Information Theoretic Active Exploration in Signed Distance Fields

Kelsey Saulnier[1] Nikolay Atanasov[2] George J. Pappas[1] Vijay Kumar[1]

Abstract— **This paper focuses on exploration and occupancy mapping of unknown environments using a mobile robot. While a truncated signed distance field (TSDF) is a popular, efficient, and highly accurate representation of occupancy, few works have considered optimizing robot sensing trajectories for autonomous TSDF mapping. We propose an efficient approach for maintaining TSDF uncertainty and predicting its evolution from potential future sensor measurements without actually receiving them. Efficient uncertainty prediction is critical for long-horizon optimization of potential sensing trajectories. We develop a deterministic tree-search algorithm that evaluates the information gain between the TSDF distribution and potential observations along sequences of robot motion primitives. Efficient planning is achieved by branch-and-bound pruning of uninformative sensing trajectories. The effectiveness of our active TSDF mapping approach is evaluated in several simulated environments with complex visibility constraints.**

I. Introduction

Signed distance functions describe the distance of each point in space to the surface of the nearest object with a sign indicating whether the point is inside or outside of the object. The surface of the objects can be readily extracted as zero crossings of the function. Signed distance functions have been used widely in the computer graphics community to represent surfaces of complex shapes [1]. A discrete form of signed distance function, Signed Distance Fields (SDFs), were introduced by Curless et al. as an optimal method for estimating surfaces from range images [2] and later used to great effect by Newcombe et al. with their Kinect Fusion algorithm [3] and by the many following algorithms [4], [5]. Mesh representations of the environment can be extracted from the SDF representation by finding zero crossings, for example using Marching Cubes [6]. More recently SDFs have seen some use in the robotics community as an alternative for the traditional occupancy grid for representing maps [7], [8], [9]. The SDF representation offers some advantages over occupancy grid maps as a method for representing an environment. First, checking collision in an SDF is trivial. For that reason versions of SDFs are often used in motion planning algorithms which use potential fields to increase the cost of unsafe paths [10], [11]. Second, the obstacle surface in an SDF can represent general surface contours more accurately than an occupancy grid because the surface is represented as a zero crossing of the field and not directly tied to the edges of the underlying grid as it is in an occupancy grid.

Active exploration has been widely studied as well. Yamauchi made the first foray into generally applicable autonomous exploration with his frontier seeking approach [12]. Holz et al. provide an evaluation of various frontier-based exploration strategies in 2D environments [13]. Oleynikova et al. also use the idea of seeking unknown areas by using number of new cells that might be seen if the robot moves to a new position as a measure of exploration potential [11].

Although frontier based approaches work well as evidenced by their continued use, they are fundamentally based on geometry and measures of coverage which do not take into account models of uncertainty. Information theoretic exploration provides an alternative to frontier based approaches in which increasing the certainty of the known map is rewarded as well as expanding the map into unknown spaces. Julian et al. examine the properties of mutual information reward surfaces and prove it is an effective value function [14]. Some specific examples of information theoretic exploration are as follows. Atanasov et al. use information theoretic measures to actively track the state of a target. By utilizing Gaussian assumptions, they compute information efficiently and plan open loop trajectories [15]. Charrow et al. use Cauchy-Schwarz quadratic mutual information to evaluate the informativeness of trajectories in occupancy grid maps [16].

Most closely related is the work of Krainin et al. [17]. The authors used an information-based cost function and a TSDF to autonomously choose next views to build 3D models of an object being manipulated by a robotic arm. The authors calculate predicted information gain on the mesh surface instead per TSDF cell and do not consider the sensor noise model.

The contribution of this paper is an information theoretic active exploration algorithm which can be used to autonomously build signed distance field maps of unknown environments. This contribution includes a method for updating the signed distance field given measurements from a realistic sensor model with distance dependant variance as well as an approach for predicting future measurements along a trajectory. In Section II we define the active exploration problem in the case of SDF maps. In Section III we define our SDF update and prediction methods. In Section IV we formulate the equation for predicted information gain along a trajectory. Section V shows how informative trajectories are selected. Lastly, in Section VI we verify our method in

We gratefully acknowledge support from ARL grant DCIST CRA W911NF-17-2-0181, NSF grant CNS-1521617, and ARO grant W911NF-13-1-0350.

[1]The authors are with GRASP Laboratory at the University of Pennsylvania, Philadelphia, PA, 19104, USA {saulnier, pappasg, kumar}@seas.upenn.edu

[2]The author is with the Department of Electrical and Computer Engineering, University of California, San Diego, La Jolla, CA 92093, USA natanasov@ucsd.edu

three 2D simulated environments of varying complexity.

II. Problem Statement

Consider a robot with kinematics:

$$\dot{\mathbf{p}} = \mathbf{R}\mathbf{v} \qquad \dot{\mathbf{R}} = \mathbf{R}\boldsymbol{\omega}_\times \tag{1}$$

where $\mathbf{p} \in \mathbb{R}^3$ is the world-frame position, $\mathbf{R} \in SO(3)$ is the world-frame orientation, $\mathbf{v} \in \mathbb{R}^3$ is the body-frame linear velocity, $\boldsymbol{\omega} \in \mathbb{R}^3$ is the body-frame angular velocity and $(\cdot)_\times : \mathbb{R}^3 \to SO(3)$ is the hat operator mapping vectors to skew-symmetric matrices. The robot is controlled by piecewise-constant velocities $\mathbf{v}(t) \equiv \mathbf{v}_k$, $\boldsymbol{\omega}(t) \equiv \boldsymbol{\omega}_k$ for $t \in [t_k,\ t_{k+1})$, leading to discrete-time kinematics:

