J Intell Robot Syst
https://doi.org/10.1007/s10846-017-0710-7

Autonomous Exploration with Exact Inverse Sensor Models

Evan Kaufman[1] · Kuya Takami[1] · Taeyoung Lee[1] · Zhuming Ai[2]

Received: 21 December 2016 / Accepted: 28 August 2017

Abstract This paper is focused on probabilistic occupancy grid mapping and motion planning such that a robot may build a map and explore a target area autonomously in real time. The desired path of the robot is developed in an optimal fashion to maximize the information gain from the sensor measurements on its path, thereby increasing the accuracy and efficiency of mapping, while explicitly considering the sensor limitations such as the maximum sensing range and viewing angle. Most current exploration techniques require frequent human intervention, often developed for omnidirectional sensors with infinite range. The proposed research is based on realistic assumptions on sensor capabilities. The unique contribution is that the mapping and autonomous exploration techniques are systematically developed in a rigorous, probabilistic formulation. The mapping approach exploits the probabilistic properties of the sensor and map explicitly, and the autonomous exploration is designed to maximize the expected map information gain, thereby improving the efficiency of the mapping procedure and the quality of the map substantially. The efficacy of the proposed optimal approach is illustrated by both numerical simulations and experimental results.

Keywords Occupancy grid mapping · Inverse sensor model · Autonomous exploration

1 Introduction

Robotic mapping is the process of generating maps, which represent the environment in close proximity of a robot. This problem is crucial to understanding the surrounding regions when a robot enters an uncertain environment. Various map representations include occupancy grids [1], octomaps [2], or feature-based maps [3].

For mobile robots to operate in an unknown environment autonomously, they should be able to construct a map of the environment by using onboard sensors, and estimate their position in the constructed map. Simultaneous localization and mapping problem (SLAM) serves to estimate the map of the environment and the pose of robots concurrently, and there have been various approaches developed for SLAM with successful demonstrations.

However, there are certain issues with the current SLAM techniques. First, the vast majority of work deals only with the aspect of estimating the environment and the poses of vehicles. They are passive in the sense that SLAM is performed on incoming sensor measurements from vehicles following an arbitrary path. As such, human teleoperation and monitoring are often essential to guide the vehicles safely through unknown surroundings. Therefore, it is desired that an active motion planning scheme is developed and integrated with SLAM such that vehicles are able to determine their path without human intervention, and explore unknown areas autonomously.

✉ Evan Kaufman
evankaufman@gwu.edu

Kuya Takami
kuya@gwu.edu

Taeyoung Lee
tylee@gwu.edu

[1] Department of Mechanical and Aerospace Engineering, The George Washington University, Washington, DC 20052, USA

[2] Information Management & Decision Architectures, U.S. Naval Research Laboratory, Washington, DC 20375, USA

Published online: 28 October 2017

To achieve these, it is proposed that the problems of mapping and motion planning are integrated in a probabilistic formulation. More explicitly, the map is constructed in a probabilistic sense, in contrast to the common approaches where the map is defined by deterministic objects that are definitely occupied or free. One of the desirable features for the proposed stochastic formulation is that we can extract the information on the degree of uncertainty for the each part of the map, such that the motion of the vehicle is planned to minimize the map uncertainty in an optimal fashion.

This is addressed in two steps. The first part of this paper focuses on probabilistic occupancy grid mapping given a stochastic model of a range sensor, and the second part deals with stochastic motion planning to construct optimal exploration strategies.

Probabilistic Occupancy Grid Mapping The proposed approach is based on the occupancy grid representation, where the environment is decomposed into evenly-spaced grid cells, considered as either occupied or free. The probabilistic model of a range sensor is referred to as the *forward sensor model*, which provides a probability of range measurements for the given pose of the vehicle and the exact map [4]. One can characterize the performance of a range sensor, such as its accuracy or maximum range, completely with its forward sensor model.

The mapping problem can be defined as finding the *inverse sensor model*, which is the probability of occupancy for a grid cell based on range measurements and the configuration of the mobile robot, which is of fundamental importance with occupancy grid mapping. However, the exact solution to the inverse sensor model has not been utilized in occupancy grid mapping, because the computational requirements are too cumbersome to compute the exact model in real time [4].

Instead, several techniques have been proposed to approximate the exact inverse sensor model. In the early occupancy grid mapping applications based on sonar sensors, *ad hoc* techniques to obtain an inverse sensor model were developed based off highly-innacurate bouncing sound waves [5, 6]. These techniques rely on approximate functions to represent the inverse sensor model with questionable accuracy [7] and applied to more modern sensors in [8, 9].

The other popular approach to obtain an approximate inverse sensor model is to simulate maps, robot poses, and measurements and use a learning algorithm to approximate the inverse sensor model [4, 10]. These approaches are undesirable in practice due to complexities associated with implementing a learning algorithm. For example, the accuracy of such inverse sensor models strongly depends on the samples selected for learning, but it is unclear how to select those samples, or how many samples are required to obtain a reasonable approximation. Furthermore, it is challenging to apply any learning algorithm over the large dimensional space composed of maps, poses, and measurements.

This paper proposes a computationally efficient algorithm to construct the exact inverse sensor model. More explicitly, for a given forward sensor model defined by the probability distribution of range measurements, this algorithm yields the a posteriori probabilities of occupancy of all the cells within the area covered by the range sensor from the range measurements. The key idea is reducing the computational load by using the fact that if a cell is occupied, the occupancies of the other cells blocked by it is irrelevant to the forward sensor model, and this property is systematically utilized with various probabilistic properties to derive a computationally-efficient solution to the inverse sensor model. Furthermore, the proposed approach integrates a priori probabilities of occupancy and multiple range measurements according to the Bayesian framework to obtain more accurate maps. As such, it contrasts from the existing framework based on log-odds ratios that impose an additional Markov assumption.

Optimal Autonomous Exploration Next, this paper addresses autonomous exploration to plan the trajectories of the vehicle to generate a complete, accurate map [11]. We present an approach where the 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 [11, 12]. 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.

Alternatively, there have been exploration techniques that are based on a quantitative measure of uncertainty. However, these approaches 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. [5–9]) or simulated learning techniques [4, 10]. Map-based information gain is commonly determined with Shannon's entropy [13], 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 or ad hoc learning. Furthermore, the expected entropy of an action is not directly calculated. 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. For these reasons, more careful computation of map uncertainty is warranted.

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 probabilistic solution to map information gain, under the assumption that the robot pose is known. Accounting for the stochastic properties of position and attitude is important to the field of robotics, and accounting for localization in conjunction with mapping and exploration are considered as a future step to this paper.

In short, the main contribution of this paper is proposing the exact inverse sensor model and constructing an exact Bayesian occupancy grid mapping based on that. Compared with the current mapping algorithms based on an approximate inverse sensor model, this approach yields more accurate occupancy probabilities. This constructs substantially more accurate occupancy grid maps with less uncertainty for the same set of measurements, and these are directly illustrated by numerical examples and experimental results. Additionally, we derive a computationally-efficient approach to predict map information gain for autonomous exploration with a novel structure that avoids several common assumptions and approximations. By computing the expected information precisely with the exact inverse sensor model, the proposed autonomous exploration approach constructs a more accurate map with limited sensor capabilities in an optimal fashion. While this paper is focused on mapping and exploration in two-dimensional environments, the presented results can be certainly utilized in SLAM or stochastic motion planning in three dimensions.

2 Problem Formulation

2.1 Probabilistic Occupancy Grid Mapping

Let a map m be decomposed into n_m evenly-spaced grid cells, where the i-th grid cell is assigned to a static binary random variable $\mathbf{m}_i$ for $i \in \{1, 2, \ldots, n_m\}$, 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, where a smaller cell size (greater grid resolution) better represents a space, but increases computation and memory. Therefore, a map m is defined by $\{\mathbf{m}_1, \ldots, \mathbf{m}_{n_m}\}$ (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 mutually independent,

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

Consider a range sensor that provides scans of the surrounding environment in order to identify the closest occupied space. The location and the direction of the sensor, referred to as the *pose* at time t, is denoted by $X_t = (x_t, R_t)$, 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 $Z_t = \left\{z_{t,1}, z_{t,2}, \ldots, z_{t,n_z}\right\}$. The measurement history is denoted by $Z_{1:t} = \{Z_1, Z_2, \ldots, Z_t\}$.

The probability density function, namely $p(z_{t,l}|m, X_t)$ with respect to the depth of the l-th measurement ray conditioned on the map m and the pose X_t is commonly referred to as the *forward sensor model*, which characterizes the corresponding depth sensor, such as the maximum range or accuracy. The forward sensor model satisfies (i) the ranges of all depth measurements are positive and finite, and (ii) measurement rays cannot pass through occupied regions. Throughout this paper, we assume that the forward sensor model of the selected sensor is given. This can be determined empirically or analytically. For example, the *beam model for range finders* satisfying the above criteria is described in [4].

Occupancy grid mapping provides occupancy probabilities based on robot poses and measurement scans. The goal is to obtain $P(\mathbf{m}_i|X_{1:t}, Z_{1:t})$, commonly referred to as the *inverse sensor model* for given forward sensor model

$p(z_{t,l}|m, X_t)$ and the initial estimate of the map $P(m)$ with $X_{1:t}$ and $Z_{1:t}$.

2.2 Autonomous Exploration

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 [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,

$$H(P(m)) = -\sum_{i=1}^{n_m}\{P(\mathbf{m}_i)\log P(\mathbf{m}_i) + P(\bar{\mathbf{m}}_i)\log(P(\bar{\mathbf{m}}_i))\}. \quad (2)$$

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).

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 a candidate pose under consideration for some future time step. The expected information gain $\mathcal{I}(X_c)$ is defined as the negative change of entropy

$$\mathcal{I}(X_c) = H(P(m|X_{1:t}, Z_{1:t})) - \mathrm{E}\left[H(P(m|X_{1:t}, Z_{1:t}, X_c, Z_c))\right], \quad (3)$$

where $H(P(m|X_{1:t}, Z_{1:t}))$ may be computed with Eq. 2 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

$$X_c^* = \underset{X_c}{\operatorname{argmax}}\,\mathcal{I}(X_c), \quad (4)$$

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

$$P_{\text{collision}}(X) = 1 - \prod_{i\in\mathcal{C}_X} P(\bar{\mathbf{m}}_i|X_{1:t}, Z_{1:t}) \le \beta, \quad (5)$$

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 threshold of the probability of collision. It is considered that once a collision-free trajectory toward X_c^* is obtained, the robot moves along the trajectory controlled by a motion control system in the inner loop. Once the robot has translated to X_c^*, the above process is repeated.

