Information-Theoretic Planning with Trajectory Optimization for Dense 3D Mapping

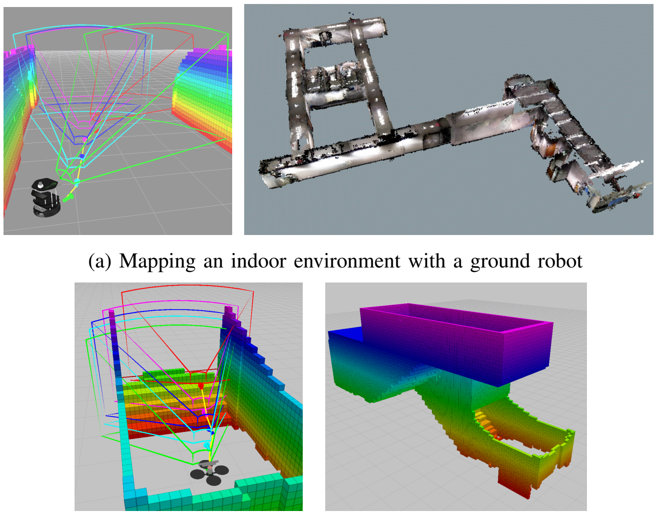
Benjamin Charrow, Gregory Kahn, Sachin Patil, Sikang Liu,
Ken Goldberg, Pieter Abbeel, Nathan Michael and Vijay Kumar

***Abstract*—We propose an information-theoretic planning approach that enables mobile robots to autonomously construct dense 3D maps in a computationally efficient manner. Inspired by prior work, we accomplish this task by formulating an information-theoretic objective function based on Cauchy-Schwarz quadratic mutual information (CSQMI) that guides robots to obtain measurements in uncertain regions of the map. We then contribute a two stage approach for active mapping. First, we generate a candidate set of trajectories using a combination of global planning and generation of local motion primitives. From this set, we choose a trajectory that maximizes the information-theoretic objective. Second, we employ a gradient-based trajectory optimization technique to locally refine the chosen trajectory such that the CSQMI objective is maximized while satisfying the robot's motion constraints. We evaluated our approach through a series of simulations and experiments on a ground robot and an aerial robot mapping unknown 3D environments. Real-world experiments suggest our approach reduces the time to explore an environment by 70% compared to a closest frontier exploration strategy and 57% compared to an information-based strategy that uses global planning, while simulations demonstrate the approach extends to aerial robots with higher-dimensional state.**

I. Introduction

Using mobile robots to construct dense 3D maps of environments has the potential to benefit a number of industries such as mining, construction, and agriculture, and to plan rescue efforts in disaster response operations. It is also an important prerequisite for autonomous robots like self-driving cars that operate in unstructured environments. In this work, we focus on the problem of enabling robots to autonomously construct such maps as efficiently and quickly as possible.

Mapping a 3D environment with a mobile robot is a challenging problem. As the robot knows nothing about the environment initially, it must be able to effectively predict how future measurements will reduce the map's uncertainty. However, due to cost or payload constraints, the robot may only be equipped with noisy sensors that suffer from short maximum ranges and limited fields of view, necessitating planning over multiple time steps. Generating these plans can be substantially complicated by the robot's own mobility limitations, which preclude it from obtaining measurements at arbitrary positions throughout the environment. Many robots – in particular aerial vehicles – also have limited battery life. Consequently, if a robot is incapable of quickly determining how it should move, it will run out of power before it has obtained a complete and low uncertainty map. Finally, robots constantly face an exploitation-exploration trade-off during the mapping process of whether to improve the map in their local vicinity, or try to explore other parts of the environment.



(a) Mapping an indoor environment with a ground robot

(b) Mapping a stairwell environment with an aerial robot

Fig. 1: The goal of this work is to quickly construct dense 3D maps using mobile robots equipped with sensors that have a limited field of view and sensing range. (a) Mapping an indoor environment with a ground robot equipped with a RGB-D sensor. (b) Mapping a stairwell environment with a quadrotor equipped with a RGB-D sensor. We contribute a two stage planning approach: 1) using a global planner and local motion primitives to choose trajectories that maximize an information-theoretic objective based on Cauchy-Schwarz quadratic mutual information (CSQMI) and 2) using trajectory optimization to further maximize the CSQMI objective.

There is extensive prior work on devising information-theoretic objectives that guide a mobile robot to obtain measurements in uncertain regions of the map. There are also several different strategies for generating control actions for the robot including local exploration strategies involving greedy gradient ascent [1, 21, 41] or local motion primitives [2, 4, 12]; and global strategies that use a variant of the frontier method to maximize the chosen objective [5, 39]. Local strategies are useful when the robot is close to uncertain portions of the map and constantly observing new areas (Fig. 2a), but are susceptible to local minima [44]. Global strategies can escape local minima when the robot is far from uncertain portions of the map (Fig. 2b), but are computationally expensive and may produce coarse paths that result in jagged robot motion [42].

In this work, we build on our previous approach [5] of using Cauchy-Schwarz quadratic mutual information (CSQMI) [38] for active mapping. The primary contribution of this paper is a two stage approach for generating and refining trajectories that substantially improves a robot's ability to quickly reduce

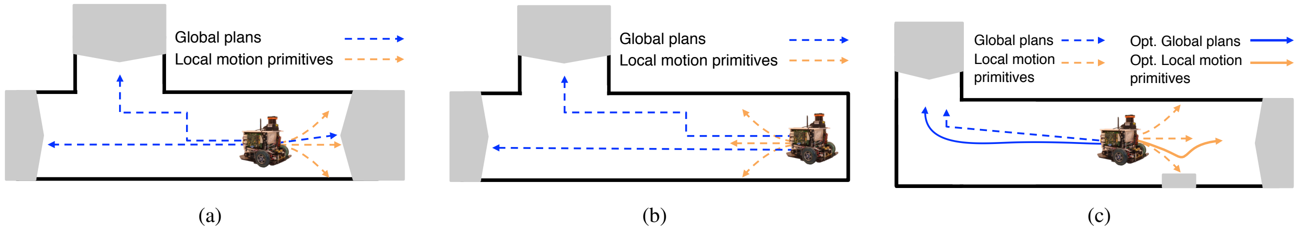


Fig. 2: (a) Local motion primitives are useful when a robot is close to uncertain portions of the map (gray blobs) and needs to react quickly to new information. (b) Global plans are useful when the robot is far from uncertain portions of the map as they can escape local minima in the information-theoretic objective. (c) Trajectory optimization can improve the informativeness of both local motion primitives and global plans while respecting a robot's motion constraints.