$$\begin{aligned}\mathbf{p}_{k+1} &= \mathbf{p}_k + \mathbf{R}_k \mathbf{J}_L((t_{k+1} - t_k)\boldsymbol{\omega}_k)\mathbf{v}_k,\\ \mathbf{R}_{k+1} &= \mathbf{R}_k \exp\left((t_{k+1} - t_k)(\boldsymbol{\omega}_k)_\times\right),\end{aligned} \tag{2}$$

where $\mathbf{p}_k := \mathbf{p}(t_k)$, $\mathbf{R}_k := \mathbf{R}(t_k)$, and $\mathbf{J}_L(\boldsymbol{\omega})$ is the left Jacobian of $SO(3)$ with closed-form expression in [18, Ch.10]. The twist $\mathbf{u}_k := \left(\mathbf{v}_k^\top, \boldsymbol{\omega}_k^\top\right)^\top$ is selected from a *finite* set $\mathcal{U}$ of possible inputs.

The robot is tasked with creating a map of an unknown static environment. The environment is discretized into a finite set of voxels. The true unknown map is represented as a vector $\mathbf{m}$, whose elements $\mathbf{m}_i \in \mathbb{R}$ specify the signed distance from each voxel centroid to the nearest obstacle. In detail, a positive distance $\mathbf{m}_i > 0$ indicates that voxel i is located in free space, while a negative distance signifies that the voxel is within an obstacle. The robot assumes a prior Gaussian distribution over the signed distance field:

$$\mathbf{m} \sim \mathcal{N}\left(\boldsymbol{\mu}_0, \mathbf{diag}(\boldsymbol{\sigma}_0)^2\right) \tag{3}$$

where $\boldsymbol{\sigma}_{0,i} \in \mathbb{R}$ is the standard deviation of the signed distance associated with the i-th voxel. While an independence assumption among the signed distance values at different voxels may lead to a loss in accuracy, it significantly reduces the memory required by the representation in large environments as well as the computational complexity of its updates.

The robot is equipped with a depth sensor (e.g., lidar or depth camera) measuring the distances $\mathbf{r}_{k,j}$ from the robot to the visible obstacles at time k along a set of rays $\boldsymbol{\rho}_j \in \mathbb{R}^3$. The rays are defined in homogeneous coordinates in the sensor frame, e.g., $\mathbf{e}_3^\top \boldsymbol{\rho}_j = 1$ where $\mathbf{e}_3$ is a standard basis vector. Given a visible obstacle point $\mathbf{x}_j$ along the j-th ray, the sensor returns:

$$\mathbf{z}_{k,j} = \mathbf{r}_{k,j} + \boldsymbol{\eta}_{k,j}, \tag{4}$$

where the true depth $\mathbf{r}_{k,j}$ and the noise $\boldsymbol{\eta}_{k,j}$ satisfy:

$$\mathbf{r}_{k,j} := \mathbf{e}_3^\top \mathbf{R}_k^\top (\mathbf{x}_j - \mathbf{p}_k) \qquad \boldsymbol{\eta}_{k,j} \sim \mathcal{N}(0, \lambda(\mathbf{r}_{k,j}^2)^2) \tag{5}$$

Note that the standard deviation λ of the measurement noise depends on the squared distance $\mathbf{r}_{k,j}^2$. This noise model has been empirically shown to work well with RGBD sensors such as the Kinect camera [19], [20].

The sensor depth measurements $\mathbf{z}_k$ are used to infer measurements associated with the signed distance field $\mathbf{m}$. Let $\mathbf{x}_i \in \mathbb{R}^3$ be the centroid of the i-th voxel. Each voxel $\mathbf{x}_i$ within the sensor field of view is projected onto the image plane to determine which measurements $\mathbf{z}_{k,j}$ it is associated with. A depth measurement $\hat{\mathbf{z}}_{k,i}$ is associated with $\mathbf{x}_i$ via interpolation from the depth measurements $\mathbf{z}_{k,j}$ of the two nearest rays $\boldsymbol{\rho}_j$. This measurement is exact in cases where the object measured is flat between the measured points. In detail, let ρ_{j_1}, ρ_{j_2} be the rays that minimize:

$$\left\|\rho_j - \frac{1}{\mathbf{e}_3^\top \mathbf{R}_k^\top (\mathbf{x}_i - \mathbf{p}_k)} \mathbf{R}_k^T (\mathbf{x}_i - \mathbf{p}_k)\right\|^2. \tag{6}$$

Then, the measurement associated with $\mathbf{x}_i$ is:

$$\begin{aligned}\hat{\mathbf{z}}_{k,i} &= h_k(\mathbf{z}_{k,j_1}, \mathbf{z}_{k,j_2}, \mathbf{x}_i)\\ &= \alpha \mathbf{z}_{k,j_1} + (1-\alpha)\mathbf{z}_{k,j_2} - \mathbf{e}_3^\top \mathbf{R}_k^\top (\mathbf{x}_i - \mathbf{p}_k)\\ &= \alpha \mathbf{r}_{k,j_1} + (1-\alpha)\mathbf{r}_{k,j_2} - \mathbf{e}_3^\top \mathbf{R}_k^\top (\mathbf{x}_i - \mathbf{p}_k) + \hat{\boldsymbol{\eta}}_{k,i},\\ &= h_k(\mathbf{r}_{k,j_1}, \mathbf{r}_{k,j_2}, \mathbf{x}_i) + \hat{\boldsymbol{\eta}}_{k,i},\end{aligned} \tag{7}$$

where the noise $\hat{\boldsymbol{\eta}}_{k,i} \sim \mathcal{N}\left(0, \boldsymbol{\zeta}_{k,i}^2(\mathbf{r}_{k,j_1}, \mathbf{r}_{k,j_2})\right)$ has variance $\boldsymbol{\zeta}_{k,i}^2(\mathbf{r}_{k,j_1}, \mathbf{r}_{k,j_2}) := \alpha^2 \lambda(\mathbf{r}_{k,j_1}^2)^2 + (1-\alpha)^2 \lambda(\mathbf{r}_{k,j_2}^2)^2$, and $\alpha \leq 1$.

