Autonomous Exploration by Expected Information Gain from Probabilisitic Occupancy Grid Mapping

Evan Kaufman, Taeyoung Lee, and Zhuming Ai

***Abstract*— Occupancy grid maps are spatial representations of environments, where the space of interest is decomposed into a number of cells that are considered either occupied or free. This paper focuses on exploring occupancy grid maps by predicting the uncertainty of the map. Based on recent improvements in computing occupancy probability, this paper presents a novel approach for selecting robot poses designed to maximize expected map information gain represented by the change in entropy. This result is simplified with several approximations to develop an algorithm suitable for real-time implementation. The predicted information gain proposed in this paper, in conjunction with an existing motion planner to avoid obstacles, yield an effective autonomous exploration strategy, which is illustrated by numerical examples.**

I. Introduction

Robotic mapping is the process of generating maps, which represent the environment in close proximity of a robot. The popular occupancy grid mapping representation provides information about the free and occupied space as the robot measures its surroundings [1]. Occupancy grid mapping is commonly used while estimating robot location and orientation, known as simultaneous localization and mapping (SLAM). Several SLAM approaches are applied to a variety of problems when mobile robots traverse through an environment to generate a map, but the trajectories are assumed given (e.g. [1], [2], [3]). A major problem is that an effective trajectory would be unavailable knowing little or nothing about an environment. Autonomous exploration solves this problem, by choosing robotic motion through uncertain space, while building a map that is used for subsequent navigation [4]. In this paper, robotic motion is chosen to increase knowledge about the map; a robot chooses a trajectory designed to maximize information gain of the occupancy grid map.

A common approach to solve the autonomous exploration problem is known as frontier-based exploration, such as [4], [5]. This is the process of executing actions that move the robot toward the closest boundary between visited and unvisited space, known as a frontier. Then the robot takes measurements at this location such that the mapped territory expands, and thus the new frontiers are pushed back. This process is repeated until the map is well-known. Frontier-based exploration assumes that repeatedly moving toward the closest frontier and taking measurements are the best actions to gain new information about the map, though these systematic actions are not based on any consideration of the future uncertainty of the probabilistic map or optimality.

Approaches aimed at determining the future uncertainty of an occupancy grid map commonly suffer from several approximations. Most evident is that all approaches rely on inaccurate probabilistic occupancy grids. This mapping representation is typically based on heuristic approximations (e.g. [6], [7], [8], [9], [10]) or simulated learning techniques [1], [11]. Map-based information gain is commonly determined with Shannon's entropy [12], an uncertainty metric based on grid cell occupancy probability. Since this probability is heuristic or learned, any measure of entropy is subject to errors from approximated probability. Furthermore, the expected entropy of an action is not directly calculated [12], [13]. Instead, these particle filter-based approaches assume that expected entropy is equivalent to entropy based on expected measurement scans [14]. This assumption is subject to inaccuracies due to the nonlinear nature of Shannon's entropy.

In this paper, we propose an autonomous exploration technique based on expected entropy change that avoids the aforementioned issues. We develop a solution for expected entropy gain based on exact occupancy grid mapping [15]. This solution directly calculates expected entropy from a probabilistic map based on similar assumptions of those applied to occupancy grid mapping. Though more accurate than several other approaches, the proposed solution requires large computational resources, making real-time implementation difficult in certain scenarios. Thus, we provide the tools to approximate the exact solution that reduces the computational requirements with realistic assumptions. The motion of the robot is formulated as an optimization problem, where the expected entropy change is minimized, or equivalently the expected information gain is maximized. Compared with the recent results (e.g. [12], [13], [16], [17]) that employ Rao-Blackwellized particle filters to deal with localization and autonomous exploration simultaneously with an approximate entropy change, this paper is based on an accurate solution to map information gain, under the assumption that the robot pose is known. In short, this paper presents an accurate and computationally-efficient approach to predict map information gain for autonomous exploration with a novel structure that avoids common assumptions of the inverse sensor model and approximations in the entropy

Evan Kaufman, Taeyoung Lee, Mechanical and Aerospace Engineering, George Washington University, Washington DC 20052 {evankaufman,tylee}@gwu.edu

Zhuming Ai, Information Management & Decision Architectures, U.S. Naval Research Laboratory, Washington, DC 20375

This research has been supported by the U.S. Naval Research Laboratory Base Program "Intelligent Microflyer" and in part by NSF under the grants CMMI-1243000, CMMI-1335008, and CNS-1337722.

change.

The paper is organized as follows. The problem is formulated in Section II. The exact solution to the expected uncertainty due to an individual 1D measurement is derived in Section III, along with approximations to the solution. The concept of entropy is extended to 2D autonomous exploration with a numerical example in Section IV, followed by conclusions.

II. Problem Formulation

A. Occupancy Grid Mapping

Let a map m be decomposed into n_m evenly-spaced 2D grid cells, where the i-th grid cell is assigned to a static binary random variable $\mathbf{m}_i$ for $i \in \{1, 2, ..., n_m\}$, that is defined as $\mathbf{m}_i = 1$ when occupied, and $\mathbf{m}_i = 0$ when free. The location and size of each grid cell is assumed known. Therefore, a map m is defined by $\{\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_{n_m}\}$, and there are 2^{n_m} possible maps.