the map's uncertainty. First, we generate trajectories using a combination of global planning and local motion primitives, enabling a robot to escape local minima and replan quickly as the map changes. Second, we contribute a method for improving both local motion primitives and global paths using gradient-based trajectory optimization that maximizes the CSQMI objective while satisfying the robot's motion constraints (Fig. 2c). This improvement is particularly important when mapping 3D environments with robots that have high-dimensional state spaces and non-trivial motion models.

We evaluated our approach through a series of simulations and experiments on a ground robot and an aerial robot mapping unknown 3D environments. In real-world experiments we show that by including global plans, local motion primitives, and trajectory optimization, a ground robot was able to construct a complete 3D map 3.3× faster than a strategy that drives to the closest frontier [52] and 2.3× faster than an information-based strategy that only considers global plans [5].

II. Related Work

There is a considerable body of prior work on autonomous mapping and exploration [45, 48]. One key question is how to represent a static map of the environment. Such representations include, but not limited to, topological maps, landmark-based representations, elevation grids, point clouds, meshes, and occupancy grids [49]. In this work, we focus on creating a dense 3D map of the environment using occupancy grids.

In prior work, strategies for mapping and exploration can be primarily grouped into two categories: (i) Frontier-based, and (ii) Information-gain based strategies.

Frontier-based strategies [52] are primarily geometric in nature and travel to the discrete boundary between the free and unknown regions in the map. Extensions of this strategy have been successfully used for building maps of unknown environments in 2D [3, 14]. Holz et al. [18] provide a comprehensive evaluation of frontier-based strategies in 2D environments. Direct extensions to voxel grid representations of 3D environments using frontier voids have also been proposed [10, 13]. Shade and Newman proposed a combination of a frontier-based and a local vector-field based strategy to generate shorter exploration trajectories [42]. Shen et al. [43] used a frontier-based strategy with a particle-based representation of free space for computational tractability.

Information-gain based mapping strategies optimize an information-theoretic measure for exploration. Prior work has investigated minimization of the map entropy, which is a measure of uncertainty associated with all grid cells [31, 51]. Several methods have been proposed that choose to maximize mutual information (MI), which predicts how future sensor measurements will reduce map uncertainty [1, 21, 41]. These approaches use a greedy controller with a one-time step look ahead to maximize MI. However, Soatto [44] suggests planning over multiple time steps to deal with the issue of a greedy explorer getting stuck in local minima. To address this issue, researchers have proposed considering a discrete set of actions and executing the one that maximizes information gain [2, 4, 12]. Charrow et al. [5] used the rate of information gain criterion while Rocha et al. [39] used the gradient of map entropy to select promising frontiers for exploration. These strategies do not locally optimize the robot trajectory to gain more information as the robot is traveling, which might result in inefficient mapping behavior. Kollar and Roy [24] formulated exploration as a constrained optimization problem and learned how to select good trajectories. In contrast, we maximize CSQMI over the continuous control input space.

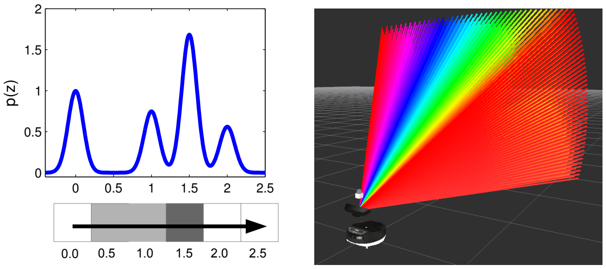
Information-theoretic objectives have also been used for planning and control in robotics for related tasks involving uncertainty such as target tracking [15, 30], target localization [4, 16], inspection [17], extrinsic calibration of LIDAR sensors [28], and visual servoing [8]. Prior work has used trajectory optimization over a finite horizon to optimize information-theoretic criteria [6, 29, 33, 40] for continuous motion and sensing models. In contrast, this work maximizes CSQMI computed over the continuous robot state space and a discrete occupancy grid representation of the environment.

In this work, we assume the robot is capable of localizing itself using a simultaneous localization and mapping (SLAM) [26, 34, 49] system. Modern SLAM systems [34], which account for uncertainty in the map and the robot's state, can be used to construct an occupancy grid using the maximum likelihood estimate of the robot's path. The grid can be updated incrementally and when estimates of previous poses change substantially (e.g., due to loop closures) a completely new occupancy grid can be quickly regenerated [47]. In cases where reliable localization is not possible, it is important to actively reduce both the uncertainty of the map and the robot's state [2, 19, 22, 46, 50].

III. Preliminaries and Problem Definition

A. Occupancy Grid Mapping

We use occupancy grids [49] to represent 3D maps as they are a dense, probabilistic, and volumetric representation



(a)

(b)

Fig. 3: **Beam based measurement model**: (a) 1 beam, the cells it intersects at various distances, and the resulting distribution over measurements when $z_{\text{min}} = 0.5$ and $z_{\text{max}} = 2$. Darker cells have a higher probability of being occupied. (b) Raycasts for multiple beams. Obtained from Charrow et al. [5].

of space. A map, $\mathbf{m}$, is a discretization of 3D space into regular sized cubes or cells $\{c_1, \dots, c_{|\mathbf{m}|}\}$ each of which corresponds to a Bernoulli random variable whose value is 1 if the corresponding region of space contains an obstacle and 0 if it is free. The standard occupancy grid mapping assumes that cells are independent of one another so $p(\mathbf{m}) = \prod_i p(c_i)$ and unobserved cells have a uniform prior of being occupied.