Given a sequence of measurements $\hat{\mathbf{z}}_{1:k}$ associated with the voxels $\mathbf{x}$ within the sensor field of view, the signed distance field distribution can be updated via the extended Kalman filter:

$$\mathbf{m} \mid \hat{\mathbf{z}}_{1:k} \sim \mathcal{N}\left(\boldsymbol{\mu}_k, \mathbf{diag}\left(\boldsymbol{\sigma}_k\right)^2\right) \tag{8}$$

Problem 1. *Given a prior Gaussian distribution over the signed distance field $\mathbf{m}$ with $\mathbf{m}_i \perp \mathbf{m}_j,\ i \neq j$ and mean $\boldsymbol{\mu}_k$ and standard deviation $\boldsymbol{\sigma}_k$, choose a sequence of twists $\mathbf{u}_k, \ldots, \mathbf{u}_{k+K-1} \in \mathcal{U}$ such that the mutual information between $\mathbf{m}$ and the observations $\hat{\mathbf{z}}_{k+1}, \hat{\mathbf{z}}_{k+K}$ is maximized:*

$$\begin{aligned}\max_{\mathbf{u}_k, \ldots, \mathbf{u}_{k+K-1} \in \mathcal{U}} &\ I(\mathbf{m}; \hat{\mathbf{z}}_{k+1}, \ldots, \hat{\mathbf{z}}_{k+K} | \hat{\mathbf{z}}_{1:k})\\ \textit{s.t.}&\ \textit{eq. } (2),\ \textit{eq. } (7)\end{aligned} \tag{9}$$

III. Truncated Signed Distance Field

Maintaining an SDF distribution $\boldsymbol{\mu}_{k,i}$, $\boldsymbol{\sigma}_{k,i}$ for every voxel $\mathbf{x}_i$ in the environment is not feasible when memory is limited. In fact, the SDF values of all voxels can be recovered based only on the obstacle boundaries, i.e., points where the SDF is zero. Hence, it is sufficient to keep in memory only voxels with indices $\mathcal{I} := \{i \mid |\mathbf{m}_i| < \tau\}$, whose SDF is truncated by a threshold $\tau > 0$. This map representation is called a *truncated signed distance field* (TSDF) and is widely used for representing maps [7], [3], [8]. Voxels $\mathbf{x}_i$ for $i \notin \mathcal{I}$ in known space far from an obstacle surface or those that have not been observed yet may be stored for exploration and planning purposes but their SDF values $\mathbf{m}_i$ are not stored or updated.

A. Updating TSDF Voxels

We rely on measurements $\hat{\mathbf{z}}_k$ at time k, obtained according to eq. (7) to update a prior SDF distribution with parameters $\boldsymbol{\mu}_k$, $\boldsymbol{\sigma}_k$. The independence assumption for the SDF values allows each voxel to be updated separately. Since after interpolation in (7) $\hat{\mathbf{z}}_{k,i}$ is a direct measurement of $\mathbf{m}_i$, the posterior parameters $\boldsymbol{\mu}_{k+1,i}$, $\boldsymbol{\sigma}_{k+1,i}$ can be computed via a Kalman filter update:

$$\begin{aligned}\mathbf{k}_{k,i} &= \boldsymbol{\sigma}_{k,i}^2 \left(\boldsymbol{\sigma}_{k,i}^2 + \boldsymbol{\zeta}_{k,i}^2(\mathbf{z}_{k,j_1}, \mathbf{z}_{k,j_2})\right)^{-1} \\ \boldsymbol{\mu}_{k+1,i} &= \boldsymbol{\mu}_{k,i} + \mathbf{k}_{k,i}\left(h_k(\mathbf{z}_{k,j_1}, \mathbf{z}_{k,j_2}, \mathbf{x}_i) - \boldsymbol{\mu}_{k,i}\right) \\ \boldsymbol{\sigma}_{k+1,i}^2 &= (1 - \mathbf{k}_{k,i})\boldsymbol{\sigma}_{k,i}^2\end{aligned} \tag{10}$$

In the context of exploring an unknown environment, voxels will be encountered for which there is no prior because $\hat{\mathbf{z}}_k, i$ is the first measurement made of that voxel. In those cases, a voxel is initialized with the mean and variance of $\hat{\mathbf{z}}_k, i$ and added to $\mathbf{m}$. Note that the update above can be simplified to:

$$\begin{aligned}\boldsymbol{\sigma}_{k+1,i}^2 &= \left(\frac{1}{\boldsymbol{\sigma}_{k,i}^2} + \frac{1}{\boldsymbol{\zeta}_{k,i}^2(\mathbf{z}_{k,j_1}, \mathbf{z}_{k,j_2})}\right)^{-1} \\ \boldsymbol{\mu}_{k+1,i} &= \boldsymbol{\sigma}_{k+1,i}^2\left(\frac{\boldsymbol{\mu}_{k,i}}{\boldsymbol{\sigma}_{k,i}^2} + \frac{h_k(\mathbf{z}_{k,j_1}, \mathbf{z}_{k,j_2}, \mathbf{x}_i)}{\boldsymbol{\zeta}_{k,i}^2(\mathbf{z}_{k,j_1}, \mathbf{z}_{k,j_2})}\right)\end{aligned} \tag{11}$$

which can be recognized as the standard SDF update with weights $w_{k,i} = 1/\boldsymbol{\sigma}_{k,i}^2$ which was proposed by Curless [21] to estimate a least squares optimal estimate of the obstacle boundary assuming measurements of the distances that are Gaussian along the line of sight of the sensor. The main difference with [21] is that the noise of the measurements depends on the distance between the robot and the obstacle surface.