Another random variable is defined as $\bar{\mathbf{m}}_i = 1 - \mathbf{m}_i$ for convenience. The probability that the i-th cell is occupied is $P(\mathbf{m}_i)$, and the probability that it is free is $P(\bar{\mathbf{m}}_i) = 1 - P(\mathbf{m}_i)$. The random variables $\mathbf{m}_i$ are assumed as mutually independent variables, i.e.,

$$P(m) = P(\mathbf{m}_1, \mathbf{m}_2, ..., \mathbf{m}_{n_m}) = \prod_{i=1}^{n_m} P(\mathbf{m}_i). \tag{1}$$

Occupancy grid mapping is the process of determining these probabilities based on robot poses and measurement scans. More explicitly, let t be a discrete variable for time. The t-th pose is denoted $X_t = \{x_t, R_t\}$, where the planar position and direction of the robot at t are denoted by $x_t \in \mathbb{R}^2$ and $R_t \in \mathsf{S}^1 = \left\{q \in \mathbb{R}^2 | \, \|q\| = 1\right\}$, i.e., the attitude corresponds to a direction on a 2D plane. Let $X_{1:t}$ denote the history of poses from the initial time to the current time, i.e., $X_{1:t} = \{X_1, X_2, \ldots, X_t\}$. At each pose, the robot receives a 2D measurement *scan*, composed of n_z measurement *rays*. These rays are 1D depth measurements of known direction from the current pose to the closest occupied space, subject to a known *forward sensor model* $p(z_{t,l}|m, X_t)$, where $z_{t,l}$ is the l-th measurement ray of the t-th scan $Z_t = \{z_{t,1}, z_{t,2}, \ldots, z_{t,n_z}\}$ and $Z_{1:t}$ is the measurement history.

The goal of occupancy grid mapping is to obtain the *inverse sensor model* $P(\mathbf{m}_i|X_{1:t}, Z_{1:t})$ for all grid cells inside the sensor field of view. Suppose that the inverse sensor model before t-th time step is given, namely $P(m|X_{1:t-1}, Z_{1:t-1})$, and the l-th measurement ray $z_{t,l}$ is available from pose X_t. The inverse sensor model due to $z_{t,l}$ at the t-th time step follows Bayes' rule,

$$\begin{aligned} &P(m|z_{t,l}, X_{1:t}, Z_{1:t-1}) \\ &\quad = \frac{p(z_{t,l}|m, X_{1:t}, Z_{1:t-1}) P(m|X_{1:t-1}, Z_{1:t-1})}{p(z_{t,l}|X_{1:t}, Z_{1:t-1})} \\ &\quad = \eta_{t,l} p(z_{t,l}|m, X_{1:t}, Z_{1:t-1}) P(m|X_{1:t-1}, Z_{1:t-1}), \end{aligned} \tag{2}$$

where $\eta_{t,l}$ is a normalizing constant independent of m, and we have used the fact that X_t carries no information about m without $z_{t,l}$. The issue is that $P(\mathbf{m}_i|z_{t,l}, X_{1:t}, Z_{1:t-1})$ is perceived as intractable to calculate because the exact solution considers every combination of map m to obtain the normalizer $\eta_{t,l}$, and therefore this approach yields $\mathcal{O}(2^{n_m})$ operations per grid cell. This perception is why heuristic solutions based on learning or intuition substitute the inverse sensor model. Recently, (2) was solved in a computationally-efficient manner to obtain an exact solution in real-time [15], which is summarized next.

Consider a reduced map, namely r_l, composed of only $n_{r,l}$ grid cells that the l-th measurement ray intersects, indexed by increasing distance from the robot. Let the k-th cell of r_l, namely $\mathbf{r}_{l,k}$, correspond to $\mathbf{m}_i$, i.e., $\mathbf{r}_k = \mathbf{m}_i$ and $\bar{\mathbf{r}}_k = \bar{\mathbf{m}}_i$. From [15], the inverse sensor model of the individual grid cell $\mathbf{m}_i$ is

$$\begin{aligned} &P(\mathbf{m}_i|z_{t,l}, X_{1:t}, Z_{1:t-1}) \\ &= \eta_{t,l} P(\mathbf{r}_{l,k}|X_{1:t-1}, Z_{1:t-1}) \\ &\quad \times \Bigg[\sum_{i=1}^{k-1} \left\{ \prod_{j=0}^{i-1} P(\bar{\mathbf{r}}_{l,j}|X_{1:t-1}, Z_{1:t-1}) \right\} \\ &\quad \times p(z_{t,l}|\mathbf{r}_{l,i+}, X_t) P(\mathbf{r}_{l,i}|X_{1:t-1}, Z_{1:t-1}) \\ &\quad + \left\{ \prod_{j=0}^{k-1} P(\bar{\mathbf{r}}_{l,j}|X_{1:t-1}, Z_{1:t-1}) \right\} p(z_{t,l}|\mathbf{r}_{l,k+}, X_t) \Bigg], \end{aligned} \tag{3}$$

where $\mathbf{r}_{l,k+}$ corresponds to all r_l such that cells with index less than k are free, the k-th cell is occupied, and cells with greater index are irrelevant. Put differently, the k-th cell is the closest occupied cell along the l-th ray. The normalizer used in (3) is defined as

$$\begin{aligned} \eta_{t,l} = \Bigg[&\sum_{i=1}^{n_{r,l}+1} \left\{ \prod_{j=0}^{i-1} P(\bar{\mathbf{r}}_{l,j}|X_{1:t-1}, Z_{1:t-1}) \right\} \\ &\times p(z_{t,l}|\mathbf{r}_{l,i+}, X_t) P(\mathbf{r}_{l,i}|X_{1:t-1}, Z_{1:t-1}) \Bigg]^{-1}. \end{aligned} \tag{4}$$

This approach to obtain the inverse sensor model is advantageous because of computational efficiency and accuracy, obtaining the exact solution in $\mathcal{O}(n_{r,l} + 1)$ operations for *all* grid cells visible by this measurement ray. In occupancy grid mapping, one may use (3) and (4) in real-time to avoid common approximations or learned solutions.