3 Occupancy Grid Mapping with an Exact Inverse Sensor Model

This section outlines issues with the common log-odds ratio formulation and perceived computational burdens with the exact solution to the inverse sensor model for occupancy grid mapping. Then, a method that exploits several mapping and probabilistic properties provides a computationally efficient exact solution to probabilistic occupancy grid mapping.

3.1 Mapping via Log-Odds Ratio

One of the popular frameworks of updating binary random variables with static state is with a binary Bayesian filter using the log-odds ratio formulation. The main idea is that instead of multiplying terms from prior and current time steps, the properties of logarithms allow these terms to be simply added, while avoiding truncation issues associated with probabilities close to 0 or 1 [4]. The log-odds ratio is also popular because learning techniques with forward models [10, 16] or ad hoc techniques [5, 6] become simplified.

However, the log-odds ratio formulation makes an assumption that is not consistent with the occupancy grid mapping problem. It is assumed that if X_t and $\mathbf{m}_i$ are known, then Z_t is independent of $X_{1:t-1}$ and $Z_{1:t-1}$, i.e.,

$$P(Z_t|\mathbf{m}_i, X_{1:t}, Z_{1:t-1}) \approx P(Z_t|\mathbf{m}_i, X_t). \quad (6)$$

In other words, the past history of measurements and poses are irrelevant to the current measurement. However, the prior measurements and poses provide a certain information to the occupancy map, which affects the probabilistic model of the current measurement.

More specifically, forward models are based on the entire map m. Since (6) conditioned only by the i-th cell, past poses $X_{1:t-1}$ and past measurement scans $Z_{1:t-1}$ may contain important information about the occupancy of the other cells, which should be considered in calculating the probability distribution of the measurement scan Z_t. For example, $X_{1:t-1}$ and $Z_{1:t-1}$ might suggest that the i-th cell is occluded, so the approximation of Eq. 6 would not be justified. In short, valuable information about the map that can be obtained by the past measurements is ignored in the

common log-odds ratio formulation. The probabilistic mapping proposed in this paper avoids this assumption of the log-odds ratio framework to construct a more accurate map by utilizing all of the past and the current measurements.

3.2 Inverse Sensor Model

In this section, we propose an algorithm to compute the exact inverse sensor model efficiently. Suppose the probability of the map conditioned on the past poses and measurements, namely $P(m|X_{1:t-1}, Z_{1:t-1})$, is known. Here we construct a posteriori probability $P(m|z_{t,l}, X_{1:t}, Z_{1:t-1})$, based on the current pose X_t, the measurement from the l-th ray $z_{t,l}$, and the given forward sensor model $p(z_{t_l}|m, X_t)$.

Bayesian Framework The occupancy probability $P(m|z_{t,l}, X_{1:t}, Z_{1:t-1})$ is based on the forward sensor model $p(z_{t,l}|m, X_{1:t}, Z_{1:t-1})$ that describe the distribution of the measurements for the given robot pose and the map. This specifies the stochastic characteristics of the sensor, such as the mean squared error, and it is assumed as a known distribution, specific to a particular depth sensor. One could certainly obtain a stochastic model by fitting a probability density to a controlled set of measurements. Bayes' rule yields

$$P(m|z_{t,l}, X_{1:t}, Z_{1:t-1}) = \frac{p(z_{t,l}|m, X_{1:t}, Z_{1:t-1})P(m|X_{1:t-1}, Z_{1:t-1})}{p(z_{t,l}|X_{1:t}, Z_{1:t-1})}, \tag{7}$$

considering that X_t carries no information about m without Z_t. If the current pose X_t and map m are known, then the measurement ray $z_{t,l}$ is independent of past poses $X_{1:t-1}$ and past measurements $Z_{1:t-1}$ to obtain

$$P(m|z_{t,l}, X_{1:t}, Z_{1:t-1}) = \eta_{t,l}\, p(z_{t,l}|m, X_t) P(m|X_{1:t-1}, Z_{1:t-1}), \tag{8}$$

where the normalizing constant $\eta_{t,l} \in \mathbb{R}$ absorbs all terms that do not depend on the map m.

Next, we compute the occupancy probability of each cell. Let $\mathcal{M}_i$ be the set of maps where the i-th cell is occupied, i.e., $\mathcal{M}_i = \{m \in \{0,1\}^{n_m} \mid \mathbf{m}_i = 1\}$. To compute the probability of occupancy of the i-th cell, all possible combinations of map in $\mathcal{M}_i$ should be considered, i.e.,

$$P(\mathbf{m}_i|z_{t,l}, X_{1:t}, Z_{1:t-1}) = \eta_{t,l} \sum_{m \in \mathcal{M}_i} p(z_{t,l}|m, X_t) P(m|X_{1:t-1}, Z_{1:t-1}). \tag{9}$$

Furthermore, to determine the normalizing constant $\eta_{t,l}$, the complement $P(\bar{\mathbf{m}}_i|z_{t,l}, X_{1:t}, Z_{1:t-1})$ must be calculated in the similar manner. Since there are 2^{n_m-1} maps in $\mathcal{M}_i$, these equations require 2^{n_m} terms to compute the summation for the i-th grid cell and normalizer, and this process should be repeated for all other cells along the measurement ray. This is the main reason why the existing results are based on certain approximations of the above expression.

Computationally Efficient Approach We propose a computational algorithm to evaluate (9) efficiently. Since the cells outside of the sensor field of view (FOV) are not affected, we focus on a reduced map r_l in the FOV of the l-th ray. This reduced map is chosen such that each cell of r_l corresponds to a grid cell of map m that the l-th ray intersects, ordered by increasing distance, which is easily determined from geometry. Let $\mathbf{r}_{l,k}$ be the binary random variable representing the occupancy of the k-th cell of the l-th ray. The number of cells in the reduced map is denoted by $n_{r,l} \leq n_m$.

Next, let $\mathbf{r}_{l,k+}$ correspond to the event that the k-th cell of the l-th ray is occupied, cells with lower index (closer cells) are free, and cells with greater index (farther cells) may or may not be occupied, i.e., event $\mathbf{r}_{l,k+}$ occurs when $r \in \left\{r \in \{0,1\}^{n_{r,l}} \,|\mathbf{r}_{l,1} = 0, \mathbf{r}_{l,2} = 0, \ldots, \mathbf{r}_{l,k-1} = 0, \mathbf{r}_{l,k} = 1\right\}$. In other words, the k-th cell $\mathbf{r}_{l,k}$ is the closest occupied cell to current pose X_t along the l-th ray. For example, see Fig. 1 that illustrates the event of $\mathbf{r}_{l,k+}$ when $l = 1$ and $k = 4$ for an one-dimensional cell array. This concept of grouping map outcomes can be easily extended to 2D using ray casting, where a 1D ray is spanning 2D space. The cells considered for the measurement ray are determined with the geometric intersection between the ray and map cell edges. Then, the forward sensor model is identical for all maps defined by $\mathbf{r}_{l,k+}$, regardless of the occupancy of the cells beyond the k-th cell, and the corresponding forward sensor model $p(z_{t,l}|\mathbf{r}_{l,k+}, X_t)$ depends on the distance from X_t to the k-th cell. Based on this, we present the mathematical expressions for the exact inverse sensor model, as summarized by Eq. 10 in Proposition 1. Later in Section 3.3, these are also rearranged as a computational algorithm.

Proposition 1 *For the l-th measurement ray, the a posteriori probability of the occupancy of the k-th cell, namely the ray inverse sensor model, is given by*

$$P(\mathbf{r}_{l,k}|z_{t,l}, X_{1:t}, Z_{1:t-1}) = \eta_{t,l} \tilde{P}(\mathbf{r}_{l,k}|z_{t,l}, X_{1:t}, Z_{1:t-1}), \tag{10}$$

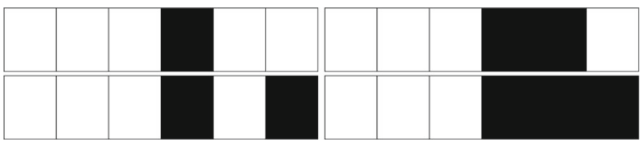


Fig. 1 A robot located to the left of a 1D occupancy grid map r_l composed of $n_{r,l} = 6$ grid cells in 4 cases. In each case, cells 1–3 are free, cell 4 is occupied, and cells 5 and 6 may or may not be occupied. All above outcomes correspond to the event $\mathbf{r}_{l,4+}$

where the unnormalized probability of the inverse sensor model is defined as

$$\begin{aligned}\tilde{P}(\mathbf{r}_{l,k}|z_{t,l}, X_{1:t}, Z_{1:t-1}) &= P(\mathbf{r}_{l,k}|X_{1:t-1}, Z_{1:t-1}) \\ &\times \left[\sum_{i=1}^{k-1}\left\{\prod_{j=0}^{i-1} P(\bar{\mathbf{r}}_{l,j}|X_{1:t-1}, Z_{1:t-1})\right\} p(z_{t,l}|\mathbf{r}_{l,i+}, X_t) P \right. \\ &\left. \times (\mathbf{r}_{l,i}|X_{1:t-1}, Z_{1:t-1})\right] \\ &+ \left\{\prod_{j=0}^{k-1} P(\bar{\mathbf{r}}_{l,j}|X_{1:t-1}, Z_{1:t-1})\right\} p(z_{t,l}|\mathbf{r}_{l,k+}, X_t) P \\ &\times (\mathbf{r}_{l,k}|X_{1:t-1}, Z_{1:t-1}), \end{aligned} \tag{11}$$

where $P(\bar{\mathbf{r}}_{l,0}|X_{1:t-1}, Z_{1:t-1}) = P(\mathbf{r}_{l,n_r+1}|X_{1:t-1}, Z_{1:t-1}) = 1$ *is chosen for convenience and* $p(z_{t,l}|\mathbf{r}_{l,(n_r+1)+}, X_t)$ *represents the probability density of the measurement when all of cells in the field of view of the* l*-th ray is not occupied. The normalizer* $\eta_{t,l}$ *is given by*

$$\begin{aligned}\eta_{t,l} = &\left[\sum_{i=1}^{n_{r,l}+1}\left\{\prod_{j=0}^{i-1} P(\bar{\mathbf{r}}_{l,j}|X_{1:t-1}, Z_{1:t-1})\right\}\right. \\ &\left. \times p(z_{t,l}|\mathbf{r}_{l,i+}, X_t) P(\mathbf{r}_{l,i}|X_{1:t-1}, Z_{1:t-1})\right]^{-1}, \end{aligned} \tag{12}$$

and it is independent of the cell index k.