B. Information-Theoretic Objective for Active Control

Given a map $\mathbf{m}$ at time t, we seek a set of control inputs for the robot so that it gathers measurements which reduce the uncertainty of the map as quickly as possible. To achieve this, we formulate an optimization problem over the time interval $\boldsymbol{\tau} \triangleq t+1 : t+T$ to find controls $\mathbf{u}_{\boldsymbol{\tau}} = [\mathbf{u}_t, \dots, \mathbf{u}_{t+T-1}]$ that move the robot to states $\mathbf{x}_{\boldsymbol{\tau}} = [\mathbf{x}_{t+1}, \dots, \mathbf{x}_{t+T}]$ where it will obtain measurements $\mathbf{z}_{\boldsymbol{\tau}} = [\mathbf{z}_{t+1}, \dots, \mathbf{z}_{t+T}]$ that reduce the map's uncertainty. The future states of the robot are determined by the deterministic dynamics of the robot, $\mathbf{x}_{i+1} = f(\mathbf{x}_i, \mathbf{u}_i)$. Our objective is to find controls that maximize the rate of some measure of information gain between the map, $\mathbf{m}$ and future measurements the robot will make, $\mathbf{z}_{\boldsymbol{\tau}}$:

$$\begin{array}{cl} \max\limits_{\mathbf{x}_{\tau},\, \mathbf{u}_{\tau}} & \dfrac{\mathrm{I}[\mathbf{m}; \mathbf{z}_{\boldsymbol{\tau}} \mid \mathbf{x}_{\boldsymbol{\tau}}]}{D(\mathbf{u}_{\boldsymbol{\tau}})} \\ \underset{\forall t < i < t+T-1}{\text{s. t.}} & \mathbf{x}_{i+1} = f(\mathbf{x}_i, \mathbf{u}_i), \\ & \mathbf{x}_i \in \mathcal{X}_{\text{feasible}}, \mathbf{u}_i \in \mathcal{U}_{\text{feasible}}, \end{array} \tag{1}$$

where $\mathrm{I}[\mathbf{m}; \mathbf{z}_{\boldsymbol{\tau}} \mid \mathbf{x}_{\boldsymbol{\tau}}]$ quantifies the expected reduction in the map's uncertainty, $\mathcal{U}_{\text{feasible}}$ is the set of valid controls, $\mathcal{X}_{\text{feasible}}$ is the set states the robot can be in, and $D(\mathbf{u}_{\boldsymbol{\tau}})$ is how long it takes to execute all of the controls. Maximizing the rate of information gain is preferable to purely maximizing information, as it enables the robot to compare the value of actions over different time and length scales.

There are many different choices for quantifying information or how learning the outcome of one random variable (e.g., range measurements) affects another (e.g., the map) [7, 38]. Shannon's mutual information (MI) is one widely used measure. In this paper, we build on our prior work [5] that proposes the use of Cauchy-Schwarz quadratic mutual information (CSQMI) [38] as it can be computed more efficiently. Omitting the conditioning on the robot's poses for brevity, CSQMI can be expressed as:

$$\mathrm{I}_{\text{CS}}[\mathbf{m}; \mathbf{z}_{\boldsymbol{\tau}}] = -\log \frac{\left(\sum \int p(\mathbf{m}, \mathbf{z}_{\boldsymbol{\tau}}) p(\mathbf{m}) p(\mathbf{z}_{\boldsymbol{\tau}})\, d\mathbf{z}_{\boldsymbol{\tau}}\right)^2}{\sum \int p^2(\mathbf{m}, \mathbf{z}_{\boldsymbol{\tau}})\, d\mathbf{z}_{\boldsymbol{\tau}} \sum \int p^2(\mathbf{m}) p^2(\mathbf{z}_{\boldsymbol{\tau}})\, d\mathbf{z}_{\boldsymbol{\tau}}},$$

where the sums are over all possible maps and the integrals are over all possible measurements the robot can receive.

Algorithm 1: Calculate $\mathrm{I}[\mathbf{m}; \mathbf{z}_{\boldsymbol{\tau}} \mid \mathbf{x}_{\boldsymbol{\tau}}]$, the information of measurements made over multiple steps each comprised of multiple 1D beams.

```
$\mathcal{Z}_{\text{indep}} = \emptyset$ // Set of nearly independent measurements
Info = 0 // Info from measurements in $\mathcal{Z}$
for each future 1D beam measurement $z_t^b$ do
    Perform raycast to find cells $\mathbf{c} \subseteq \mathbf{m}$ that $z_t^b$ intersects
    Use probability of cells in $\mathbf{c}$ being occupied or free to
    determine distribution over measurements;
    $p(z_t^b \mid \mathbf{x}_t) = \sum p(\mathbf{c}) p(z_t^b \mid \mathbf{x}_t, \mathbf{c})$
    if isIndependent($z_t^b, \mathcal{Z}_{\text{indep}}$) then
        Info += $\mathrm{I}[\mathbf{m}; z_t^b \mid \mathbf{x}_t]$ // Add info from this beam
        Add $z_t^b$ to $\mathcal{Z}_{\text{indep}}$
```

Calculating information requires a probabilistic model for measurements as a function of the map and the robot's state: $p(\mathbf{z} \mid \mathbf{m}, \mathbf{x})$. We model a measurement at time k as a collection of B one dimensional beams that cover the sensor's field of view $\mathbf{z}_k = [z_k^1, \dots, z_k^B]$. Fig. 3b shows an example model for a typical RGB-D sensor. Next, we assume that given the set of cells a beam intersects, $\mathbf{c}$, it returns the distance to the first occupied one, d, perturbed by Gaussian noise: $p(z \mid d) = \mathcal{N}(z - d, \sigma^2)$. Marginalizing over all states cells can be in makes $p(z_k^b \mid \mathbf{x}_k)$ a Gaussian mixture model (Fig. 3a) [21]. Using this model, the information of an individual beam $\mathrm{I}[\mathbf{m}; z_k^b \mid \mathbf{x}_k]$ can be calculated efficiently [5].