B. Shannon's Entropy and Map Information

Since occupancy grid mapping provides a probabilistic representation of surrounding space, this mapping scheme holds probabilistic information about the uncertainty of the map. Shannon's entropy is commonly used as a measure of uncertainty, such as in [12], [13]. Given the probabilities of each grid cell of map m, Shannon's entropy is defined as the sum of the entropy of each cell,

$$\begin{aligned} &H(P(m)) = \\ &\quad - \sum_{i=1}^{n_m} \left\{ P(\mathbf{m}_i) \log P(\mathbf{m}_i) + P(\bar{\mathbf{m}}_i) \log \left(P(\bar{\mathbf{m}}_i) \right) \right\}. \end{aligned} \tag{5}$$

The entropy of the i-th grid cell is maximized when $P(\mathbf{m}_i) = 0.5$ (greatest uncertainty) and minimized when $P(\mathbf{m}_i) \in \{0, 1\}$ (smallest uncertainty).

C. Autonomous Exploration

The goal of autonomous exploration is to determine future poses that maximize the map information gain, or equivalently minimize the map entropy. Since autonomous exploration is conducted in uncertain environments, a complete trajectory cannot be known before the robot takes measurements of the entire reachable space. Instead, an objective function governs the motion between intermediate poses, building an occupancy grid map along the way.

Suppose at time t, $P(m|X_{1:t}, Z_{1:t})$ is given. Let X_c be the pose of the robot after the t-th time step. The expected information gain $\mathcal{I}(X_c)$ is defined as the negative change

$$\begin{aligned}\mathcal{I}(X_c) = &H(P(m|X_{1:t}, Z_{1:t})) \\ &- \mathrm{E}\left[H(P(m|X_{1:t}, Z_{1:t}, X_c, Z_c))\right], \end{aligned} \tag{6}$$

where $H(P(m|X_{1:t}, Z_{1:t}))$ may be computed with (5) and $\mathrm{E}\left[H(P(m|X_{1:t}, Z_{1:t}, X_c, Z_c))\right]$ is the expected entropy when the robot is located at X_c, where the expectation is taken over Z_c. Since autonomous exploration is formulated as a trajectory optimization, the goal is to repeatedly choose optimal X_c^* that maximizes the objective function

$$X_c^* = \underset{X_c}{\operatorname{argmax}} \mathcal{I}(X_c), \tag{7}$$

where X_c^* satisfies the inequality constraint to avoid collisions,

$$P_{\text{collision}}(X) = 1 - \prod_{i \in \mathcal{C}_X} \left(P(\bar{\mathbf{m}}_i|X_{1:t}, Z_{1:t}) \leq \beta, \right. \tag{8}$$

where $\mathcal{C}_X$ is the set of grid cells falling inside a volume of preselected size around X that may cause collision and $\beta > 0$ is a small acceptable probability of collision. Once the robot has translated to X_c^*, the above process is repeated.

III. 1D Ray Expected Information Gain

In this section, we present a method to determine the expected entropy from an individual measurement ray from a known pose. This method is based on the exact solution to occupancy grid mapping from [15].

A. Single Ray Expected Value of Entropy

We consider the expected entropy due to the l-th ray $z_{c,l}$ at position x_c,

$$\begin{aligned}&\mathrm{E}[H(P(m|X_{1:t}, Z_{1:t}, x_c, z_{c,l}))] \\ &\quad = \int_{z_{\min}}^{z_{\max}} H(P(m|X_{1:t}, Z_{1:t}, x_c, z_{c,l})) \\ &\qquad \times p(z_{c,l}|X_{1:t}, Z_{1:t}, x_c) dz_{c,l}. \end{aligned} \tag{9}$$

We discretize the measurement ray space such that $z_{c,l}$ falls on points along the measurement ray intersecting with grid cell edges, similar with how occupancy grid mapping decomposes continuous space into grid cells of length α. The discretized expected value of (9) is

$$\begin{aligned}&\mathrm{E}[H(P(m|X_{1:t}, Z_{1:t}, x_c, z_{c,l}))] \\ &\quad = \sum_{k=1}^{n_{r,l}+1} \Big\{ H(P(m|X_{1:t}, Z_{1:t}, x_c, z_{c,l,k})) \\ &\qquad \times P(z_{c,l,k}|X_{1:t}, Z_{1:t}, x_c) \Big\}, \end{aligned} \tag{10}$$

where index $z_{c,l,k}$ is the distance from x_c to the k-th grid cell of the l-th reduced map r_l, known from geometry.