B. Predicting TSDF Updates

When exploring an uncertain map, it is essential to be able to predict the change in variance of $\mathbf{m}$ from potential future sensor measurements without actually receiving them. This is critical for planning an informative sensing path based on the current SDF distribution $\boldsymbol{\mu}_k$, $\boldsymbol{\sigma}_k$. To predict a measurement $\hat{\mathbf{z}}_{k,i}$, we must predict the value of $\mathbf{r}_{k,j_1}$ and $\mathbf{r}_{k,j_2}$ as seen in (7). Let $\mathcal{I}_{k,j}$ be the subset of voxels within the sensor field of view intersected by ray ρ_j at time k. Then, the true depth $\mathbf{r}_{k,j}$ of ray ρ_j is closely approximated by

$$\mathbf{r}_{k,j} = \mathbf{m}_{i^*} + \mathbf{e}_3^\top \mathbf{R}_k^\top \left(\mathbf{x}_{i^*} - \mathbf{p}_k\right), \tag{12}$$

where $i^* \in \mathcal{I}_{k,j}$ is the voxel with minimum absolute TSDF value. When $\mathbf{m}_i$ are uncertain, the priors $\boldsymbol{\mu}_{k,i}$, $\boldsymbol{\sigma}_{k,i}$ can be used to obtain a Laplace approximation [21] $\mathbf{r}_{k,j} \sim \mathcal{N}\left(\tilde{\boldsymbol{\mu}}_{k,j}, \tilde{\boldsymbol{\sigma}}_{k,j}^2\right)$, where:

$$\begin{aligned}\tilde{\boldsymbol{\sigma}}_{k,j}^2 &= \left(\sum_{i \in \mathcal{I}_{k,j}} \frac{1}{\boldsymbol{\sigma}_{k,i}^2}\right)^{-1} \\ \tilde{\boldsymbol{\mu}}_{k,j} &= \frac{1}{\tilde{\boldsymbol{\sigma}}_{k,j}^2} \sum_{i \in \mathcal{I}_{k,j}} \frac{1}{\boldsymbol{\sigma}_{k,i}^2}\left(\boldsymbol{\mu}_{k,i} + \mathbf{e}_3^\top \mathbf{R}_k^\top \left(\mathbf{x}_i - \mathbf{p}_k\right)\right)\end{aligned} \tag{13}$$

This approximate Gaussian distribution of $\mathbf{r}_{k,j}$ allows us to compute an update to the variance $\boldsymbol{\sigma}_{k+1,i}$ of voxel i based on (7):

$$\boldsymbol{\sigma}_{k+1,i}^2 = \left(\frac{1}{\boldsymbol{\sigma}_{k,i}^2} + \frac{1}{\boldsymbol{\zeta}_{k,i}^2(\tilde{\boldsymbol{\mu}}_{k,j_1}, \tilde{\boldsymbol{\mu}}_{k,j_2})}\right)^{-1} \tag{14}$$

Note that an accurate update to the mean $\boldsymbol{\mu}_k$ is not possible without actual measurements. However, it is also not needed because the cost function in (9) depends only on $\boldsymbol{\sigma}_k$ as discussed in Sec. IV.

C. Variance Multiplier

While eq. (11) may be used to update the TSDF distribution directly, it may be desirable to artificially increase the measurement variance for cells greater than a chosen distance $\epsilon < \delta$ behind the observed surface indicating uncertainty that those cells behind the surface are within the occupied space. This is effect can recognized as the weight taper as is used in the majority of the TSDF literature [21], [7], [8]. This leads to a weighted measurement variance $w\boldsymbol{\zeta}_{k,i}^2(\mathbf{z}_{k,j_1}, \mathbf{z}_{k,j_2})$, where:

$$w = \begin{cases} 1 & -\epsilon < h_k(\mathbf{z}_{k,j_1}, \mathbf{z}_{k,j_2}, \mathbf{x}_i) < \delta \\ \frac{\delta - \epsilon}{h_k(\mathbf{z}_{k,j_1}, \mathbf{z}_{k,j_2}, \mathbf{x}_i) + \delta} & -\delta < h_k(\mathbf{z}_{k,j_1}, \mathbf{z}_{k,j_2}, \mathbf{x}_i) < -\epsilon \end{cases}$$

This is equivalent to the linear weight taper used in [7].

IV. Information Gain

A strong benefit of using SDF maps over occupancy grid maps is the simplicity of calculating mutual information in them. Under the assumption that the cells in the map hold independent Gaussian variables and that the sensor can provide Gaussian updates to those variables, the Shannon mutual information between the current map and the cell measurement can be computed in closed form.