Proof See Appendix. □

Note that the a priori estimate, $P(\mathbf{r}_{l,k}|X_{1:t-1}, Z_{1:t-1})$ and its compliment $P(\bar{\mathbf{r}}_{l,k}|X_{1:t-1}, Z_{1:t-1}) = 1 - P(\mathbf{r}_{l,k}|X_{1:t-1}, Z_{1:t-1})$ are available at the t-th step. Then, Eq. 10 yields a sequential occupancy grid mapping that can be applied whenever new measurements are available.

Compared with Eq. 9, where the terms of the summation should be repeated $2^{n_{r,l}}$ times *per each cell of the reduced map*, the proposed expressions (10) and (12) are *substantially* simpler. In fact, if Eqs. 11 and 12 are obtained recursively, requiring the summation of $O(n_{r,l}+1)$ rather than $O(n_{r,l} \times 2^{n_{r,l}})$ as previously thought for *all cells of the reduced map*, this yields an algorithm that is $\mathbf{n_{r,l} \times 2^{n_{r,l}}/(n_{r,l}+1)}$ **times faster**.

Scan Inverse Sensor Model The above one-dimensional ray inverse sensor model may be generalized into a two-dimensional scan inverse sensor model by updating the map sequentially for each ray, according to

$$\begin{aligned}&P(\mathbf{m}_i|X_{1:t}, Z_{1:t}) \\ &= P((\ldots(((\mathbf{m}_i|X_{1:t}, Z_{1:t-1})|z_{t,1})|z_{t,2})|\ldots)|z_{t,n_z}). \end{aligned} \tag{13}$$

In short, this approach updates the map occupancies along the first measurement ray, then repeats this process for each subsequent ray until all rays composing the measurement scan have updated the map.

3.3 Summary of Algorithm

In short, the proposed scan inverse sensor model is computed by Eqs. 10–13. These are summarized in Algorithm 1 as a computationally efficient, recursive algorithm that avoids several repeated calculations of the same quantity. This algorithm utilizes temporary variables, where the required computational resources are minimized by computing the updated occupancy probabilities of all grid cells along a measurement ray together. Note that this function is only considering l-th ray at the t-th time step, where probabilities are subject to conditions on $X_{1:t}$, $Z_{1:t-1}$, $z_{t,1:l-1}$, so these are removed from the algorithm for simplicity.

Algorithm 1 Inverse Sensor Model of an Individual Measurement Ray

```
Function: P(r_k|z) = InverseSensorModel(P(r_k), z) ∀ k ∈ {1, 2, ..., n_r};
Initialize η^{-1} = 0 and P(r̄_0) = 1;
for k = 1, 2, ..., n_r do
    P(r_{k+}) = P(r̄_{0:k-1}) P(r_k);
    P(r̄_{0:k}) = P(r̄_{0:k-1})(1 − P(r_k));
    a_temp = P(r_{k+}) p(z_l|r_{k+}, X);
    P̃(r_k|z) = P(r_k) η_{t,l}^{-1} + a_temp;
    η^{-1} = η^{-1} + a_temp;
η^{-1} = η^{-1} + P(r̄_{0:k}) p(z_max);
Return P(r_k|z) = η P̃(r_k|z) for all k ∈ {1, 2, ..., n_r};
```

In contrast to the current approximate inverse sensor models, the proposed algorithm evaluates the exact inverse sensor model (9) efficiently without relying on assumptions like (6). Therefore, it yields substantially more accurate maps for the same set of measurements. These are illustrated by numerical examples and experiments in Sections 6 and 7.

This proposed mapping approach follows a Bayesian framework that focusses on occupied and free space, not tracking particular systems or mapping features, such as visual servoing [17]. The proposed algorithm may be combined with feature-based approaches for object tracking or stabilization purposes, with the added challenge of associating occupied spaces with features.

4 Computation of Expected Information Gain

In this section, we present a method to determine the expected entropy from an individual measurement ray from

a known pose. Based on this, an autonomous exploration algorithm will be developed in the next section.

4.1 Single Ray Expected Value of Entropy

Suppose there is a probabilistic map constructed by past measurements, namely $P(m|X_{1:t}, Z_{1:t})$. We wish to compute the expected entropy when the next measurement is taken at the candidate pose $X_c = (x_c, R_c)$, consisting of location $x_c \in \mathbb{R}^2$ and attitude $R_c \in \mathsf{S}^1$. Assume that the direction of the l-th ray is determined by the attitude R_c. The expected entropy due to the l-th ray $z_{c,l}$ at position x_c is

$$\begin{aligned}\mathrm{E}[H(P(m|X_{1:t}, Z_{1:t}, x_c, z_{c,l}))] \\ = \int_{z_{\min}}^{z_{\max}} H(P(m|X_{1:t}, Z_{1:t}, x_c, z_{c,l})) \\ p(z_{c,l}|X_{1:t}, Z_{1:t}, x_c) dz_{c,l}. \end{aligned} \tag{14}$$

We discretize the measurement ray space such that $z_{c,l}$ falls on points along the measurement ray intersecting with grid cell edges. The discretized expected value of (14) is

$$\begin{aligned}\mathrm{E}[H(P(m|X_{1:t}, Z_{1:t}, x_c, z_{c,l}))] \\ = \sum_{k=1}^{n_{r,l}+1} \Big\{ H(P(m|X_{1:t}, Z_{1:t}, x_c, z_{c,l,k})) \\ P(z_{c,l,k}|X_{1:t}, Z_{1:t}, x_c) \Big\}, \end{aligned} \tag{15}$$

where index $z_{c,l,k}$ is the distance from position 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 15 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

$$P(z_{c,l,k}|X_{1:t}, Z_{1:t}, x_c) = \frac{p(z_{c,l,k}|X_{1:t}, Z_{1:t}, x_c)}{\sum_{i=1}^{n_{r,l}+1} p(z_{c,l,i}|X_{1:t}, Z_{1:t}, x_c)}. \tag{16}$$

Conveniently, this density corresponds to the inverse normalizer according to 7,

$$p(z_{c,l,k}|X_{1:t}, X_c, Z_{1:t}) = \eta_{c,l,k}^{-1}, \tag{17}$$

where $\eta_{c,l,k}^{-1}$ is defined in 12. Updating variable subscripts, the normalizer becomes

$$\begin{aligned}\eta_{c,l,k}^{-1} = \sum_{i=1}^{n_{r,l}+1} \left\{ \prod_{j=0}^{i-1} P(\bar{\mathbf{r}}_{l,j}|X_{1:t}, Z_{1:t}) \right\} \\ p(z_{c,l,k}|\mathbf{r}_{l,i+}, X_c) P(\mathbf{r}_{l,i}|X_{1:t}, Z_{1:t}). \end{aligned} \tag{18}$$

By substituting 16 and 17 into 15, the expected entropy at X_c from individual measurement ray $z_{c,l}$ is

$$\begin{aligned}\mathrm{E}[H(P(m|X_{1:t}, Z_{1:t}, x_c, z_{c,l}))] \\ = \left(\sum_{i=1}^{n_{r,l}+1} \eta_{c,l,i}^{-1} \right)^{-1} \sum_{k=1}^{n_{r,l}+1} \Big\{ H(P(m|X_{1:t}, Z_{1:t}, x_c, \\ z_{c,l,k})) \eta_{c,l,k}^{-1} \Big\}, \end{aligned} \tag{19}$$

where $\eta_{c,l,k}^{-1}$ is taken from Eq. 18. Here, Eq. 19 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.

4.2 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 Eq. 18 are embedded in Eq. 19. However, several 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 18 substantially. This goal is achieved by systematically selecting a smaller set of grid cells to consider over the summations of Eqs. 18 and 19. 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

$$\begin{aligned}P(\mathbf{r}_{l,k+}|X_{1:t}, Z_{1:t}) \\ = \left\{ \prod_{j=0}^{k-1} P(\bar{\mathbf{r}}_{l,j}|X_{1:t}, Z_{1:t}) \right\} P(\mathbf{r}_{l,k}|X_{1:t}, Z_{1:t}), \end{aligned} \tag{20}$$

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 of Eqs. 20 (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 Eqs. 18 and 19, 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. It can be noted that if $\hat{n} = n_{r,l}+1$, the ray objective function is computed without approximation.

4.3 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 (Algorithm 2). Much like the algorithm pseudocode for the inverse sensor model (Algorithm 1), the variable a_{temp} serves as an intermediate variable designed to avoid repeated calculations. Since this algorithm operates as a function, fixed indices and condition variables are removed for simplification.

Algorithm 2 Expected Information Gain from a Measurement Ray

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

4.4 Numerical Justification for Approximation

The purpose of this numerical example is to provide evidence that the approximations are reasonable and increase the algorithm speed 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 on the Microsoft Kinect [9, 18] with maximum reading depth $z_{\max} = 4$m (20 grid cells inside the sensor FOV). The goal is to compare the expected entropy $\text{E}[H(P(m|X_{1:t}, Z_{1:t}, x_c, z_{c,l}))]$ from Eq. 19 and with an approximation $\text{E}[H_{\text{approx}}(P(m|X_{1:t}, Z_{1:t}, x_c, z_{c,l}))]$, which only considers $\hat{n}$ grid cells with highest detection probability (20).

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 a 20% chance of receiving an a priori probability uniformly distributed between 0 and 1. Several metrics serve to evaluate $\text{E}[H(P(m|X_{1:t}, Z_{1:t}, x_c, z_{c,l}))]$ with $\text{E}[H_{\text{approx}}(P(m|X_{1:t}, Z_{1:t}, x_c, z_{c,l}))]$. The median expected entropy change is $\text{E}[H(P(m|X_{1:t}, Z_{1:t}, x_c, z_{c,l}))] - H(P(m|X_{1:t}, Z_{1:t})) = -0.83792$. The error for the 100 Monte Carlo cases is defined simply as

$$e_H = \frac{1}{100}\sum_{i=1}^{100} \text{abs}\Big(\text{E}[H(P(m|X_{1:t}, Z_{1:t}, x_c, z_{c,l}))] - \text{E}[H_{\text{approx}}(P(m|X_{1:t}, Z_{1:t}, x_c, z_{c,l}))]\Big). \quad (21)$$

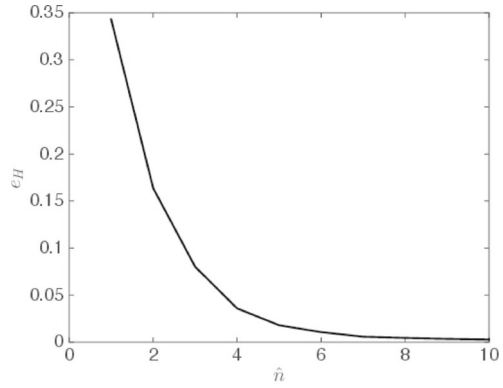
The Monte Carlo trials are repeated for $\hat{n} = \{1, 2, \ldots, 10\}$ and the results are plotted in Fig. 2. This example shows a typical case when the summation limits generated from Eq. 20 have only small effects on Eq. 19, while providing very large improvements in reducing computation.