Standing alone, the term $P(z_{c,l,k}|X_{1:t}, Z_{1:t}, x_c)$ from (10) has a convoluted meaning because the depth $z_{c,l,k}$ does not directly depend on the map. However, we present a method to obtain this discretized probability. Following the assumption that $z_{c,l}$ is discretized to known distances, the probabilities are proportional to their densities, as the area under the density curve is infinitesimal and fixed. Accounting for all cases, the probability is

$$P(z_{c,l,k}|X_{1:t}, Z_{1:t}, x_c) = \frac{p(z_{c,l,k}|X_{1:t}, Z_{1:t}, x_c)}{\sum_{i=1}^{n_{r,l}+1} p(z_{c,l,i}|X_{1:t}, Z_{1:t}, x_c)}. \tag{11}$$

Conveniently, this density corresponds to the inverse normalizer defined in (2),

$$p(z_{c,l,k}|X_{1:t}, X_c, Z_{1:t}) = \eta_{c,l,k}^{-1}, \tag{12}$$

where $\eta_{c,l,k}^{-1}$ is defined in (4). However, instead of evaluating the t-th pose X_t and measured ray $z_{t,l}$, we simply consider X_c and ray depth $z_{c,l,k}$ based on histories $X_{1:t}$ and $Z_{1:t}$ to obtain the inverse normalizer,

$$\begin{aligned}\eta_{c,l,k}^{-1} = &\sum_{i=1}^{n_{r,l}+1} \left\{ \prod_{j=0}^{i-1} P(\bar{\mathbf{r}}_{l,j}|X_{1:t}, Z_{1:t}) \right\} \\ &\times p(z_{c,l,k}|\mathbf{r}_{l,i+}, X_c) P(\mathbf{r}_{l,i}|X_{1:t}, Z_{1:t}). \end{aligned} \tag{13}$$

By substituting (11) and (12) into (10), the expected entropy at X_c from individual measurement ray $z_{c,l}$ is

$$\begin{aligned}\mathrm{E}[H(P(m|X_{1:t}, Z_{1:t}, x_c, z_{c,l}))] = &\left(\sum_{i=1}^{n_{r,l}+1} \eta_{c,l,i}^{-1} \right)^{-1} \\ \times &\sum_{k=1}^{n_{r,l}+1} \left\{ H(P(m|X_{1:t}, Z_{1:t}, x_c, z_{c,l,k})) \eta_{c,l,k}^{-1} \right\}, \end{aligned} \tag{14}$$

where $\eta_{c,l,k}^{-1}$ is taken from (13). Here, (14) provides the expected entropies for those cells inside the field of view of r_l. The entropies of cells outside the field of view remain unchanged.

B. Approximation of Expected Ray Entropy

The order of computation for each measurement ray is $\mathcal{O}((n_{r,l}+1)^2)$ since the summations of (13) are embedded in (14). Such computational requirements might be too cumbersome for some processors, especially if the measurement ray is expected to intersect many grid cells. Many of those intersections provide negligible information since the

probability of the l-th measurement ray capturing certain cell depths is close to zero.

The approximation of expected ray entropy provides a method to reduce the computation of (13) substantially. This goal is achieved by systematically selecting a smaller set of grid cells to consider over the summations of (13) and (14). The smaller set is determined by the probability that each cell is captured by the measurement ray, known as the detection probability. This can be found recursively as

$$P(\mathbf{r}_{l,k+}|X_{1:t}, Z_{1:t}) = \left\{ \prod_{j=0}^{k-1} P(\bar{\mathbf{r}}_{l,j}|X_{1:t}, Z_{1:t}) \right\} P(\mathbf{r}_{l,k}|X_{1:t}, Z_{1:t}), \quad (15)$$

which is the probability that $\mathbf{r}_{l,k}$ is the closest occupied grid cell based on past poses and measurement scans, independent of cells beyond the k-th cell from x_c. Let $\hat{n} > 0$ be a fixed number of grid cells for all rays such that $\hat{n} \leq n_{r,l} + 1$ for all l. Let $\hat{r}_l$ correspond to the grid cells that yield the $\hat{n}$ maximum values (15) (the $\hat{n}$ most likely ray detections), indexed by increasing distance from candidate location x_c. By replacing the reduced map r_l with $\hat{r}_l$ and changing the summation limits to $\{1, 2, ..., \hat{n}\}$ in (13) and (14), the order of computation is reduced to $\mathcal{O}(\hat{n}^2)$. Even though the value of $n_{r,l}$ is different among various rays in general, $\hat{n}$ is fixed, so the computational order is fixed as well. In short, this method reduces the required computation substantially by systematically neglecting those grid cells with little effect.

C. 1D Ray Expected Information Gain Algorithm

We present an algorithm pseudocode providing the necessary steps to obtain the objective function for a single measurement ray (Table I). The variable a_{temp} is an intermediate variable designed to avoid repeated calculations. Since this algorithm operates as a function, fixed indices and condition variables are removed for simplification. It can be noted that if $\hat{n} = n_{r,l}+1$, the ray objective function is computed without approximation.

D. Numerical Justification for Approximation

The purpose of this numerical example is to provide evidence that the approximations are reasonable and speed up the algorithm substantially. Since a measurement ray produces a depth measurement in a single direction, we only consider a 1D map where the grid cells have spacing $\alpha = 0.2$ meters, and the properties of the range sensor are based off the Microsoft Kinect [10], [18] with maximum reading depth $z_{\max} = 4$m (20 grid cells inside the sensor FOV). The goal is to compare the expected entropy $\mathrm{E}[H(P(m|X_{1:t}, Z_{1:t}, x_c, z_{c,l}))]$ from (14) and with an approximation $\mathrm{E}[H_{\text{approx}}(P(m|X_{1:t}, Z_{1:t}, x_c, z_{c,l}))]$, which only considers $\hat{n}$ grid cells with highest detection probability (15).