Given the distribution of the map $\mathbf{m}$, we would like to chose a plan $\mathbf{u}_{1:T}$ producing measurements $\mathbf{z}_{1:T}$ to a planning horizon T such that the uncertainty in the map after those measurements is minimized. Conditional Shannon entropy (H) can be used to quantify uncertainty, but in the case of incremental updates this is equivalent to maximizing the Shannon mutual information, from here on referred to as mutual information (I).

$$H(\mathbf{m}|\mathbf{z}_{1:T}) = H(\mathbf{m}) - I\left(\mathbf{m}; \mathbf{z}_{1:T}|\mathbf{p}_{1:T}\right) \tag{15}$$

The expression $I\left(\mathbf{m}; \mathbf{z}_{1:T}|\mathbf{p}_{1,T}\right)$ can be simplified by the assumption that the elements of $\mathbf{m}$ are independent and the rays in $\mathbf{z}$ are independent given the robot pose $\mathbf{p}$

$$I\left(\mathbf{m}; \mathbf{z}_{1:T}|\mathbf{p}_{1:T}\right) = \sum_{i \in \mathcal{F}_{1\times T}} I\left(\mathbf{m}_i; \hat{\mathbf{z}}_{1:T,i}|\mathbf{p}_{1:T}\right) \tag{16}$$

Where $\mathcal{F}_{1\times T} = \bigcup_{t=1}^{T} \mathcal{F}_t$ and $\mathcal{F}_t$ is the set of voxel indices seen at time t.

$$\begin{aligned}I\left(\mathbf{m}_i; \hat{\mathbf{z}}_{1:T,i}|\mathbf{p}_{1:T}\right) = {} & I\left(\mathbf{m}_i; \hat{\mathbf{z}}_{1,i}|\mathbf{p}_1\right) \\ & + I\left(\mathbf{m}_i; \hat{\mathbf{z}}_{2,i}|\hat{\mathbf{z}}_{1,i}, \mathbf{p}_{1:2}\right) \\ & + ... + I\left(\mathbf{m}_i; \hat{\mathbf{z}}_{T,i}|\hat{\mathbf{z}}_{1:T-1,i}, \mathbf{p}_{1:T}\right)\end{aligned} \tag{17}$$

Thus the mutual information for a sequence of measurements $\mathbf{z}_{1:T}$ can be calculated as

$$I\left(\mathbf{m}; \mathbf{z}_{1:T}|\mathbf{p}_{1:T}\right) = \sum_{i \in \mathcal{F}_{1\times T}} \sum_{t=1}^{T} I\left(\mathbf{m}_i; \hat{\mathbf{z}}_{t,i}|\hat{\mathbf{z}}_{1:t-1,i}, \mathbf{p}_{1:t}\right) \tag{18}$$

with the observation that $I(\mathbf{m}_i; \mathbf{z}_{t,i} | \mathbf{z}_{1:t-1,i}, \mathbf{p}_{1:t-1}) = 0$ for $i \notin \mathcal{F}_t$ indicating if that no measurement of voxel i was made at time t, no information is gained about voxel i at time t.

The quantity $I(\mathbf{m}_i; \mathbf{z}_{t,i} | \mathbf{z}_{1:t-1,i}, \mathbf{p}_{1:t-1})$ is equivalent to the difference

$$H(\mathbf{m}|\mathbf{z}_{1:t-1,i}, \mathbf{p}_{1:t-1}) - H(\mathbf{m}|\mathbf{z}_{1:t,i}, \mathbf{p}_{1:t}) \tag{19}$$

which can be computed via the definition of differential entropy for Gaussian random variables

$$\frac{1}{2}\log(2\pi e \boldsymbol{\sigma}^2_{t-1,i}) - \frac{1}{2}\log(2\pi e \boldsymbol{\sigma}^2_{t,i}) \tag{20}$$

At $t = 1$, $\boldsymbol{\sigma}^2_{t-1,i}$ is known from the map prior for voxel i, $\mathbf{m}_i \sim \mathcal{N}(\boldsymbol{\mu}_{0,i}, \boldsymbol{\sigma}^2_{0,i})$. For $t > 1$, $\boldsymbol{\sigma}^2_{t,i}$ can be estimated using (13) and (14) recursively so that $\mathbf{m}_i|\mathbf{z}_{1:t,i} \sim \mathcal{N}(\boldsymbol{\mu}_{0,i}, \boldsymbol{\sigma}^2_{t,i})$. This leads to the mutual information calculation

$$I(\mathbf{m}_i; \mathbf{z}_{t,i} | \mathbf{z}_{1:t-1,i}, \mathbf{p}_{1:t}) = \frac{1}{2}\log\left(\frac{\boldsymbol{\sigma}^2_{t-1,i} + \boldsymbol{\zeta}^2_{t,i}}{\boldsymbol{\zeta}^2_{t,i}}\right) \tag{21}$$

where $\boldsymbol{\zeta}_{t,i}$ is computed with the prior mean $\boldsymbol{\mu}_{0,i}$ and recursively estimated variance $\boldsymbol{\sigma}^2_{t-1,i}$. The mean is not updated during these predictive steps as this would require new sensor measurements.