5 Autonomous Exploration

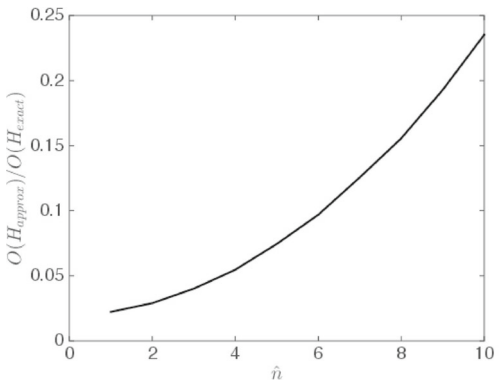
In this section, we develop an autonomous exploration scheme inside a 2D environment based on choosing movements that minimize map entropy and avoid collisions.

5.1 Pose Selection Optimization

The objective is to choose a future pose that minimizes the map uncertainty, such that the robot can autonomously move toward this optimal goal. The robot pose is selected among various positions and attitudes. Since searching over all locations and attitudes would require infinite computational resources, these search spaces are discretized into a limited number of robot attitudes and positions. The procedure involves three steps. First, the attitude that maximizes the information gain objective function is selected at each candidate future pose location. Second, the location with the maximum objective function of all candidates is selected. Thus, the optimal attitude and the optimal location compose the optimal future pose. Finally, the robot determines and follows a collision-free path to the optimal



(a) Entropy Error



(b) Computation Ratio

Fig. 2 The entropy error is decreased at the cost of increasing computation time in this Monte Carlo 1D measurement ray expected entropy case

future pose. This process is repeated, resulting in autonomous exploration.

Attitude Optimization Consider a collision-free pose candidate at an arbitrary location x_c. At this pose, consider n_d evenly-spaced measurement rays oriented radially outward from the robot in a circular pattern (see red lines in Fig. 3). These rays are denoted $\{z_{c,1}, z_{c,2}, \ldots, z_{c,n_d}\}$. Attitudes correspond to the rays such that the robot sensor is aligned with the ray direction, and these are denoted $\{R_{c,1}, R_{c,2}, \ldots, R_{c,n_d}\}$, where the scan with attitude $R_{c,d}$ might cover several ray directions depending on the sensor field of view (FOV). We choose optimal attitude R_c^* as the summation of the expected entropy changes covered by the scan,

$$R_c^* = \underset{R_{c,d}}{\operatorname{argmax}} \sum_{z_{c,i} \in R_{c,d}\ \mathrm{FOV}} \Big(H(P(m|X_{1:t}, Z_{1:t})) - \mathrm{E}[H(P(m|X_{1:t}, Z_{1:t}, x_c, z_{c,i}))] \Big). \quad (22)$$

This method provides the attitude that maximizes the information gain at an arbitrary location in 2D space. The set

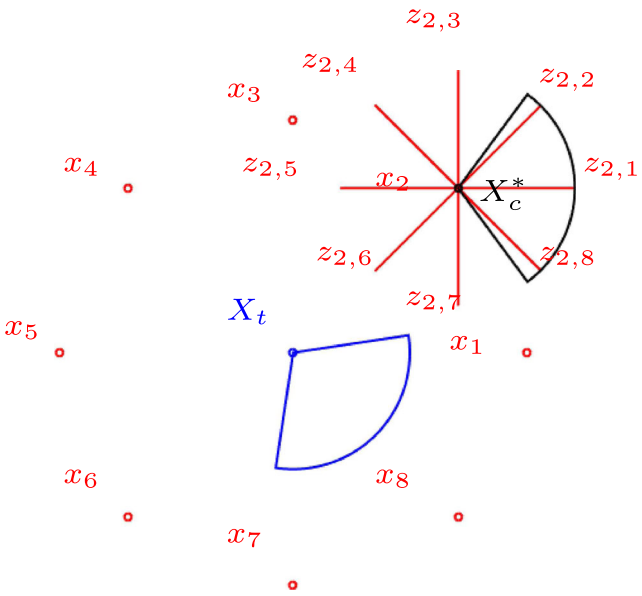


Fig. 3 This figure illustrates the expanding ring method. Initially, a robot (blue circle) views down and left (blue sector). Then, the robot considers $n_c = 8$ poses (red circles) as possible candidates for where to move next. For each candidate, $n_d = 8$ directions are considered (red lines, only 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

of candidate positions that warrant consideration is determined with one of two methods, namely *expanding ring* and *complete Cartesian*, described next.

Expanding Ring The expanding ring technique is advantageous for searching local solutions quickly, only considering distant future poses when necessary. The key idea is that future candidate locations lie on a circular "ring" centered around the robot, evenly spaced around the ring (see red circles in Fig. 3), and the ring is expanded if certain criteria are not met. More explicitly, consider n_c candidate locations denoted by $x_c \in \{x_1, x_2, \ldots, x_{n_c}\}$ located with the distance δ away from the robot. Any location that violates inequality constraint (5) is excluded to avoid collisions. The set of optimal attitudes at each candidate location is $\{R_1^*, R_2^*, \ldots, R_{n_c}^*\}$, obtained from Eq. 22. The objective function (3) is computed by a summation about the n_d rays as

$$\mathcal{I}(x_c, R_c^*) \approx \sum_{z_{c,i} \in R_c^*\ \mathrm{FOV}} \Big(H(P(m|X_{1:t}, Z_{1:t})) - \mathrm{E}\left[H(P(m|X_{1:t}, Z_{1:t}, x_c, R_c^*, z_{c,i}))\right] \Big), \quad (23)$$

$$x_c^* = \underset{x_c}{\operatorname{argmax}}\, \mathcal{I}(x_c, R_c^*). \quad (24)$$

At the optimal pose $X_c^* = (x_c^*, R_c^*)$, the resulting information gain must satisfy $\mathcal{I}(X_c^*) \geq \mathcal{I}_{\min}$, where $\mathcal{I}_{\min}$ is some minimum threshold for expected information gain; robot motion is only justified when the expected information gain is significantly large. If the minimum threshold is not met, the ring of candidate pose locations is increased such that n_c and δ are multiplied by a scale function $\lambda > 1$. This process is repeated until the expected information gain of the optimal pose is at least $\mathcal{I}_{\min}$, or all candidates lie in collision zones or outside map limits.

The primary advantage of the expanding ring approach to search the 2D space is that the number of pose candidates need not be proportional to the map area, and that local maxima tend to fall within short distances of the current robot pose. The main drawback is that poses outside a small local region of the robot are frequently neglected, and the robot frequently executes repeated short motions, often without large information gains.

Complete Cartesian The complete Cartesian method to search the 2D space provides candidates located fixed distance d apart in each Cartesian direction. This approach is advantageous for capturing expected information gains in regions outside of the immediate vicinity of the robot, but must be applied carefully to avoid issues with computational bottlenecks due to the potentially-large search space.

Similar to the expanding ring approach, n_c candidate pose locations are considered, where those violating (5) are neglected from further consideration. However, unlike the ring expanding approach, the candidates have different distances from the robot in general, so the objective function is modified to enforce a cost on the squared distance from the current pose location x_t to the candidate location x_c, i.e.,

$$\begin{aligned}\mathcal{I}(x_c, R_c^*, x_t) \approx \sum_{z_{c,i} \in R_c^* \text{ FOV}} \Big(& H(P(m|X_{1:t}, Z_{1:t})) \\ & - \mathrm{E}\left[H(P(m|X_{1:t}, Z_{1:t}, x_c, R_c^*(x_c), z_{c,i}))\right] \Big) \\ & - k_{\text{dist}}(x_t - x_c)^{\mathsf{T}}(x_t - x_c), \end{aligned} \tag{25}$$

where weighting function k_{dist} represents the sensitivity to avoiding large motions across the map. Furthermore, the same candidate locations are considered throughout the repeated processes of exploration, and the map cell occupancies are assumed static. Therefore, if candidate location x_c fails to satisfy $\mathcal{I}(x_c, R_c^*, x_t) \geq \mathcal{I}_{\min}$, then x_c need *never* be considered again. Avoiding these unnecessary calculations yields a greatly lowered computational burden, particularly late in exploration where several regions are collision-free but disadvantageous to revisit.

Motion Planning Once the optimal future pose X_c^* is selected, the robot must traverse its environment to reach X_c^* without collisions. The locations of cells on a grid-based map provide nodes for Dijkstra's algorithm [19, 20]. At each node, Eq. 5 must be satisfied for the node to be considered reachable, and edges between nodes correspond to neighboring grid cells. Dijkstra's algorithm provides a relatively simple motion planning strategy to move from x_t to x_c^* that avoids collisions and local minima.

The trajectory to follow the path outlined by Dijkstra's algorithm is determined with a linear constrained least squares optimization, assuming the robot follows a polynomial trajectory with fixed speed. The starting and ending positions and attitudes may be constrained, and polynomials patched together for long trajectories share a common position and velocity with respect to time. Then, the robot tracks this trajectory until the robot falls within acceptable thresholds of the final optimal pose. For generality of the proposed exploration algorithm, any position controller yielding robotic motion that follows this desired trajectory may be selected. Once the robot completes this motion, the entire process is repeated.

6 Numerical Examples

The efficacy of the proposed algorithms are verified by numerical examples. First, the occupancy grid mapping strategies are simulated in Section 6.1, followed by numerical examples for autonomous exploration in Section 6.2.