Unfortunately, calculating the information of multiple beams over multiple time steps, $\mathrm{I}[\mathbf{m}; \mathbf{z}_{\boldsymbol{\tau}} \mid \mathbf{x}_{\boldsymbol{\tau}}]$, is expensive, as it involves taking expectations over the joint distribution of all measurements and the map. To avoid this issue, many researchers assume that measurements are independent of one another [20, 21, 25], which might result in poor plans over multiple time steps when there is significant overlap between the sensor field of views. Instead, we calculate information by summing the information from a subset of measurements that are nearly independent (Alg. 1). We determine this subset by bounding the probability that any two beams intersect the same cell. This approximation is valid for any measure of information that is additive over pairwise independent events and can also be implemented efficiently [5].

IV. Approach

We propose a two stage planning approach. We construct a candidate set of trajectories by supplementing global plans with local motion primitives generated by control sampling. We then choose a trajectory that maximizes the objective given by (1). In doing so, we combine the benefits of both global planning and local exploration using primitives, as shown in Fig. 2. In the second stage, we optimize the chosen trajectory

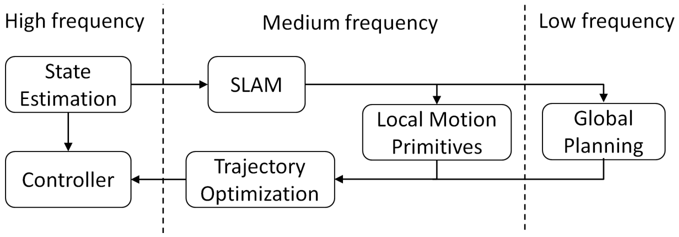


Fig. 4: **System architecture**: We generate a candidate set of trajectories by combining global planning and local motion primitives and then select the one that maximizes the CSQMI objective. This selection is improved using trajectory optimization and then executed by the robot. Different system components run asynchronously at different frequencies.

using gradient-based trajectory optimization over multiple time steps to maximize the CSQMI objective while satisfying the robot's motion model. The controls generated by the trajectory optimization routine are fed to the low level motion controller.

A schematic of our system architecture is shown in Fig. 4.

A. *Combining Global Planning and Local Motion Primitives*

Global Planning: The global planner's task is to generate a set of paths from the robot's current location whose rate of information gain is high. These paths should extend to every portion of the known map, so that the robot can consider the utility of traveling far away from its current location. To generate these paths, we plan shortest paths to destinations where frontiers can be observed, as illustrated in Fig. 5. We previously developed an algorithm to do this [5] and describe it here for completeness.

To start, we find uncertain portions of the map that can likely be observed by the robot by identifying "frontier voxels," which are unknown voxels that are adjacent to voxels that are very likely unoccupied [52]. To reduce the number of paths we need to plan, we greedily cluster multiple nearby frontier voxels by randomly selecting a frontier voxel and grouping it with all other frontier voxels within a user-specified distance (0.4 m worked well for our experiments). After removing these voxels from consideration, we repeat this process until every frontier voxel belongs to a cluster.

Given clusters, we create global paths by finding shortest paths to destinations that can view a cluster. To do this, we use Dijkstra's algorithm to plan single source shortest paths to every destination in the map and at each destination check which clusters are visible. A cluster is visible from a pose, if 1) the cluster's center of mass (i.e., average position of all voxels in the cluster) is within the field of view of the sensor and 2) a raycast from the sensor is unlikely to hit an obstacle. To reduce the number of visibility checks and make the search faster, we precompute poses that can view clusters and store these in a lookup table to query during the search (see [5]).

While the global planner is useful, it has the following shortcomings: 1) generating paths throughout the environment takes a long time; 2) the paths it generates are coarse and may not satisfy the robot's dynamics; and 3) even when paths can be followed, they are frequently not smooth, resulting in jagged robot motion [42].

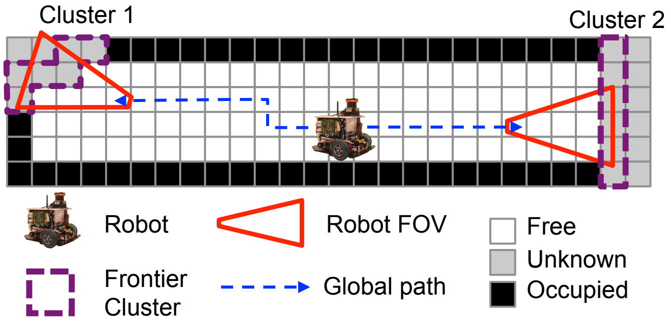


Fig. 5: **Global Planning**: Generate global plans by 1) clustering frontier voxels and 2) planning paths to places that can view them. When a robot can view a cluster from many places, we pick the one that can be reached the fastest.

Local motion primitives: The goal of local motion primitives is to quickly generate short trajectories. This is primarily useful when the robot is near an uncertain part of the map and constantly getting new information that affects where it should go (Fig. 2a). As it plans throughout the entire environment, in these cases, the global planner generates paths too slowly.

We evaluated the benefits of two types of control sampling methods: 1) lattice planners [27] that generate dynamically feasible trajectories using a fixed library of motion primitives that are chained together to plan over multiple timesteps; and 2) randomized planners that generates trajectories by randomly sampling from the robot's control space over the local time horizon. In both cases, if the motion model predicts that the robot might enter an obstacle or unknown space in the environment, the set of controls is rejected. To specify "local," we heuristically generate motions whose total length is comparable to the maximum range of the sensor.