V. Planning

Planning is done with a receding horizon where T is the length of the planning horizon indicating the number of successive predicted measurements which are evaluated before selecting the next best step. We seek the next set of twists $\mathbf{u}_{1:T}$ such that

$$\max_{\mathbf{u}_1,\ldots,\mathbf{u}_T \in \mathcal{U}} V = \sum_{i \in \mathcal{F}_{1\times T}} \sum_{t=1}^{T} \frac{1}{2}\log\left(\frac{\boldsymbol{\sigma}^2_{t-1,i} + \boldsymbol{\zeta}^2_{t,i}}{\boldsymbol{\zeta}^2_{t,i}}\right) \tag{22}$$
$$\text{s.t. eq. (2), eq. (7), eq. (13), eq. (14)}$$

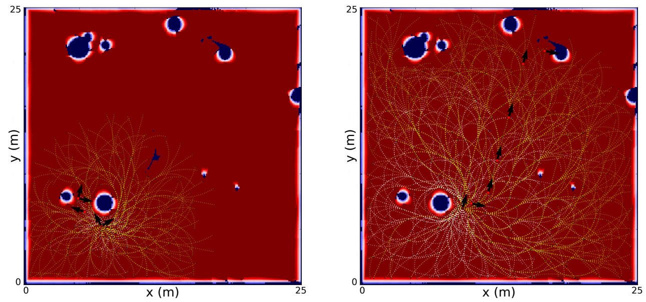
which is Problem 1 given (18) and (21). The complexity of the evaluation of the next best step scales linearly with $\|\mathcal{F}_{1\times T}\|$ the total number of voxel updates predicted, therefore the planning horizon trades off the computational complexity of plan evaluation with the myopia of the resulting plan. The information computation described in (22) can be used with any method for generating sets of trajectories e.g. with lattice based motion primitive planners [22] or pruned motion primitive trees [15].

For the purposes of this paper, trajectories are generated as a tree of successive motion primitives starting with the current robot state and resulting in a set of poses $\mathbf{p}_{1:T} := \{\mathbf{p}_1, ..., \mathbf{p}_T\}$ where $\mathbf{p}_t \in \mathcal{P}_t$. Simple pruning is done to ensure that the tree remains a reasonable size by preventing overlap of new each tree level with the previous levels as an unpruned tree will grow exponentially in size.

The mutual information is calculated for each trajectory $\mathbf{p}_{1:T}$ using V from (22) and the first step of the maximizing plan, $\mathbf{u}_1^*$ is executed.

In a partially explored map which contains frontiers, the robot may encounter voxels for which no previous measurements have been made. For these voxels the information calculations in Section IV cannot be used because there is no prior $\boldsymbol{\mu}_i$ or $\boldsymbol{\sigma}_i$ for computing an expected measurement. Unmeasured voxels adjacent to free space are frontiers and for the purposes of planning these boundaries are treated as obstacle surfaces with high (but bounded) variance. Thus (11) may still be computed with the pseudo-prior $\mathbf{m}_i \sim \mathcal{N}(0, \boldsymbol{\sigma}_0)$ where $\boldsymbol{\sigma}_0$ functions as a parameter controlling the perceived value of free space.

To encourage longer path planning when necessary, the robot will extend its current search tree to another level if the best path does not have more information than a preselected information threshold. An example of a deep search tree can be seen in Figure 1.



(a) T=4 **(b)** T=6

Fig. 1: Search trees of various depths showing possible trajectories shown in white with the chosen trajectory shown with black arrows.

Algorithm 1: Planning

Input Current Pose: $curPose$
Param Min. Horizon: T, Min. Info: INFOMIN
Output Next Pose

```
maxValue = −1
maxValueLeaf = []
MP[0] = generateMotionPrimitives(curPose)
for level = 0 to T do
    for mp in MP[level] do
        (newcells, info) = predictView(mp.Pose)
        value = calculateValue(newcells, info)
        if (value > maxValue) then
            maxValue = value
            maxValueLeaf = mp
        end if
        newMP = generateMotionPrimitives(mp.Pose)
        MP[level + 1].append(newMP)
    end for
    if level == T and (maxValue ≤ INFOMIN) then
        T = T + 1
    else
        break
    end if
end for
return first step that lead to maxValueLeaf
```

VI. SIMULATIONS

We conduct simulated exploration in three 2D environments. The first is a simple square environment containing circular obstacles randomly situated around the space. The second is a simple hallway environment with a few dead ends. The third is the Intel Research Lab, an office floorplan generated from a popular dataset collected by Dirk Haehnel [23]. This environment is an interesting one for exploration because it is complex with both narrow hallways, curved walls, and many rooms.

In our simulations we consider a Kinect sensor whose noise properties have been measured empirically [19], [20]. Though the resulting empirically derived equations differ in minor details, they indicate that σ^2 scales with z^4. For its simplicity we choose [19] for simulations.

$$\sigma^2(z) = (1.425 \times 10^{-3} z^2)^2 = 2.03 \times 10^{-6} z^4 \tag{23}$$

Due to the 2D nature of the exploration environment, we simulate a single row of depth pixels from the depth image and use this as our sensor model. In future work we will use the full image for mapping 3D environments.

The robot carrying the sensor is simulated as differential drive vehicle according to (2) with $\mathbf{v}_k = [v_k, 0, 0]$ and $\boldsymbol{\omega}_k = [0, 0, \omega_k]$ as body frame inputs. With this model we construct motion primitives consisting of constant velocity trajectories created using a set of possible linear and angular velocities (v_k, ω_k). As the tree is expanded, the motion primitives are pruned to reduce conflict with the rest of the tree. The pseudo-prior used for calculating information on frontiers was set to be $\sigma_0 = 0.001$. The truncation distance selected to be unequal on the positive and negative parts of the TSDF. The positive truncation distance is chosen to be the robot's radius plus a safety margin to allow for fast collision checking, while negative truncation distance is chosen to be smaller, just large enough to accommodate 2 voxels in the TSDF discretization. The negative truncation distance limits the minimum size of the obstacles as the negative voxel measurements from one surface may no be contained in the interior of the obstacle if it is too thin.