6.1 Occupancy Grid Mapping

First, we compare the proposed exact solution to the inverse sensor model summarized in Algorithm 1 with an approximate algorithm presented in [9, 18], summarized as follows. The probability that the i-th grid cell $\mathbf{m}_i$ is occupied conditioned on the measurement ray $z_{t,l}$ at the pose X_t is the continuous function

$$P(\mathbf{m}_i|z_{t,l}, X_t) = \begin{cases} 0.3 + \left(\frac{k}{\sigma\sqrt{2\pi}} + 0.2\right) e^{-\frac{1}{2}\left(\frac{\hat{z}_{l,i} - z_{t,l}}{\sigma}\right)^2} & \text{if } \hat{z}_{l,i} \leq z_{t,l}, \\ 0.5 + \frac{k}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\hat{z}_{l,i} - z_{t,l}}{\sigma}\right)^2} & \text{otherwise,} \end{cases}$$

which is based on the expected distance to the cell $\hat{z}_{l,i}$ with parameters $k = \sigma = 0.6$. This follows the structure of the approximate inverse sensor model proposed in [8]. The main idea of this approach is that the probability of a cell being occupied (i) near a measurement is high (measurement likely hits this cell), (ii) between the robot and the measurement is low (measurement passes through this cell), and (iii) beyond the measurement is unchanged (the robot cannot measure through a wall/object). Then, these probabilities are combined in a weighted fashion such that all

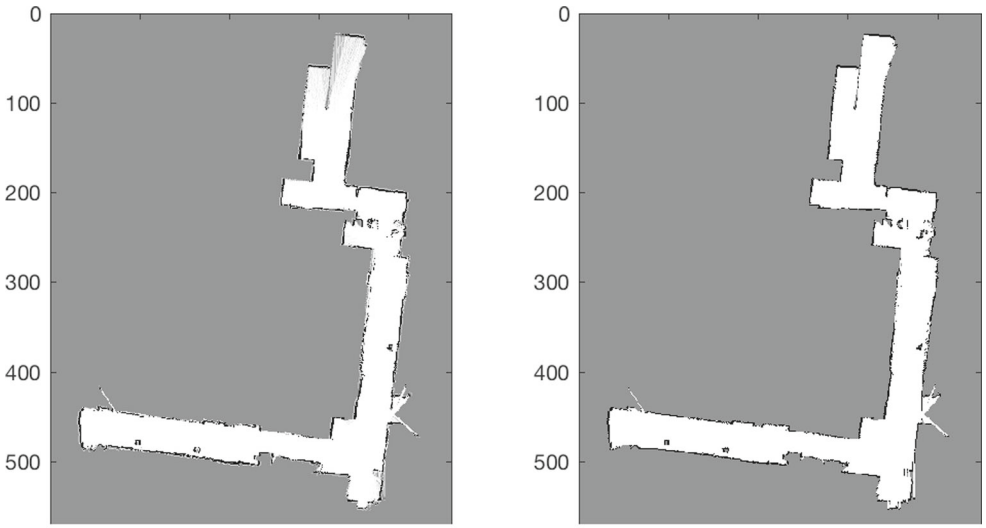
Table 1 Experimental parameters provided from dataset

Parameters	Value
Map dimension	576x448 [pixel]
Resolution	1/16 [m]
Scan angle	[-2.36, 2.36]
Ray number	1081
Total frame	Every 120 scans out of 3701

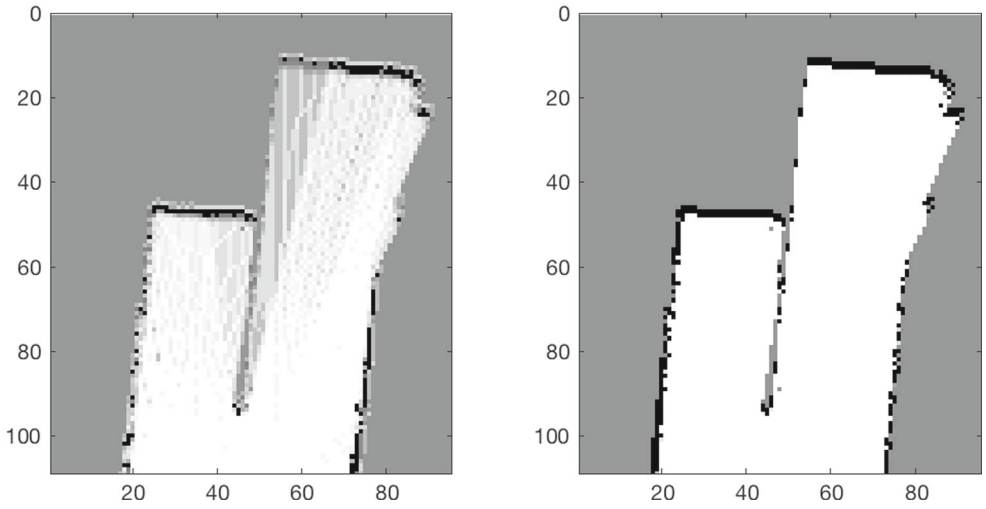
measurements rays of scan Z_t simultaneously update the same grid cell in a log-odds format,

$$\log\left(\frac{P(\mathbf{m}_i|Z_t, X_t)}{1-P(\mathbf{m}_i|Z_t, X_t)}\right) = \frac{1}{\sum_{z_{t,l}\in\mathbf{m}_i}\hat{z}_{l,i}}\sum_{z_{t,l}\in\mathbf{m}_i}$$
$$\log\left(\frac{P(\mathbf{m}_i|z_{t,l}, X_t)}{1-P(\mathbf{m}_i|z_{t,l}, X_t)}\hat{z}_{l,i}\right).$$

Table 1 shows the mapping parameters and data specifications for the actual odometry and lidar measurements from University of Pennsylvania through Coursera open course [21]. Among these parameters, cell resolution and the number of rays have the greatest impact on computation. In general, the finer the grid, the greater the memory and computational requirements are because the computational order of Eq. 10 grows linearly with the number of grid cells inside the sensor range limits. Since the inverse sensor models are combined sequentially as shown in Eq. 13, the number of rays considered are proportional to the computation order as well. Figure 4 shows the direct comparison of the resulting map based on the approximate and proposed inverse sensor model. As shown in the figure, both approaches capture the structure of the environment. However, the significant differences can be seen in the certain

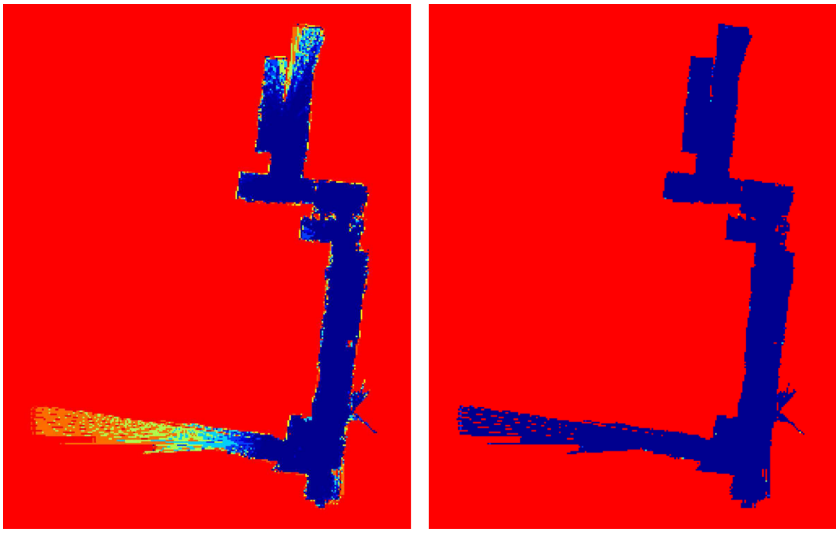


(a) Comparison of the total mapped environment based on the approximate (left) and proposed (right) inverse sensor model.



(b) Evidence of significant improvement at certain area of the map.

Fig. 4 Using experimental sensor data, the map was generated using an approximate and the proposed exact inverse sensor models. The laser sensor and odometry data used for the experiment was provided by University of Pennsylvania open course robotics estimation and learning on Coursera [21]



(a) Approximate inverse sensor model

(b) Exact inverse sensor model

Fig. 5 The entropy is compared between the approximate and proposed exact inverse sensor models at step 2280. The blue areas show the lowest entropy while red shows highest

area of the map such as in Fig. 4. The differences can be clearly depicted in Fig. 5 where entropy of maps are compared at frame 2280. The red part of the map shows the maximum entropy whereas blue shows lowest entropy for the cell. Finally, overall improvement was demonstrated in Fig. 6 resulting improved entropy value per grid cell throughout the mapping process. In short, the proposed exact occupancy grid mapping produced a cleaner map with lower entropy from the same set of measurements.

6.2 Autonomous Exploration

Next, the proposed exploration approach is applied to the floor plan of Intel Research Lab, illustrated in Fig. 7. The robot explores its surroundings with an occupancy grid with 90,000 cells where grid cell edges are $\alpha = 0.1$m, composing a map with dimensions 30m $\times$ 30m. 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 0.3m and

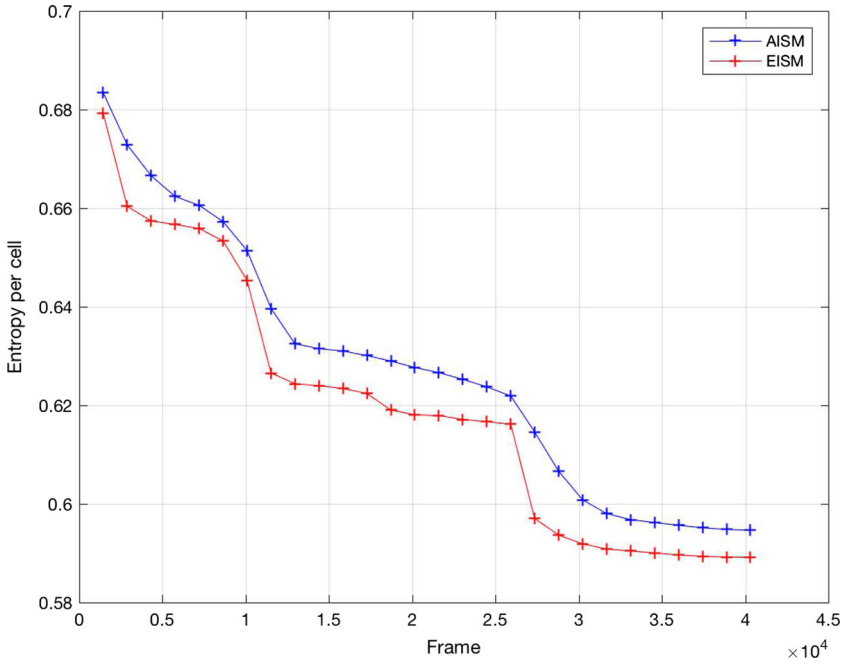


Fig. 6 The map entropies of the approximate inverse sensor model (blue) and the proposed exact inverse sensor model (red) show that the exact version yields a more certain occupancy grid map

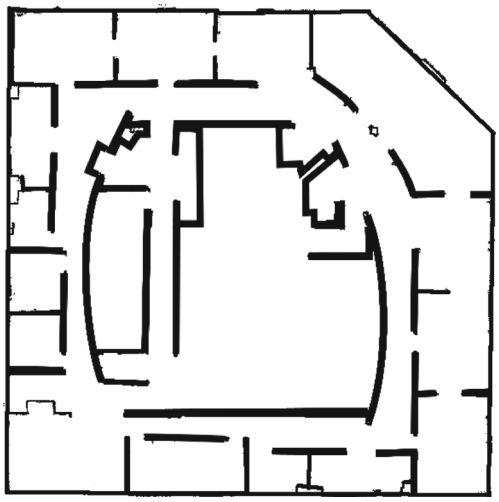


Fig. 7 Modified Intel Research Lab floor plan form the SLAM benchmark [22]

$P(\mathbf{m}_i) = 0.5$ for all other cells. For added safety, cells falling within 0.6m of the robot are considered inside the possible collision zone $\mathcal{C}_X$, from Eq. 5, where $\beta = 0.1$ is selected.