B. *Trajectory Optimization for Refinement*

Neither the global plans nor local motion primitives are locally optimal with respect to the rate of information gain as they do not consider the full control input space of the robot. To address this issue, we use gradient-based trajectory optimization that can refine a trajectory in the continuous control space. Ideally, we would like to optimize all the trajectories within the candidate set and select the one that maximizes the objective. However, optimizing all the candidate trajectories would be computationally expensive due to all the gradient computations involved. Instead, we first select the best trajectory that maximizes the CSQMI objective (1) and use that to initialize the optimization.

Gradient of Information: We consider gradient-based trajectory optimization methods. Because our underlying occupancy grid map representation (Sect. III-A) and beam model for the RGB-D sensor (Sect. III-B) are discrete, it is not immediately obvious that a meaningful gradient exists. Prior work [20] showed that the gradient of information of a single measurement can be calculated for a 2D mapping scenario if one assumes that individual beams are independent and the sensor has a wide field of view. One important observation is that the reward surface is smoother if the position of the robot is restricted to the center of the grid cells. We extend this work by considering the gradient of information of multiple measurements with respect to the robot's full state and control inputs, even when beams are modeled as dependent.

To evaluate gradients, we view the information reward surface as a function of discrete robot positions along the occupancy grid and numerically evaluate gradients using finite differences. Note that the grid cell resolution is often small (e.g., $0.1\,\mathrm{m}$) and that the robot's orientation and control inputs are not restricted to a discrete set.

Sequential Quadratic Programming (SQP): We use SQP [32] to locally optimize the non-convex, constrained optimization problem to find a set of controls over a finite horizon. Strictly speaking, SQP requires that the objective has a Lipschitz continuous second derivative to guarantee a quadratic convergence rate to a local optimum. We cannot formally do this due to nature of the discrete occupancy grid and the sensor model. However, prior work [1, 21, 41] has shown that it is possible to perform gradient ascent on such objectives even in the absence of theoretical guarantees.

To optimize a trajectory, we adapt (1) to a form that can be solved using SQP. In particular, we assume a discrete-time robot motion model with a fixed number of time steps, making the duration penalty a constant that can be removed from the objective. We also encourage the robot's states to be feasible by rewarding being sufficiently far away from obstacles:

$$\begin{aligned} \max_{\mathbf{x}_\tau,\ \mathbf{u}_\tau} \quad & \mathrm{I}[\mathbf{m}; \mathbf{z}_{\boldsymbol{\tau}} \mid \mathbf{x}_{\boldsymbol{\tau}}] + \alpha \sum_{i=t+1}^{t+T} \min\left(d_{\max}, \mathrm{dist}(\mathbf{x}_i, \mathbf{m})\right) \\ \underset{\forall t<i<t+T-1}{\text{s.t.}} \quad & \mathbf{x}_{i+1} = f(\mathbf{x}_i, \mathbf{u}_i), \qquad (2) \\ & \mathbf{x}_i \in \mathcal{X}_{\text{feasible}}, \mathbf{u}_i \in \mathcal{U}_{\text{feasible}}, \end{aligned}$$

where $\mathrm{dist}(\mathbf{x}, \mathbf{m}) \geq 0$ is the minimum unsigned distance from $\mathbf{x}$ to any obstacle in the map, $d_{\max}$ is a parameter that limits the distance reward, and $\alpha \geq 0$ is a weighting parameter that balances the objective. After a parameter search with logged data, we obtained good results with $\alpha = 20$ and $d_{\max} = 0.5$ m. To support constant time distance queries, before the optimization starts, we construct the Euclidean Distance Transform [11] for the entire 3D map. Note that the time horizon can also be included as an optimization variable.

In our implementation, the innermost QP solver was generated by a numerical optimization code framework that generates QPs specialized for convex multistage problems such as trajectory optimization [9]. Even though it is possible to compute the entire Hessian, it is computationally very expensive to do. We used the symmetric rank 1 (SR1) update method to update the Hessian using the computed gradients [32].

Caching Information: Numerically calculating the objective's gradient is computationally expensive as each evaluation requires calculating the information gain resulting from all poses. The complexity of calculating information scales linearly in the number of poses in the trajectory, while the complexity of calculating information from all beams in a pose is determined by the time it takes to evaluate the information of individual beams, t_{Info}, and the time it takes to find a subset of beams that are independent t_{Indep} [5]. With T poses per trajectory, evaluating the gradient via central differences takes $O(T^2(t_{\text{Info}} + t_{\text{Indep}}))$ time.

Fortunately, we can calculate the gradient faster by caching redundant computations. Calculating each partial derivative of the gradient via central differences involves perturbing an individual optimization variable. Because we include the poses in the optimization, each perturbation of $\mathbf{x}$ or $\mathbf{u}$ only affects a pose at a single time step. Consequently, many of the information calculations across different perturbations are redundant and can be cached. Although this process is complicated by the fact that measurements are dependent – meaning information does not decompose into a sum over the information at each pose – we can cache the information gain from individual beams and the cells that the beams pass through using raycasting. This reduces the complexity of evaluating the gradient to $O(Tt_{\text{Info}} + T^2 t_{\text{Indep}})$ which is a substantial speedup as $t_{\text{Info}} \gg t_{\text{Indep}}$. In experiments optimizing over 5 time steps, produces a 4.0× speedup.

V. Experiments

There are four primary questions that we seek to answer with our experiments— **Q1**: How much information gain is obtained by optimizing local motion primitives? **Q2**: Which scenarios do local motion primitives provide the most benefit in? **Q3**: How does our approach compare to previous work and a human teleoperator with respect to speed, distance traveled, and map completion? **Q4**: How well does trajectory optimization scale to high-dimensional systems?

We designed simulation and real-world experiments with a ground robot and simulations with an aerial robot to answer these questions.

A. Platforms and System Details

Ground Robot: We conducted experiments using a differential drive ground robot equipped with a 2D Hokuyo UTM-30LX laser range finder and an Asus Xtion Pro RGB-D camera. The robot's computer was an Intel i5 processor with 8 GB of RAM, enabling it to run our entire approach on-board.