We consider 100 probabilistic maps to obtain Monte Carlo results to evaluate the approximate entropy. In every Monte Carlo trial, each grid cell has an 80% chance of being free, and the remaining 20% of grid cells are given an a priori probability uniformly distributed between 0 and 1. Several metrics serve to evaluate $\mathrm{E}[H(P(m|X_{1:t}, Z_{1:t}, x_c, z_{c,l}))]$ with $\mathrm{E}[H_{\text{approx}}(P(m|X_{1:t}, Z_{1:t}, x_c, z_{c,l}))]$. The median expected entropy change is $\mathrm{E}[H(P(m|X_{1:t}, Z_{1:t}, x_c, z_{c,l}))] - H(P(m|X_{1:t}, Z_{1:t})) = -0.83792$. The error for the 100 Monte Carlo cases is defined simply as

$$e_H = \frac{1}{100} \sum_{i=1}^{100} \text{abs} \Big(\mathrm{E}[H(P(m|X_{1:t}, Z_{1:t}, x_c, z_{c,l}))] - \mathrm{E}[H_{\text{approx}}(P(m|X_{1:t}, Z_{1:t}, x_c, z_{c,l}))] \Big). \quad (16)$$

The Monte Carlo trials are repeated for $\hat{n} = \{1, 2, \ldots, 10\}$ and the results are plotted in Figure 1. This example shows a typical case when the summation limits generated from (15) have only small effects on (14), while providing very large improvements in reducing computation.

Function: $\mathcal{I}_{\text{ray}} = \text{RayExpInfoGain}(x, P(r), z_{1:n_{r,l}})$
Define $P(\bar{\mathbf{r}}_0) = P(\bar{\mathbf{r}}_{n_{r,l}+1}) = 1$;
for $k = 1, 2, \ldots, n_{r,l} + 1$
  $P(\mathbf{r}_{k+}) = P(\bar{\mathbf{r}}_{0:k-1})P(\mathbf{r}_k)$;
  $P(\bar{\mathbf{r}}_{0:k}) = P(\bar{\mathbf{r}}_{0:k-1})P(\bar{\mathbf{r}}_k)$;
end for
Find $\hat{r} \subset r$ corresponding to the $\hat{n}$ greatest values of $\left\{P(\mathbf{r}_{1+}), P(\mathbf{r}_{2+}), \ldots, P(\mathbf{r}_{(n_{r,l}+1)+})\right\}$;
Define $P(\hat{\bar{\mathbf{r}}}_0) = 1$;
for $k = 1, 2, \ldots, \hat{n}$
  $P(\hat{\mathbf{r}}_{k+}) = P(\hat{\bar{\mathbf{r}}}_{0:k-1})P(\hat{\mathbf{r}}_k)$;
  $P(\hat{\bar{\mathbf{r}}}_{0:k}) = P(\hat{\bar{\mathbf{r}}}_{0:k-1})P(\hat{\bar{\mathbf{r}}}_k)$;
end for
for $k_{\text{m}} = 1, 2, \ldots, \hat{n}$
  Initialize $\eta_{k_{\text{m}}}^{-1} = 0$;
  for $k_{\text{c}} = 1, 2, \ldots, \hat{n}$
    $a_{\text{temp}} = P(\hat{\mathbf{r}}_{k_{\text{c}}+})p(z_{k_{\text{m}}}|\hat{\mathbf{r}}_{k_{\text{c}}+}, x)$;
    $\tilde{P}(\hat{\mathbf{r}}_{k_{\text{c}}}|x, z_{k_{\text{m}}}) = P(\hat{\mathbf{r}}_{k_{\text{c}}})\eta_{k_{\text{m}}}^{-1} + a_{\text{temp}}$;
    $\eta_{k_{\text{m}}}^{-1} = \eta_{k_{\text{m}}}^{-1} + a_{\text{temp}}$;
  end for
  $P(\hat{\mathbf{r}}_{k_{\text{c}}}|x, z_{k_{\text{m}}}) = \eta_{k_{\text{m}}}\tilde{P}(\hat{\mathbf{r}}_{k_{\text{c}}}|x, z_{k_{\text{m}}})$ for all $k_{\text{c}} = 1, 2, \ldots, \hat{n}$;
end for
$P(z_{k_{\text{m}}}|x) = \frac{\eta_{k_{\text{m}}}^{-1}}{\sum_{i=1}^{\hat{n}} \eta_i^{-1}}$ for $k_{\text{m}} = 1, 2, \ldots, \hat{n}$;
Initialize $\mathcal{I}_{\text{ray}} = 0$;
for $k_{\text{c}} = 1, 2, \ldots, \hat{n}$
  $\mathcal{I}_{\text{ray}} = \mathcal{I}_{\text{ray}} + H(P(\hat{\mathbf{r}}_{k_{\text{c}}}))$;
  for $k_{\text{m}} = 1, 2, \ldots, \hat{n}$
    $\mathcal{I}_{\text{ray}} = \mathcal{I}_{\text{ray}} - H(P(\hat{\mathbf{r}}_{k_{\text{c}}}|x_c, z_{k_{\text{m}}}))P(z_{k_{\text{m}}}|x_c)$;
  end for
end for
Return: $\mathcal{I}_{\text{ray}}$

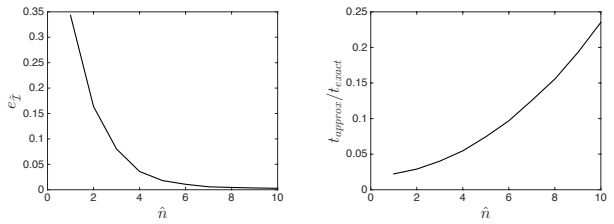
TABLE I
EXPECTED INFORMATION GAIN FROM A MEASUREMENT RAY

IV. AUTONOMOUS EXPLORATION

In this section, we develop an autonomous exploration scheme on a 2D map based on choosing movements that minimize entropy and avoid collisions.