In the first environment, the robot has many options to explore because there are very few obstacles restricting its movement. This simulation is shown in Figure 2 where the robot trajectory can be seen in white. The robot explores the environment by traveling across the diagonal then circling around, discovering much of the free space before returning to refine the measurements it has made.

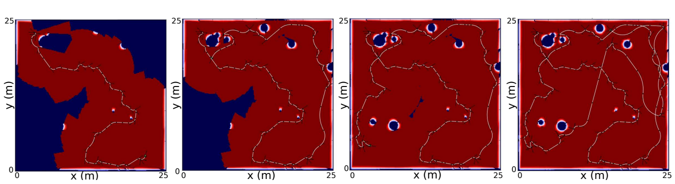


Fig. 2: Robot exploring a nearly open environment. The robot can be seen exploring the much of the free space before touching up the rest of the map.

Figure 3 shows the robot actively exploring the second environment. This environment has a series of narrow hallways which restrict the robot's motion. By $t = 99s$, the robot has explored around the bottom loop and reached its starting point. Between $t = 99s$ and $t = 125s$, the robot returns to improve the TSDF in the small hallway to the right. At $t = 228s$ the robot has explored most of the environment though there are still places of high uncertainty. For the remainder of the simulation the robot refines the TSDF by travelling to locations of high uncertainty.

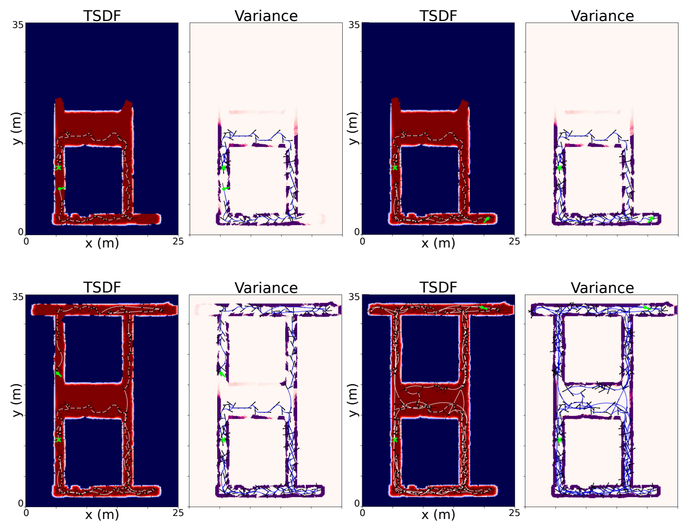


Fig. 3: Snapshots of the robot's exploration of the second environment starting from the green x. From left to right, top to bottom $t = 99s$, $t = 125s$, $t = 228s$, $t = 391s$. At each time step the current TSDF and the current variance, with darker regions indicating a smaller variance. Robot pose is shown with a green dot.

Figure 4 shows the results from the final and most complex environment. The emergent strategy appears the similar to the in the previous environment. At $t = 180s$ the robot has completed a loop around a hallway, from $t = 180s$ to $t = 450s$ the robot travels through the remaining hallways. By $t = 1050s$ much of the map has been seen and the robot returns to observe small areas and improve the TSDF. At $t = 1478s$, the robot can be seen returning to the large room in the top left which is still very uncertain.

The active exploration algorithm proposed in this paper was evaluated in the Intel Map environment. The results of this evaluation can be seen in Figure 5. The first two plots show that the entropy $H(\mathbf{m}|\mathbf{z}_{1:t})$ decreases with time The average entropy $H(\mathbf{m}|\mathbf{z}_{1:t})/\|\mathbf{m}\|$, shown in the second plot in general decreases. Increases in this quantity are due to high uncertainty measurements as new TSDF cells are discovered. Lastly, the third figure shows the number of TSDF cells in the map compared with the total number of cells that have been seen. This shows the memory efficiency benefits of using the TSDF representation as opposed to the un-truncated SDF.

VII. CONCLUSION & FUTURE WORK

We present a method for active exploration in truncated signed distance fields using Shannon mutual information.

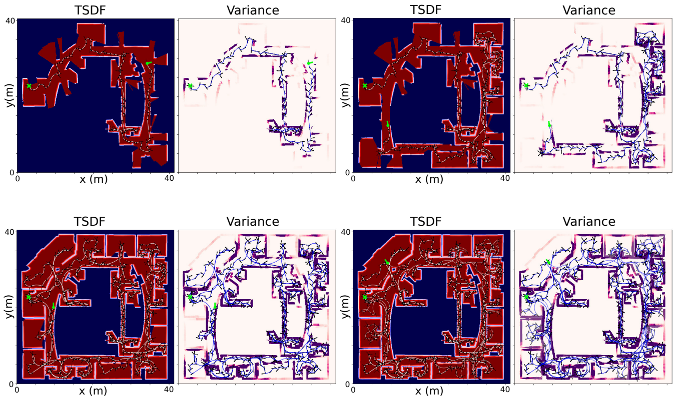


Fig. 4: Snapshots of the robot's exploration of the Intel Map starting from the green x. From left to right, top to bottom $t = 180s$, $t = 450s$, $t = 1050s$, $t = 1478s$. At each time step the current TSDF and the current variance, with darker regions indicating a smaller variance. Robot pose is shown with a green dot.

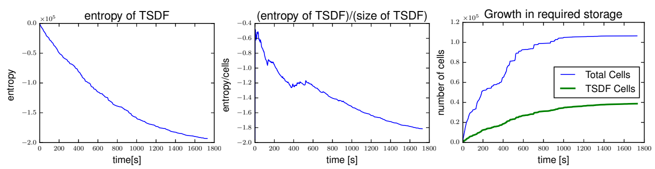


Fig. 5: Evaluation of the entropy and cell storage while exploring the Intel Map: Left: The sum of entropy in the TSDF not counting unknown cells. Middle: The average entropy of a cell in the TSDF. Right: Compares the number of cells which have been explored to the number of cells required to be stored in the TSDF.