The robot follows the complete Cartesian candidate search method to determine future poses, with $k_{\text{dist}} = 5$ to avoid large motions with little added expected information gains. Due to the large size of the map, candidates below the expected information gain of 1.25 are neglected from future consideration.

The time sequence of the exploration for every 5 minutes are simulated in ROS Stage 2D environment. The corresponding simulation results are illustrated in Fig. 8 and a video is available at https://www.youtube.com/watch?v=5VdzKHreB_s. Lighter red dots and darker blue dots in the explored area correlate to the level of information gain at those positions with low and high value, respectively. It is shown that the proposed autonomous exploration algorithm successfully mapped the complicated environment composed of a number of rooms, narrow hallways, and open spaces that are irregularly shaped.

7 Experimental Results

The efficacy of the proposed mapping and autonomous exploration algorithms are also verified with experiments.

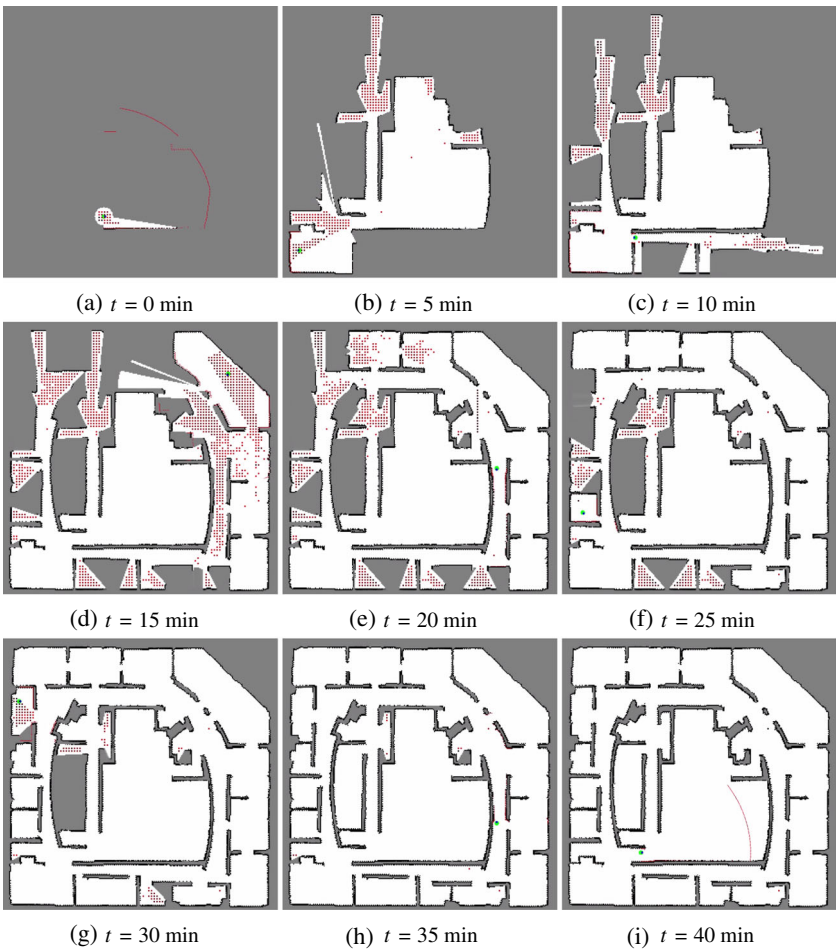


Fig. 8 A robot (green) measures a room in ROS Stage simulator using modified Intel Research Lab floor plan. The exact inverse sensor model is used for the autonomous exploration algorithm

The experiment involves a ground vehicle producing a probabilistic occupancy grid map in real time. The robot motion is governed by the information gain-maximizing policy described in this paper.

Hardware Configuration The Pioneer 3 ground robot was chosen. The vehicle accepts two inputs, namely linear velocity (aligned with the robot wheels) and angular velocity (about a central axis passing vertically through the robot). The robot pose was estimated via Vicon Tracker, which provides the location and attitude of a rigid body. The depth readings were obtained by a Kinect depth sensor. Since the experiment was to map and explore a 2D environment, only a single central row of the 3D Kinect depth scan provided the necessary measurements. An environment with features of walls and obstacles was constructed by Styrofoam.

Software Configuration The proposed algorithm was implemented with ROS. The ROS framework provided seamless communication betweens sensors and actuators, and the executable programs (referred to as "nodes") operate quickly when compiled from C++ code. The robot and map are plotted using "OpenGL" libraries.

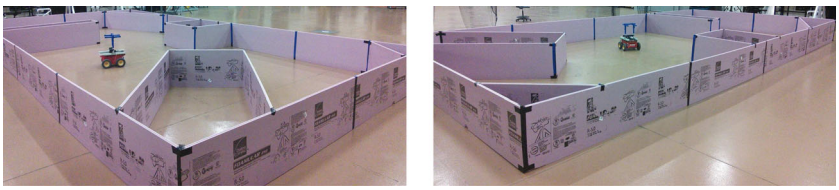
Synchronization between the sensors is of high importance to the proposed occupancy grid mapping approach. Any time delay between the robot pose estimation and the sensor depth readings can cause conflicting information, which may be harmful to the probabilistic map. Thus, ROS approximate message filters were applied such that the pose from Vicon Tracker (100Hz) and the Microsoft Kinect (30Hz) were nearly synchronized, which provided validity of the assumption that the measurement ray positions and directions are known deterministically.

Exploring and Mapping a 2D Environment The environment consisted of Styrofoam walls on a flat floor. The parameter was oriented in a rectangle, with several obstacles and angled walls (see Fig. 9). The robot autonomously explored the space using the complete Cartesian searching approach, and Dijkstra's algorithm provided collision-free waypoints to the optimal future poses. Based on these waypoints, a constrained least squares polynomial fitting method provided a smooth trajectory to follow the waypoints with fixed velocity. Finally, a simple controller was implemented such that the Pioneer moved along the trajectory by moving and orienting toward the desired robot location 1sec in the future.

The resulting occupancy grid maps and trajectories are depicted in Fig. 10 and a video is available at https://www.youtube.com/watch?v=CRQfhhICSj0&feature=youtu.be. Most importantly, the robot builds a clear occupancy grid map despite sensor noise and imperfect sensor synchronization. The autonomous exploration successfully guides the robot such that it builds a map of the reachable space without colliding with any obstacles.

The resulting map and exploration commands are generated in real time. The grid cells are stored as double-float variables in a vector, where the vector index is mapped to a location on the occupancy grid. Thus, the memory requirements are proportional to the number of grid cells, which is roughly the mapping area divided by the area of a grid cell. Since cell locations need not be saved, memory requirements are reduced. On a Lenovo T540p laptop with an Intel Core i7-4900MQ processor (quad-core, 2.8GHz per core) and 16GB of RAM, the mean time for updating the occupancy grid is 0.0115 seconds, and the mean time to determine an exploration strategy is 0.6820 seconds. Time requirements for mapping and exploration are easily modified by changing grid cell resolution or the number of exploration pose candidates, respectively. In short, there is a tradeoff between computational speed and accuracy of mapping or exploration.

The autonomous exploration is governed by a policy to maximize expected entropy decreases, so entropy and entropy change with time (Fig. 11) are important metrics. In this example, the a priori occupancy probabilities of all cells are 0.1, so the entropies could temporarily rise for likely occupied cells. The total map entropy begins at roughly 2613 and decreases by roughly half to 1338. Many grid cells within the map limits are occluded by walls, preventing the blocked cells from occupancy probability changes. In some cases, the robot is able to update the cell occupancy probabilities; in other cases, cells fall inside regions occluded from all reachable space, and the robot cannot update these



(a) Bottom-left view (b) Bottom-right view

Fig. 9 Images from two perspectives show the walls and obstacles of the experimental environment