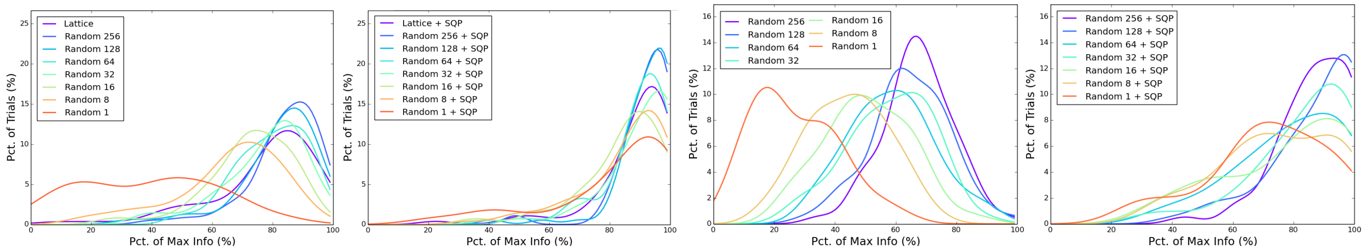
We parameterized the robot's state $\mathbf{x}$ as a 3D vector that consists of the robot's 2D position and orientation. The control input $\mathbf{u}$ is a 2D vector of its left and right wheel speeds:

$$\mathbf{x} = \begin{bmatrix} x, & y, & \theta \end{bmatrix}^\intercal, \ \ \mathbf{u} = \begin{bmatrix} v^l, & v^r \end{bmatrix}^\intercal$$
$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \\ \theta_{i+1} \end{bmatrix} = \begin{bmatrix} x_i \\ y_i \\ \theta_i \end{bmatrix} + \begin{bmatrix} \frac{1}{2}(v_i^l + v_i^r)\cos(\theta_i) \\ \frac{1}{2}(v_i^l + v_i^r)\sin(\theta_i) \\ (v_i^r - v_i^l)/\ell \end{bmatrix}, \qquad (3)$$

where ℓ is the wheel separation. For trajectory optimization, we limited the wheel speeds to be at most 0.5 m/s and imposed the motion model's nonlinear equality constraints.

Aerial Robot: We simulated a quadrotor as an AscTec Pelican that is equipped with a UTM-30LX laser range finder and an Asus Xtion Pro RGB-D camera using Gazebo [23].

We parameterized the quadrotor's state $\mathbf{x}$ as an 8D vector consisting of its 3D position, 3D velocity, yaw, and yaw velocity. The control input $\mathbf{u}$ is a 4D vector of the linear accelerations and yaw acceleration. We assumed the quadrotor



(a) **Ground Robot**

(b) **Aerial Robot**

Fig. 6: **Information gain performance on recorded data**: Over 157 situations for the ground robot and 63 situations for the aerial robot, we evaluate the optimality of the initial trajectory input into the trajectory optimization for various initializing methods (left column). We then evaluate the optimality of the trajectory output by trajectory optimization (right column). We see dominant modes shift towards 100% optimality when incorporating trajectory optimization, showing trajectory optimization improves overall information gain.

does not fly aggressively and that its state evolved with a constant acceleration model:

$$\mathbf{x} = \begin{bmatrix} x, & y, & z, & \theta, & \dot{x}, & \dot{y}, & \dot{z}, & \dot{\theta} \end{bmatrix}^\mathsf{T}$$

$$\mathbf{u} = \begin{bmatrix} \ddot{x}, & \ddot{y}, & \ddot{z}, & \ddot{\theta} \end{bmatrix}^\mathsf{T}$$

$$\mathbf{x}_{i+1} = \begin{bmatrix} I_4 & \Delta t \cdot I_4 \\ 0 & I_4 \end{bmatrix} \mathbf{x}_i + \begin{bmatrix} \frac{1}{2}\Delta t^2 \cdot I_4 \\ \Delta t \cdot I_4 \end{bmatrix} \mathbf{u}_i$$

where Δt is a user-defined time discretization value. For trajectory optimization, we bounded the linear acceleration between [-0.5,0.5] m/s^2, yaw acceleration between [-$\pi/2$,$\pi/2$] rad/s^2, the velocity in each dimension between [-1,1] m/s, and the yaw between [-$\pi/2$,$\pi/2$] rad/s^2. We also imposed linear constraints on the robot's motion model.

RGB-D Sensor: Both the ground and the aerial robot were equipped with a RGB-D sensor that was modeled after the Asus Xtion Pro sensor. We assume it has a minimum range of $z_{\min} = 0.5$ m, maximum range of $z_{\max} = 4.0$ m, and its noise is $\sigma = 0.03$ m. To predict future measurements and evaluate CSQMI, we discretized the 58° horizontal field of view and 48° vertical field of view into 20 separate values each, resulting in 400 separate beams.

Occupancy Grid: For both platforms, the 3D occupancy grid was constructed with a resolution of 0.1 m using data from the RGB-D sensor. In experiments, the ground robot obtained position estimates from a 2D laser based SLAM system, while in simulation the quadrotor obtained a noisy pose estimate from the simulator.

B. Evaluating Trajectory Optimization

To quantify how much trajectory optimization can improve the information gain of local motion primitives (**Q1**), we looked at its performance in a variety of situations. To create these different situations, we sampled different maps and poses for the robots from our previous real-world experimental data [5]. For each situation, we generated a fixed number of motion primitives using a lattice planner or random sampling (Sect. IV-A). The motion primitive with the highest information was then optimized. Because information gain can vary substantially across different maps and poses, we normalize the information gain of each approach by the maximum information gain achieved by any approach with the same initial conditions. Results for the ground air and robot are shown in Fig. 6a and Fig. 6b. As expected, using larger number of motion primitives improves the performance of the unoptimized trajectories. However, regardless of the number of motion primitives used to find an initialization, trajectory optimization consistently produces trajectories whose information gain is close to the maximum achieved, demonstrating that it can substantially improve upon the search based strategies. Comparing the trajectory optimization information gain for the ground robot (Fig. 6a) with the aerial robot (Fig. 6b), we see the benefit of using trajectory optimization is greater in higher-dimensional systems (**Q4**).