A. Pose Selection Optimization

The robot pose at a future time step is selected to maximize (6) over the search space $\mathbb{R}^2 \times \mathsf{S}^1$. We reduce the search by discretizing the search space in both position and



(a) Entropy Error (b) Computation Ratio

Fig. 1. The entropy error is decreased at the cost of increasing computation time in this Monte Carlo 1D measurement ray expected entropy case.

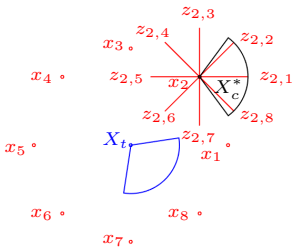


Fig. 2. Initially, a robot (blue circle) views down and left (blue sector). Then, the robot considers $n_c = 8$ poses (red circles) as possible candidates for where to move next. For each candidate, $n_d = 8$ directions are considered (red lines, only displayed on candidate location $c = 2$). Then, the expected entropy of each ray is calculated. The scan (red sector) covering those rays with the largest expected entropy decreases is chosen for each candidate, and the best candidate X_c^* (black circle: location, black sector: scan) is chosen to maximize information gain. Finally, Dijkstra's algorithm provides the collision-free motion between X_t and X_c^*, and the process is repeated.

attitude as follows. First, we choose n_c candidate locations evenly-spaced along a circle of radius δ, centered around the current location x_t, where the c-th candidate location is denoted $x_c \in \{x_1, x_2, \ldots, x_{n_c}\}$, illustrated in Figure 2. Any location that violates (8) is excluded to avoid collisions. For each collision-free candidate location, we consider n_d evenly-spaced directions. The rays and attitudes at x_c are denoted $\{z_{c,1}, z_{c,2}, \ldots, z_{c,n_d}\}$ and $\{R_{c,1}, R_{c,2}, \ldots, R_{c,n_d}\}$, respectively, where the scan with attitude $R_{c,d}$ might cover several ray directions depending on the sensor field of view (FOV). For each candidate location, we choose optimal attitude R_c^* as the summation of the expected entropy changes covered by the scan,

$$R_c^* = \underset{R_{c,d}}{\operatorname{argmax}} \sum_{z_{c,i} \in R_{c,d} \text{ FOV}} \Big(H(P(m|X_{1:t}, Z_{1:t})) - \mathrm{E}[H(P(m|X_{1:t}, Z_{1:t}, x_c, z_{c,i}))] \Big). \quad (17)$$

Therefore, the optimal attitude R_c^* is given as a function of the location x_c. The objective function (6) is computed by a summation about the n_d rays as

$$\mathcal{I}(x_c, R_c^*(x_c)) \approx \sum_{z_{c,i} \in R_c^* \text{ FOV}} \Big(H(P(m|X_{1:t}, Z_{1:t})) - \mathrm{E}\left[H(P(m|X_{1:t}, Z_{1:t}, x_c, R_c^*(x_c), z_{c,i}))\right] \Big), \quad (18)$$

$$x_c^* = \underset{x_c}{\operatorname{argmax}} \mathcal{I}(x_c, R_c^*(x_c)). \quad (19)$$

The objective function of the optimal pose $X_c^* = \{x_c^*, R_c^*\}$ must satisfy $\mathcal{I}(X_c^*) \geq \mathcal{I}_{\text{min}}$ to avoid motions where little information is expected to be gained. If this condition is not satisfied, then n_c and δ are multiplied by a scale function $\lambda > 1$, candidates are chosen again, and $\mathcal{I}(X_c^*)$ is reevaluated. Once the condition is satisfied, the robot must move to this pose while avoiding collisions subject to the inequality constraint (8). In certain robotic applications, other cost parameters might be included in the optimization, e.g., driving time.

We choose Dijkstra's algorithm [19] because it provides a relatively simple motion planning strategy from x_t to x_c^* that avoids local minima. Once the robot completes this motion, the entire process is repeated. This is summarized in the autonomous exploration pseudocode in Table II.

