM. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 1885–1891). doi:10.1109/IROS.2012.6385637.

Xu, D. (1998) Energy, entropy and information potential for neural computation. PhD thesis, Gainesville, FL, USA, aAI9935317.

Yamauchi, B. (1998) Frontier-based exploration using multiple robots. In *Proceedings of the second international conference on autonomous agents*, ACM, AGENTS '98 (pp. 47–53). doi:10.1145/280765.280773.

Zhang, Q., Whitney, D., Shkurti, F., & Rekleitis, I. (2014) Ear-based exploration on hybrid metric/topological maps. In *Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS)* (pp. 3081–3088). doi:10.1109/IROS.2014.6942988.

Zhang, Q., Rekleitis, I., & Dudek, G. (2015) Uncertainty reduction via heuristic search planning on hybrid metric/topological map. In *12th conference on computer and robot vision* (pp. 222–229). doi:10.1109/CRV.2015.36.

Henry Carrillo received the B.Eng. degree in Electronics Engineering from Universidad del Norte, Barranquilla, Colombia, in 2007; the M.Sc. degree in Electronics Engineering from Pontificia Universidad Javeriana, Bogotá, Colombia, in 2010; and the Ph.D. degree in Computer Science from Universidad de Zaragoza, Zaragoza, Spain, in 2014. He is an Associate Professor with Escuela de Ciencias Exactas e Ingeniería, Univesidad Sergio Arboleda, Bogotá, Colombia. His research interests include the problems of mapping and localization for autonomous systems.

Philip Dames received his B.S. and M.S. degrees in Mechanical Engineering from Northwestern University in 2010 and his Ph.D. degree in Mechanical Engineering and Applied Mechanics from the University of Pennsylvania in 2015. He is currently an Assistant Professor of Mechanical Engineering at Temple University. His current research interests include the intersection of estimation, control, and communication in multiagent systems.

Vijay Kumar is the Nemirovsky Family Dean of the School of Engineering and Applied Science at the University of Pennsylvania. He received his Ph.D. from The Ohio State University in 1987. He served as the Deputy Dean for Research in the School of Engineering and Applied Science from 2000–2004. He directed the GRASP Laboratory, a multidisciplinary robotics and perception laboratory from 1998–2004. He was the Chairman of the Department of Mechanical Engineering and Applied Mechanics from 2005–2008 and the Deputy Dean for Education in the School of Engineering and

Applied Science from 2008–2012. He was the assistant director for robotics and cyber physical systems at the White House Office of Science and Technology Policy from 2012–2014. He served on the editorial boards of the IEEE Transactions on Robotics and Automation, IEEE Transactions on Automation Science and Engineering, ASME Journal of Mechanical Design, the ASME Journal of Mechanisms and Robotics, and the Springer Tracts in Advanced Robotics (STAR). He is the recipient of the 1991 National Science Foundation Presidential Young Investigator award, the 1997 Freudenstein Award for significant accomplishments in mechanisms and robotics, the 2012 ASME Mechanisms and Robotics Award, the 2012 IEEE Robotics and Automation Society Distinguished Service Award, a 2012 World Technology Network Award, and the 2017 IEEE Robotics and Automation Society George Saridis Leadership Award in Robotics and Automation. He has won best paper awards at DARS 2002, ICRA 2004, ICRA 2011, RSS 2011, and RSS 2013, and his doctoral students won Best Student Paper Awards at ICRA 2008, RSS 2009, and DARS 2010.

José A. Castellanos received the M.S. and Ph.D. degrees in Industrial Electrical Engineering from the University of Zaragoza, Spain, in 1994 and 1998, respectively. He is in charge of courses in SLAM, Control Engineering and DEVS and Continuous Simulation at the Departamento de Informatica e Ingenieria de Sistemas of the University of Zaragoza. He has served as Associate Editor for the IEEE Transactions on Robotics, as Editor for the IEEE/RSJ International Conference on Intelligent Robots and Systems, and currently he serves as Editor for the IEEE International Conference on Robotics and Automation. His current research interests include SLAM, Localization, Mapping, and Autonomous Vehicle Navigation.