	Random 1	Random 8	Random 16	Random 32	Random 64	Random 128	Random 256	Lattice
Avg. Initial Time (s)	0.00 ± 0.0	0.04 ± 0.0	0.07 ± 0.1	0.15 ± 0.1	0.29 ± 0.3	0.58 ± 0.6	1.17 ± 1.2	0.26 ± 0.3
Avg. SQP Time (s)	0.78 ± 0.8	0.55 ± 0.6	0.43 ± 0.4	0.42 ± 0.4	0.40 ± 0.4	0.37 ± 0.4	0.34 ± 0.3	0.42 ± 0.4
Avg. Total Time (s)	0.79 ± 0.8	0.59 ± 0.6	0.51 ± 0.5	0.56 ± 0.6	0.69 ± 0.7	0.95 ± 1.0	1.51 ± 1.5	0.69 ± 0.7

(a) **Ground Robot**

	Random 1	Random 8	Random 16	Random 32	Random 64	Random 128	Random 256
Avg. Initial Time (s)	0.00 ± 0.0	0.02 ± 0.0	0.05 ± 0.0	0.09 ± 0.1	0.18 ± 0.2	0.37 ± 0.4	0.75 ± 0.7
Avg. SQP Time (s)	0.84 ± 0.8	0.61 ± 0.6	0.62 ± 0.6	0.60 ± 0.6	0.54 ± 0.5	0.56 ± 0.6	0.62 ± 0.6
Avg. Total Time (s)	0.84 ± 0.8	0.64 ± 0.6	0.67 ± 0.7	0.69 ± 0.7	0.73 ± 0.7	0.93 ± 0.9	1.37 ± 1.4

(b) **Aerial Robot**

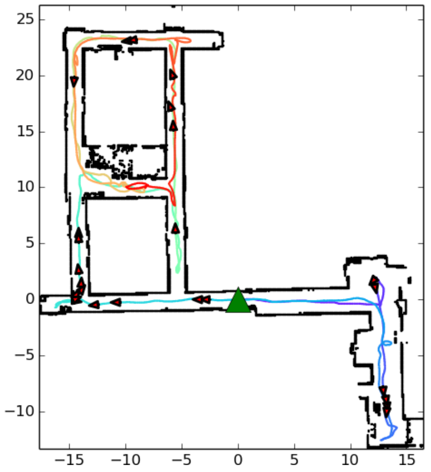
Fig. 8: **Planning times on recorded data**: Comparing the initialization, trajectory optimization, and total times for different initialization methods using the same data as in Fig. 6. The double line show when sampling controls becomes more computationally expensive than trajectory optimization.

We also examined the average computation time over all situations for each approach. Results are shown in Fig. 8a for the ground robot and Fig. 8b for the aerial robot. While evaluating a small number of motion primitives is fast, these trajectories tend to have small gains in information. However, using these trajectories to initialize optimization yields finds trajectories that are higher in information, with less time than searching over large numbers of motion primitives.

C. Mapping Experiments with a Ground Robot

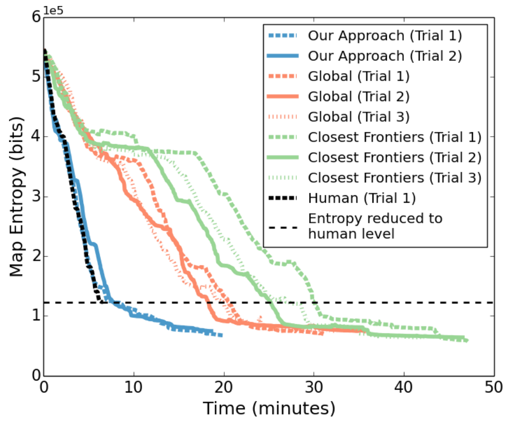
We tested our approach by using a ground robot to map a 40 m × 35 m × 2 m portion of an office environment over two trials. For the local motion primitives, we sampled 32 random trajectories. Results and comparisons are in Fig. 7. The final 3D map from one trial is shown in Fig. 1a.

To assess the overall performance of our system (**Q3**), we compare it to three other approaches. The first is "Global", a strategy that evaluates information along paths to frontiers and does not use local motion primitives or trajectory optimization, while the second is a closest frontiers planner [52]. In order to obtain an approximate lower bound on achievable performance, we also compare to a human teleoperator who knows



	Local Motion Primitives	Global Plan
Traj. Opt. Initialization (% of times)	66%	34%

(a) **Final map (2D projection), exploration path, and poses where trajectory optimization performed best**: The green triangle is the robot's start position. The colored path is its exploration path. The red arrows show the 25 poses where trajectory optimization provided the largest gain in information over the initializing trajectory.



	Averages when entropy reduced to human level		
	Time (min)	Distance (m)	Speed (m/s)
Human	6:36 (24.0%)	153 (86%)	0.41 (3.2×)
Our Approach	8:15 (29.9%)	156 (88%)	0.34 (2.6×)
Global	19:19 (71.2%)	156 (88%)	0.15 (1.2×)
Closest Frontiers	27:37 (100.0%)	178 (100%)	0.13 (1×)

(b) **Comparisons to previous work**: Comparing our approach with 1) Human teleoperator who knows the map apriori, 2) Information based planner that only plans globally [5], and 3) Closest frontiers [52].