Given: $t = 1$, X_1, Z_1, $P(m|X_1, Z_1)$ from the inverse sensor model
TrajectoryComplete = false;
while TrajectoryComplete = false
  if X_{t+1} is unknown
    MotionPlanning = false;
    while MotionPlanning = false
      for $c = 1, 2, \ldots, n_c$
        Candidate location: x_c from evenly-spaced circle around X_t with radius δ;
        for $d = 1, 2, \ldots, n_d$
          Obtain reduced map $r_{c,d}$ with $n_{r_{c,d}}$ intersections through grid cells with depths $z_{c,d,1:n_{r,d}}$ from geometry;
          $\mathcal{I}_{\text{ray}}(x_c, z_{c,d}) = \text{RayExpInfoGain}(x_c, P(r_{c,d}), z_{c,d,1:n_{r,d}})$;
        end for
        $R_c^* = \operatorname{argmax}_{R_c} \left(\sum_{z_{c,i} \in R_c \text{ FOV}} \mathcal{I}_{\text{ray}}(x_c, z_{c,i}) \right)$;
        $\mathcal{I}(x_c, R_c^*) = \sum_{z_{c,i} \in R_c^* \text{ FOV}} \mathcal{I}_{\text{ray}}(x_c, z_{c,i})$;
      end for
      $x_c^* = \operatorname{argmax}_{x_c} \mathcal{I}(x_c, R_c^*)$;
      $X_c^* = \{x_c^*, R_c^*\}$;
      if $\mathcal{I}(X_c^*) < \mathcal{I}_{\text{min}}$
        $n_c = \text{round}(\lambda n_c)$;
        $\delta = \lambda \delta$;
      else
        Obtain $X_{t+1}, X_{t+2}, \ldots, X_c^*$ using Dijkstra's algorithm planner and minimum rotation;
        Reset n_c and δ to initial values;
        MotionPlanning = true;
      end if
    end while
  end if
  t = t+1;
  Obtain $P(m|X_{1:t}, Z_{1:t})$ from the inverse sensor model;
  if map m is completely explored
    TrajectoryComplete = true;
  end if
end while

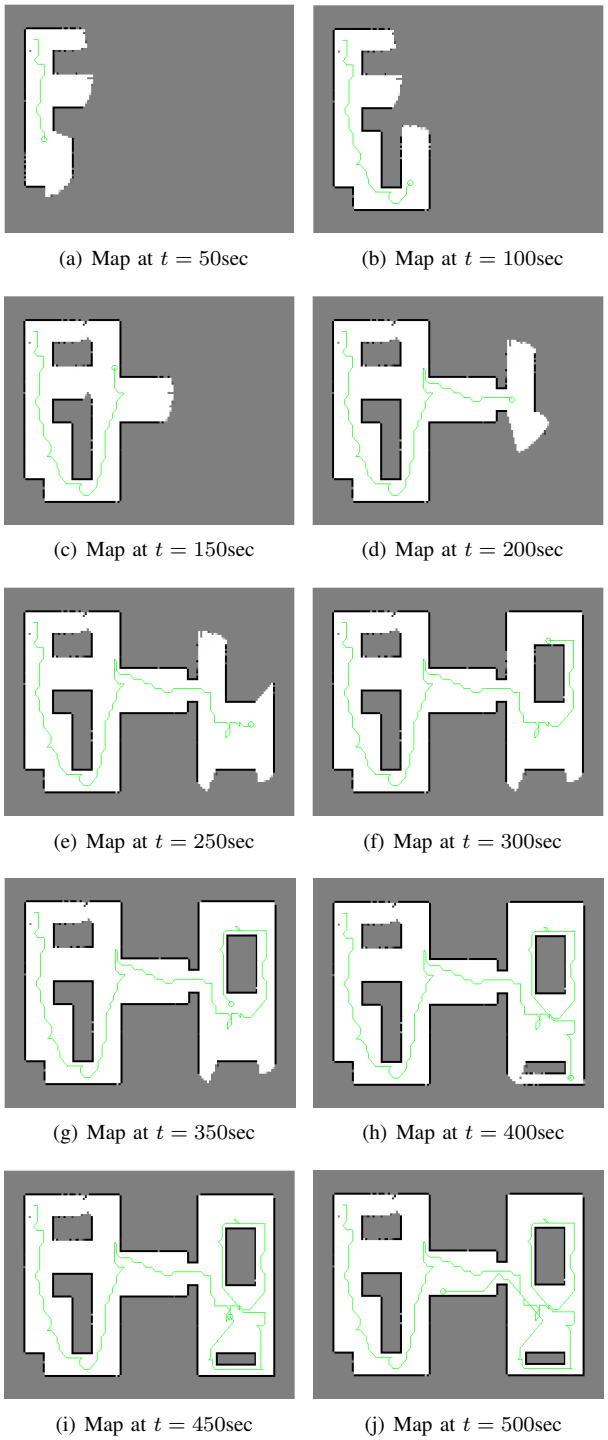
TABLE II
AUTONOMOUS EXPLORATION VIA EXPECTED INFORMATION GAIN

B. Numerical Example

The robot models its surroundings with an occupancy grid with 15000 cells where grid cell edges are $\alpha = 0.2$m, composing a map with dimensions 30m $\times$ 20m. The initial probability $P(\mathbf{m}_i) = 1 \times 10^{-10} \approx 0$ (minimum value for free space) for grid cells covered by the circular robot of radius 0.1m and $P(\mathbf{m}_i) = 0.5$ for all other cells. At each time step, the robot receives a measurement scan, where the probabilistic properties of the sensor are taken from [10], [18]. Then, the $n_c = 8$ evenly-spaced candidate locations about a circle of radius $\delta = 0.5$m around the current pose location are considered, where $n_d = 32$ measurement rays are evenly-spaced about the candidate location. When no current candidates yield expected information gains above $\mathcal{I}_{\text{min}} = 2$, $\lambda = 1.25$ is multiplied to the current n_c and δ values. The motion of the robot is restricted to movement on grid cells satisfying (8) with $\beta = 0.01$. Dijkstra's algorithm for robotic motion planning guides the robot through the environment without collision. The results are illustrated in Figure 3.