This method is able to efficiently compute Shannon mutual information by taking advantage of the closed form expression for Shannon mutual information when using Gaussian random variables. Used in conjunction with a graph or tree search trajectory planning method, trajectories can be chosen to decrease the uncertainty in the environment. Our proposed SDF update method directly uses a realistic sensor model for an RGB-D camera whose depth measurement variance is dependant on the distance from the measured surface. We show a robot equipped with our proposed active exploration algorithm successfully explore three simulated 2D environments of varying complexity. Future work will focus on improving the map estimation by relaxing the independence assumption between cells in the map, accounting for uncertainty in localization of the robot within the map, and cooperative exploration with multiple robots.

REFERENCES

[1] J. C. Hart, "Sphere tracing: a geometric method for the antialiased ray tracing of implicit surfaces," *The Visual Computer*, vol. 12, no. 10, pp. 527–545, 1996.

[2] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," *in Proc. of SIGGRAPH*, pp. 303–312, 1996.

[3] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," *10th IEEE Int. Symp. on Mixed and Augmented Reality (ISMAR)*, pp. 127–136, 2011.

[4] L. Han and L. Fang, "FlashFusion: Real-time Globally Consistent Dense 3D Reconstruction using CPU Computing," in *Robotics: Science and Systems (RSS)*, Pittsburgh, PA, 2018.

[5] T. Whelan, J. McDonald, M. Kaess, M. Fallon, H. Johannsson, and J. J. Leonard, "Kintinuous: Spatially Extended KinectFusion," *RSS Workshop on RGB-D: Advanced Reasoning with Depth Cameras*, p. 7, 2012.

[6] E. V. Chernyaev, "Marching Cubes 33: Construction of Topologically Correct Isosurfaces," CN/95-17, CERN, Tech. Rep., 1995.

[7] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto, "Voxblox: Incremental 3D Euclidean signed distance fields for on-board MAV planning," in *IEEE Int. Conf. on Intelligent Robots and Systems*, 2017, pp. 1366–1373.

[8] E. Bylow, J. Sturm, C. Kerl, F. Kahl, and D. Cremers, "Real-Time Camera Tracking and 3D Reconstruction Using Signed Distance Functions," *Robotics: Science and Systems IX (RSS)*, 2009.

[9] B. Lee, C. Zhang, Z. Huang, and D. D. Lee, "Online Continuous Mapping using Gaussian Process Implicit Surfaces," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2019.

[10] M. Zucker, N. Ratlif, A. D. Dragan, M. Pivtoraiko, M. Klingensmith, C. M. Dellin, J. A. Bagnell, and S. S. Srinivasa, "CHOMP: Covariant Hamiltonian Optimization for Motion Planning," *Int. Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1164–1193, 2013.

[11] H. Oleynikova, M. Burri, Z. Taylor, J. I. Nieto, R. Siegwart, and E. Galceran, "Continuous-time trajectory optimization for online uav replanning," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2016.

[12] B. Yamauchi, "A frontier-based approach for autonomous exploration," in *Proc. IEEE Int. Symp. on Computational Intelligence in Robotics and Automation (CIRA)*, 1997, pp. 146–151.

[13] D. Holz, N. Basilico, F. Amigoni, and S. Behnke, "Evaluating the Efficiency of Frontier-based Exploration Strategies," in *41st Int. Symp. on Robotics (ISR) and 6th German Conference on Robotics (ROBOTIK)*, 2010, pp. 36–43.

[14] B. J. Julian, S. Karaman, and D. Rus, "On mutual information-based control of range sensing robots for mapping applications," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. IEEE, 2013.

[15] N. A. Atanasov, J. Le Ny, and G. J. Pappas, "Distributed Algorithms for Stochastic Source Seeking With Mobile Robot Networks," *Journal of Dynamic Systems, Measurement, and Control*, vol. 137, no. 3, oct 2014.

[16] B. Charrow, S. Liu, V. Kumar, and N. Michael, "Information-theoretic mapping using Cauchy-Schwarz Quadratic Mutual Information," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2015.

[17] M. Krainin, B. Curless, and D. Fox, "Autonomous generation of complete 3D object models using next best view manipulation planning," *IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011.

[18] G. Chirikjian, *Stochastic Models, Information Theory, and Lie Groups, Volume 2: Analytic Methods and Modern Applications*. Springer Science & Business Media, 2011, vol. 2.

[19] K. Khoshelham and S. O. Elberink, "Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications," *Sensors*, vol. 12, no. 2, pp. 1437–1454, feb 2012.

[20] C. V. Nguyen, S. Izadi, and D. Lovell, "Modeling kinect sensor noise for improved 3D reconstruction and tracking," *Proc. 2nd Joint 3DIM/3DPVT Conf.: 3D Imaging, Modeling, Processing, Visualization and Transmission, (3DIMPVT)*, pp. 524–530, 2012.

[21] B. L. Curless, "New Methods for Surface Reconstruction From Range Images," Ph.D. dissertation, Stanford University, 1997.

[22] S. Liu, "Motion planning for micro aerial vehicles," Ph.D. dissertation, University of Pennsylvania, 2018.

[23] D. Hähnel, "Intel research lab," http://www.ipb.uni-bonn.de/datasets/.