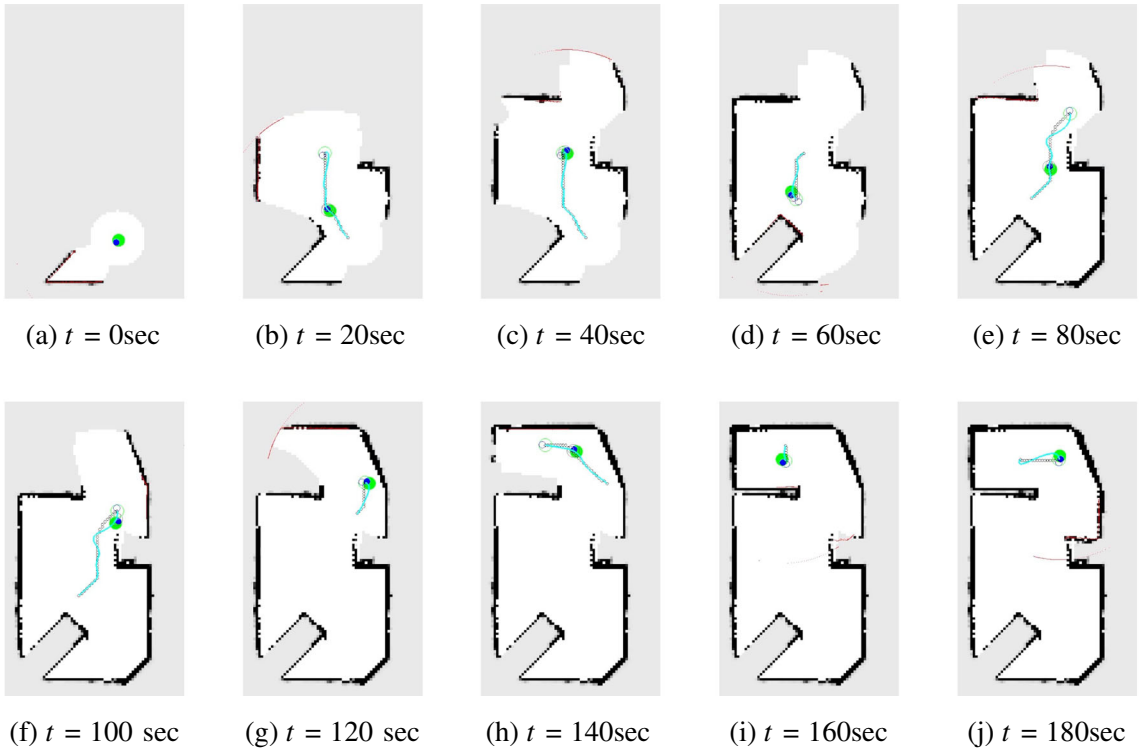


Fig. 10 A robot autonomously explores the experimental environment and produces an occupancy grid map along the way. The robot (shaded green circle body with shaded blue circle sensor) is controlled toward the desired pose 1sec in the future (unshaded gray robot), which is following a constrained least-squares trajectory (cyan curve). This trajectory is based on waypoints (black circles) from Dijkstra's algorithm to arrive at the optimal pose (unshaded green circle body with unshaded blue circle sensor) as proposed in this paper

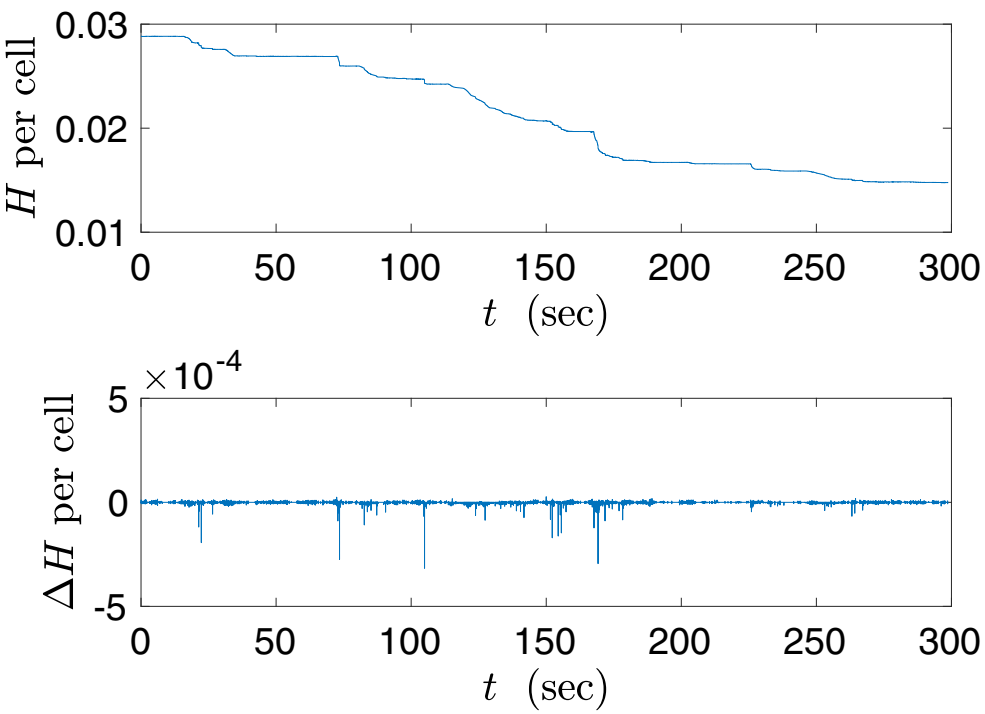


Fig. 11 The map entropy H averaged over all 90601 grid cells is reduced during experimentation. The quick changes ΔH per cell correspond to map updates when the robot captures new territory

cells. Among those grid cells visible from collision-free locations, the vast majority of cells were updated in less than 3min while following a very slow (5cm/sec) trajectory.

Both the numerical and experimental results demonstrated successful exploration strategies, though the algorithm can be improved. In particular, the robot sometimes revisits spaces that it has explored before, most frequently when the robot did not take measurements toward a particular direction. A few methods to improve this issue are treated as future work. One options is adding cost functions (e.g. distance based, number of past measurements inside an area) to prevent the robot from leaving an area before it is well-known. A second option is partitioning the space into sections, where the robot moves toward sections with large numbers of pose candidates with high expected information gains. Another option is reevaluating the future pose optimization while the robot is translating between poses because the occupancy grid map is changing, and so is expected information gain. The efficacy of these proposed changes will be evaluated, particularly as they relate to more useful systems, such as 3D map generation and multi-robot exploration.

8 Conclusions

This paper serves to improve the accuracy occupancy grid mapping and the policy governing robotic motion during autonomous exploration. With other approaches, each probability of an occupancy grid map is approximated in practice due to the perceived complications of the exact solution. This paper provides an exact solution to this complicated probability problem using cell occlusion in the forward sensor model. This yields a substantially simpler inverse sensor model, which avoids a potentially harmful Markov assumption that commonly appears in the log-odd ratio formulations. The main contribution of this paper is that all of the information available in the measurements and the a priori estimate is extracted and integrated to generate a more accurate a posteriori map. The numerical and experimental results show that maps using this exact approach are substantially improved.

The exact probabilistic occupancy grid mapping framework is utilized in autonomous exploration. The robot pose is chosen to minimize the expected map information gain, which is now directly solved more precisely in an unprecedented way. 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. Numerical examples and an experiment show how robots navigate uncertain maps governed by the proposed autonomous exploration algorithm while building accurate probabilistic maps.

Appendix

Proof of Proposition 1

Unnormalized Reduced Map Inverse Sensor Model For the reduced map of the l-th ray from Eq. 9, the unnormalized part corresponding to terms without $\eta_{t,l}$ in Eq. 10 is

$$\begin{aligned}&\tilde{P}(\mathbf{r}_{l,k}|z_{t,l}, X_{1:t}, Z_{1:t-1})\\&= \sum_{r\in\mathcal{M}_k} p(z_{t,l}|r, X_t)P(r|X_{1:t-1}, Z_{1:t-1}).\end{aligned} \tag{26}$$

Recall $\mathcal{M}_k$ corresponds to the set of maps where the k-th cell is occupied. Define a subset $\mathcal{N}_{i,k} \subset \mathcal{M}_k$ for $1 \le i \le k$ be the set of maps where the i-th cell is the first occupied cell. More explicitly,

$$\begin{aligned}\mathcal{N}_{i,k} &= \{r \in \mathcal{M}_k|\mathbf{r}_{l,i+}\} = \{r \in \mathcal{M}_k|\mathbf{r}_{l,1} = 0, \ldots, \mathbf{r}_{l,i-1}\\&\quad = 0, \mathbf{r}_{l,i} = 1, \mathbf{r}_{l,k} = 1\}.\end{aligned}$$

Then, $\mathcal{M}_k$ can be written as $\mathcal{M}_k = \bigcup_{i=1}^{k} \mathcal{N}_{i,k}$. Using this, the summation over $\mathcal{M}_k$ at 26 can be decomposed of the summation over each $\mathcal{N}_{i,k}$ to obtain

$$\tilde{P}(\mathbf{r}_{l,k}|z_{t,l}, X_{1:t}, Z_{1:t-1}) = \sum_{i=1}^{k}\left\{\sum_{r\in\mathcal{N}_{i,k}} p(z_{t,l}|r, X_t)P(r|X_{1:t-1}, Z_{1:t-1})\right\}.$$

This is motivated by the fact that the forward sensor model $p(z_{t,l}|r, X_t)$ is identical for all maps in $\mathcal{N}_{i,k}$, such that $p(z_{t,l}|r, X_t) = p(z_{t,l}|\mathbf{r}_{l,i+}, X_t)$ and it is moved left of the summation to obtain

$$\begin{aligned}&\tilde{P}(\mathbf{r}_{l,k}|z_{t,l}, X_{1:t}, Z_{1:t-1})\\&=\sum_{i=1}^{k}\left\{p(z_{t,l}|\mathbf{r}_{l,i+}, X_t)\sum_{r\in\mathcal{N}_{i,k}} P(r|X_{1:t-1}, Z_{1:t-1})\right\}.\end{aligned} \tag{27}$$

The last term of the above expression corresponds to the a priori probability of $\mathcal{N}_{i,k}$. When $i < k$, it is given by

$$\begin{aligned}\sum_{r\in\mathcal{N}_{i,k}} P(r|X_{1:t-1}, Z_{1:t-1}) = &\left\{\prod_{j=0}^{i-1} P(\bar{\mathbf{r}}_{l,j}|X_{1:t-1}, Z_{1:t-1})\right\}\\&\times P(\mathbf{r}_{l,i}|X_{1:t-1}, Z_{1:t-1})P\\&\times(\mathbf{r}_{l,k}|X_{1:t-1}, Z_{1:t-1}),\end{aligned} \tag{28}$$

and when $i = k$,

$$\sum_{r\in\mathcal{N}_{k,k}} P(r|X_{1:t-1}, Z_{1:t-1}) = \left\{\prod_{j=0}^{k-1} P(\bar{\mathbf{r}}_{l,j}|X_{1:t-1}, Z_{1:t-1})\right\} P(\mathbf{r}_{l,k}|X_{1:t-1}, Z_{1:t-1}). \quad (29)$$

Substituting (28) and (29) into Eq. 27, we obtain (11).