Knowing only that the robot is inside free space at the beginning, the robot carefully navigates the environment while avoiding collisions. The robot motion is governed by a policy that maximizes the map information gain within its set of pose candidates, where the $\hat{n} = 6$ is chosen to approximate the expected entropy of each ray. When running the exploration algorithm in Robot Operating System (ROS), the mean computation time is 0.0194 seconds to determine the optimal future pose and complete Dijkstra's algorithm on the occupancy grid. The computation times for this map reached roughly 1 second at maximum, but this corresponds to cases when the robot is locally surrounded by a highly-certain environment; the robot must traverse a large distance to learn information about another part of the map. Such a task requires evaluating many more pose candidates and motion planning over a larger terrain. Thus, the efficient computation makes the proposed autonomous exploration algorithm an effective strategy for real-time implementation.

Throughout the numerical example, the robot must make several important decisions on which actions to take. These actions are evaluated based on their expected information gains, not frontiers a predicted measurement scans. If the obstacles were known a priori, the motion planning problem could be solved with a smoother path; however, the autonomous exploration is based on only the information of the map that the robot generates, so the motion planning must be reevaluated repeatedly. Even with these limitations, the robot explores the vast majority of reachable space in the 500 second period.



(a) Map at $t = 50$sec (b) Map at $t = 100$sec

(c) Map at $t = 150$sec (d) Map at $t = 200$sec

(e) Map at $t = 250$sec (f) Map at $t = 300$sec

(g) Map at $t = 350$sec (h) Map at $t = 400$sec

(i) Map at $t = 450$sec (j) Map at $t = 500$sec

Fig. 3. A robot (green circle, green marker of its previous pose locations) measures a room with a Kinect depth sensor. The robot moves to maximize its information gain. In the end, the robot is returning to the left room to gain more information about the uncertain regions it originally left behind.

V. Conclusions

We developed a novel approach to autonomous exploration through an occupancy grid map based on a direct solution to expected entropy gain. Unlike common techniques based on finding frontiers or predicting measurement scans with particle filters, the proposed approach is based on an expected value of the entropy directly. The approach is approximated

with large improvements in computation and small losses in accuracy. A numerical example shows how a robot navigates an uncertain map governed by the proposed autonomous exploration algorithm.

ACKNOWLEDGMENT

The authors would like to acknowledge Dr. Ira S. Moskowitz for helpful discussions on this topic.

REFERENCES

[1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, ser. Intelligent Robotics and Autonomous Agents. Cambridge, Massachusetts: Massachusetts Institute of Technology, 2005.

[2] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part 1," *IEEE Robotics and Automation Magazine*, vol. 13, no. 2, pp. 99–110, June 2006.

[3] C. Chen and Y. Cheng, "Research of mobile robot slam based on ekf and its improved algorithms," in *Third International Symposium on Intelligent Information Technology Application*, vol. 1. IEEE, November 2009, pp. 548–552.

[4] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *International Symposium on Computational Intelligence in Robotics and Automation*. IEEE, 1997, pp. 146–151.

[5] ——, "Frontier-based exploration using multiple robots," in *Second International Conference on Autonomous Agents*, ACM, Ed., 1998, pp. 47–53.

[6] H. P. Moravec and A. Elfes, "High resolution maps from wide angle sonar," in *IEEE Conference on Robotics and Automation*, 1985.

[7] A. Elfes, "Using occupancy grids for mobile robot perception and navigation," *IEEE Computer*, pp. 46–57, 1989.

[8] H. Choset, K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*, ser. Intelligent Robotics and Autonomous Agents. Cambridge, Massachusetts: Massachusetts Institute of Technology, 2005.

[9] F. Andert, "Drawing stereo disparity images into occupancy grids: Measurement model and fast implementation," in *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009.

[10] K. Pirker, M. Ruther, H. Bischof, and G. Schweighofer, "Fast and accurate environment modeling using three-dimensional occupancy grids," in *Proceedings of the 2011 IEEE International Conference on Computer Vision Workshops*, 2011.

[11] S. Thrun, "Learning occupancy grids with forward models," in *Proceedings of the 2001 IEEE/RSU International Conference on Intelligent Robots and Systems*, 2001.

[12] H. Carrillo, P. Dames, V. Kumar, and J. A. Castellanos, "Autonomous robotic exploration using occupancy grid maps and graph SLAM based on Shannon and Rényi entropy," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015.

[13] C. Stachniss, G. Grisetti, and W. Burgard, "Information gain-based exploration using Rao-Blackwellized particle filters," in *In RSS*, 2005, pp. 65–72.

[14] D. Joho, C. Stachniss, P. Pfaff, and W. Burgard, "Autonomous exploration for 3D map learning," in *Autonome Mobile Systeme (AMS)*, K. Berns and T. Luksch, Eds. Springer, 2007, pp. 22–28.

[15] E. Kaufman, T. Lee, Z. Ai, and I. S. Moskowitz, "Bayesian occupancy grid mapping via an exact inverse sensor model," in *American Control Conference*. IEEE, 2016, pp. 5709–5715.

[16] J. J.L. Blanco, J.A. Fernández-Madrigal, "A novel measure of uncertainty for mobile robot slam with rao–blackwellized particle filters," *International Journal of Robotics Research*, vol. 27, no. 1, pp. 73–81, 2008.

[17] M. K. N. B. B. M. I. L. Carlone, Jingjing Du, "Active slam and exploration with particle filters using kullback-leibler divergence," *Journal of Intelligent & Robotic Systems*, vol. 75, no. 2, pp. 291–311, 2014.

[18] K. Khoshelham and S. O. Elberink, "Accuracy and resolution of kinect depth data for indoor mapping applications," *Sensors*, pp. 1437–1454, 2012.

[19] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.