Complement of the Unnormalized Reduced Map Inverse Sensor Model An analytic expression for the complement of the unnormalized inverse sensor model is also required for obtaining the normalizer of the l-th ray at time t, namely $\eta_{t,l}$. Let $\bar{\mathcal{M}}_k$ be the set of maps where the k-th cell is unoccupied, i.e., $\bar{\mathcal{M}}_k = \{r \in \{0, 1\}^{n_{r,l}} \mid \mathbf{r}_{l,k} = 0\}$. Similar to 26,

$$\tilde{P}(\bar{\mathbf{r}}_{l,k}|z_{t,l}, X_{1:t}, Z_{1:t-1}) = \sum_{r\in\bar{\mathcal{M}}_k} p(z_{t,l}|r, X_t) P(r|X_{1:t-1}, Z_{1:t-1}). \quad (30)$$

Let $\bar{\mathcal{N}}_{i,k} \subset \bar{\mathcal{M}}_k$ for $1 \le i \le n_{r,l}$ and $i \ne k$ be the set of maps where the i-th cell is the first occupied cell. More explicitly,

$$\bar{\mathcal{N}}_{i,k} = \{r \in \bar{\mathcal{M}}_k \mid \mathbf{r}_{l,1} = 0, \ldots, \mathbf{r}_{l,i-1} = 0, \mathbf{r}_{l,i} = 1, \mathbf{r}_{l,k} = 0\}.$$

Then, we have $\bar{\mathcal{M}}_k = \bigcup_{\substack{i=1\\ i\ne k}}^{n_{r,l}} \bar{\mathcal{N}}_{i,k}$. Note that the forward sensor model $p(z_{t,l}|r, X_t)$ is identical for any maps in $\bar{\mathcal{N}}_{i,k}$, such that $p(z_{t,l}|r, X_t) = p(z_{t,l}|\mathbf{r}_{l,i+}, X_t)$. Similar to Eq. 27,

$$\tilde{P}(\bar{\mathbf{r}}_{l,k}|z_{t,l}, X_{1:t}, Z_{1:t-1}) = \sum_{\substack{i=1\\ i\ne k}}^{n_{r,l}} \left\{ p(z_{t,l}|\mathbf{r}_{l,i+}, X_t) \sum_{r\in\bar{\mathcal{N}}_{i,k}} P(r|X_{1:t-1}, Z_{1:t-1}) \right\}, \quad (31)$$

where the last term corresponds to the a priori probability of $\bar{\mathcal{N}}_{i,k}$. When $i < k$, it is given by

$$\sum_{r\in\bar{\mathcal{N}}_{i,k}} P(r|X_{1:t-1}, Z_{1:t-1}) = \left\{\prod_{j=0}^{i-1} P(\bar{\mathbf{r}}_{l,j}|X_{1:t-1}, Z_{1:t-1})\right\} \times P(\mathbf{r}_{l,i}|X_{1:t-1}, Z_{1:t-1}) \times P(\bar{\mathbf{r}}_{l,k}|X_{1:t-1}, Z_{1:t-1}), \quad (32)$$

and when $k < i$,

$$\sum_{r\in\bar{\mathcal{N}}_{i,k}} P(r|X_{1:t-1}, Z_{1:t-1}) = \left\{\prod_{j=0}^{i-1} P(\bar{\mathbf{r}}_{l,j}|X_{1:t-1}, Z_{1:t-1})\right\} \times P(\mathbf{r}_{l,i}|X_{1:t-1}, Z_{1:t-1}). \quad (33)$$

Substituting (32) and (33) into Eq. 31, we obtain

$$\begin{aligned}\tilde{P}(\bar{\mathbf{r}}_k|z_{t,l}, X_{1:t}, Z_{1:t-1}) &= P(\bar{\mathbf{r}}_{l,k}|X_{1:t-1}, Z_{1:t-1}) \\ &\times \left[\sum_{i=1}^{k-1} \left\{\prod_{j=0}^{i-1} P(\bar{\mathbf{r}}_{l,j}|X_{1:t-1}, Z_{1:t-1})\right\} p(z_{t,l}|\mathbf{r}_{l,i+}, X_t) \times P(\mathbf{r}_{l,i}|X_{1:t-1}, Z_{1:t-1})\right] \\ &+ \left[\sum_{i=k+1}^{n_{r,l}+1} \left\{\prod_{j=0}^{i-1} P(\bar{\mathbf{r}}_{l,j}|X_{1:t-1}, Z_{1:t-1})\right\} p(z_{t,l}|\mathbf{r}_{i+}, X_t) \times P(\mathbf{r}_{l,i}|X_{1:t-1}, Z_{1:t-1})\right], \end{aligned} \quad (34)$$

where $P(\mathbf{r}_{l,n_{r,l}+1}|X_{1:t-1}, Z_{1:t-1}) = 1$ is selected for convenience and $p(z_{t,l}|\mathbf{r}_{(n_r+1)+}, X_t)$ corresponds to the forward sensor model of maximum reading. For this special case, probability density can be easily represented with a uniform distribution over a short distance. The properties of this distribution depend on the depth sensor fidelity, because a maximum reading may indicate that all cells within the sensor range are empty, but it might also represent a failed reading.

Normalizer We have

$$P(\mathbf{r}_{l,k}|z_{t,l}, X_{1:t}, Z_{1:t-1}) + P(\bar{\mathbf{r}}_{l,k}|z_{t,l}, X_{1:t}, Z_{1:t-1}) = 1.$$

Since they share the same normalizer $\eta_{t,l}$, this implies

$$\eta_{t,l} = \frac{1}{\tilde{P}(\mathbf{r}_{l,k}|z_{t,l}, X_{1:t}, Z_{1:t-1}) + \tilde{P}(\bar{\mathbf{r}}_{l,k}|z_{t,l}, X_{1:t}, Z_{1:t-1})}.$$

Substituting 11 and 34, and rearranging, we obtain 12.

References

1. Wolf, D., Sukhatme, G.: Mobile robot simultaneous localization and mapping in dynamic environments. Auton. Robot. **1**(19), 53–65 (2005)
2. Wurm, K., Hornung, A., Bennewitz, M., Stachniss, C., Burgard, W.: Octomap: A probabilistic, flexible, and compact 3d map representation for robotic systems. In: Proceeding of the ICRA 2010 Workshop on Best Practice in 3D Perception and Modeling for Mobile Manipulation (2010)
3. Montemerlo, M., Thrun, S., Koller, D., Wegbreit, B.: Fastslam: A factored solution to the simultaneous localization and mapping problem. In: Proceeding of the National Conference on Artificial Intelligence (AAAI), Edmonton, pp. 593–598 (2002)
4. Thrun, S., Burgard, W., Fox, D.: Probabilistic robotics, ser. intelligent robotics and autonomous agents. Massachusetts Institute of Technology, Cambridge (2005)
5. Moravec, H.P., Elfes, A.: High resolution maps from wide angle sonar. In: IEEE Conference on Robotics and Automation (1985)
6. Elfes, A.: Using occupancy grids for mobile robot perception and navigation. IEEE Computer, pp. 46–57 (1989)
7. Choset, H., Lynch, K., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L., Thrun, S.: Principles of robot motion: theory, algorithms, and implementations, ser. intelligent robotics and

autonomous agents. Massachusetts Institute of Technology, Cambridge (2005)
8. Andert, F.: 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)
9. Pirker, K., Ruther, M., Bischof, H., Schweighofer, G.: Fast and accurate environment modeling using three-dimensional occupancy grids. In: Proceedings of the 2011 IEEE International Conference on Computer Vision Workshops (2011)
10. Thrun, S.: Learning occupancy grids with forward models. In: Proceedings of the 2001 IEEE/RSU International Conference on Intelligent Robots and Systems (2001)
11. Yamauchi, B.: A frontier-based approach for autonomous exploration. In: International Symposium on Computational Intelligence in Robotics and Automation, pp. 146–151. IEEE (1997)
12. Yamauchi, B.: Frontier-based exploration using multiple robots. In: Second International Conference on Autonomous Agents, pp. 47–53. ACM (1998)
13. Stachniss, C., Grisetti, G., Burgard, W.: Information gain-based exploration using Rao-Blackwellized particle filters. In: RSS, pp. 65–72 (2005)
14. Joho, D., Stachniss, C., Pfaff, P., Burgard, W.: Autonomous exploration for 3D map learning. In: Berns, K., Luksch, T. (eds.) Autonome Mobile Systeme (AMS), pp. 22–28. Springer (2007)
15. Kaufman, E., Lee, T., Ai, Z., Moskowitz, I.S.: Bayesian occupancy grid mapping via an exact inverse sensor model. In: American Control Conference, pp. 5709–5715. IEEE (2016)
16. Thrun, S.: Learning occupancy grid maps with forward sensor models. Auton. Robot. **15**(2), 111–127 (2003)
17. Chen, H., Ding, S., Chen, X., Wang, L., Zhu, C., Chen, W.: Global finite-time stabilization for nonholonomic mobile robots based on visual servoing. Int. J. Adv. Robot. Syst. **11**, 11 (2014)
18. Khoshelham, K., Elberink, S.O.: Accuracy and resolution of kinect depth data for indoor mapping applications. Sensors, pp. 1437–1454 (2012)
19. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numer. Math. **1**(1), 269–271 (1959)
20. Gasilov, N., Dogan, M., Arici, V.: Two-stage shortest path algorithm for solving optimal obstacle avoidance problem. IETE J. Res. **57**(3), 278–285 (2011)
21. C. Inc. (2016) Robotics: estimation and learning. [Online]. Available: https://www.coursera.org/
22. Kümmerle, R., Steder, B., Dornhege, C., Ruhnke, M., Grisetti, G., Stachniss, C., Kleiner, A.: On measuring the accuracy of slam algorithms. Auton. Robot. **27**(4), 387–407 (2009)

Evan Kaufman received his B.S. degree in Mechanical Engineering from Bucknell University in 2012, and is currently a Ph.D. candidate at The George Washington University in the Mechanical and Aerospace department. His research focusses are in robotic mapping and exploration with range-based sensors, estimation, data association, and flight controls. He has worked on several research projects through the U.S. Air Force Research Laboratory and Naval Research Laboratory.

Kuya Takami is a postdoctoral research fellow at the Department of Mechanical and Aerospace Engineering at the George Washington University. He received his doctoral degree in Mechanical Engineering at Virginia Polytechnic Institute and State University in 2015 specializing in robot perception. His main interests include mobile robot perception and autonomous systems.

Taeyoung Lee is an associate professor of the Department of Mechanical and Aerospace Engineering at The George Washington University. He received his doctoral degree in Aerospace Engineering and his master's degree in Mathematics at the University of Michigan in 2008. His research interests include geometric mechanics and control with applications to complex aerospace systems.

Zhuming Ai Ph.D., is a Computer Engineer in the Information Technology Division at the U.S. Naval Research Laboratory (NRL). Before he joined NRL, he served as the Director of the Virtual Reality in Medicine Lab at the University of Illinois at Chicago. He is interested in virtual reality, augmented reality, computer graphics, image processing, and robotics.