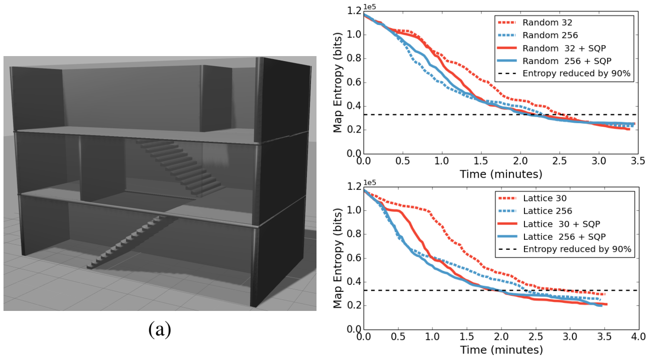
Fig. 7: **Ground robot experiments:** Mapping a real-world office environment. Our approach reduces the map's uncertainty faster than previous approaches and only does slightly worse than a human operator with a priori knowledge of the environment.

the entire environment a priori and manually drives the robot to gather measurements. For comparisons with the first two approaches, we use data from our previous experiments [5], which were obtained using the same robot.

To measure the speed of each approach, we look at the time it takes to obtain a low uncertainty map measured using Shannon's entropy. The entropy of a map with $|m|$ cells is $\sum_{i=1}^{|m|} -f_i \log_2 f_i - (1-f_i)\log_2(1-f_i)$ bits where f_i is the probability that the i^{th} cell is free [7]. Fig. 7b shows the map's entropy over time for each approach. The human operator decreased the uncertainty of the map the fastest. However, our approach did not take substantially longer to achieve a similarly certain map, despite the robot's absence of prior knowledge about the environment. Our approach was also substantially faster than the other autonomous approaches. Note that the strategies that use information and the human all travel essentially the same distance. This suggests the difference in time is primarily due to how quickly each approach can determine an informative trajectory to follow.

It is interesting that the human operator stopped once they believed they had obtained a low uncertainty map and that all autonomous approaches continue reducing the map's entropy beyond this point, as they continue until no frontiers are left. However, the final maps are qualitatively hard to differentiate, suggesting a better termination condition is needed.

Fig. 7a shows a 2D projection of the final map, the path the robot followed, and poses where trajectory optimization provided the most improvement in information. These poses are predominantly in open spaces, junctions, approaching corners, and dead-ends (**Q2**). This agrees with intuition as these areas are more complicated than straight narrow hallways where finding an informative trajectory is simpler.



(a)

(b)

(a) Stairwell with walls removed (b) Map entropy over time using random (top) and lattice (bottom) local motions with and without trajectory optimization

	Averages when entropy reduced by 90%		
	Time (min)	Distance (m)	Speed (m/s)
Random 32	2:32 (87%)	36.2 (88%)	0.21
Random 256	2:14 (77%)	30.4 (73%)	0.20
Lattice 30	2:54 (100%)	41.4 (100%)	0.18
Lattice 256	2:38 (90%)	37.5 (91%)	0.18
Random 32 + SQP	2:11 (74%)	27.0 (65%)	0.19
Random 256 + SQP	2:09 (74%)	26.9 (65%)	0.19
Lattice 30 + SQP	1:57 (67%)	26.5 (64%)	0.19
Lattice 256 + SQP	1:53 (66%)	26.4 (64%)	0.20

(c) Comparison of distance traveled and time to reduce entropy

Fig. 9: **Aerial robot stairwell simulations:** Trajectory optimization enables the quadrotor to map the environment more efficiently.

D. Mapping with an Aerial Robot

To further study the impact of trajectory optimization on the information of local primitives (**Q1**) and study how well our approach scales to robots with higher dimensional state (**Q4**), we evaluated it by having a quadrotor map a 10 m × 5 m × 7.5 m 3-story stairwell in simulation Fig. 9a. The final 3D map from one trial is shown in Fig. 1b.

We are interested in the effect of different motion primitives and whether the improvement in trajectory optimization results in different *overall* performance compared to control policies that use motion primitives and global plans. For these simulations, we generate local plans over 5 time steps and again use a random or lattice planner to generate motion primitives. The random planner repeatedly sampled from the robot's 4D control space to generate trajectories. The lattice planner generated straightline constant velocity trajectories whose directions were determined by discretizing polar and azimuthal angles in spherical coordinates. We chose this as even coarse discretizations of the control space result in thousands of trajectories, resulting in slow overall performance.

Fig. 9b shows the map's entropy over time using different numbers of initial trajectories averaged over 5 trials for each approach. The lattice planner generated 30 trajectories using 6 evenly spaced polar angles and 5 evenly spaced azimuthal angles. To generate 256 primitives, it discretized both angles into 16 distinct values. Overall, using trajectory optimization resulted in the entropy of the map decreasing faster (note that the simulator is asynchronous, so the reported difference in time includes computation time). This difference is particularly noticeable for the lattice planner, which had difficulty rounding corners due the limited number of directions it could travel.

Fig. 9c contains additional statistics about distance traveled and time to reduce entropy. We see that by using trajectory optimization – regardless of initialization strategy – the robot reduced most of the map's entropy faster, while traveling a shorter distance. This is due to the trajectory optimization's ability to find feasible trajectories that exploit the mobility of the quadrotor and execute informative turning actions (see the video submission for additional details).

VI. Limitations and Discussion

Our work has three limitations that we briefly discuss. First, we assume that the robot is capable of localizing itself reliably using a SLAM system. Fig. 10 illustrates this issue. Fig. 10a shows a 2D occupancy grid which we created by artificially lowering the maximum range of the ground robot's UTM-30LX to 4.0 m and manually driving it along a trajectory (solid blue line). The most recent laser scan (red dots) is not aligned with the map, showing accumulated position error. The robot should follow action A and close the loop to improve its state estimate and the map's consistency. However, our approach only considers the map's uncertainty and would select action B as the bottom of the map is more uncertain. Fig. 10b shows a different scenario where a robot is equipped with a short-range omnidirectional sensor. Our approach would select action B, as the robot will observe more of the map. However, during parts of this trajectory, all features will be outside the robot's FOV (red circle), precluding accurate localization; consequently action A is preferable. One avenue for future work is to extend our approach to active SLAM by accounting for robot state uncertainty [2, 19, 22, 37, 46, 50].

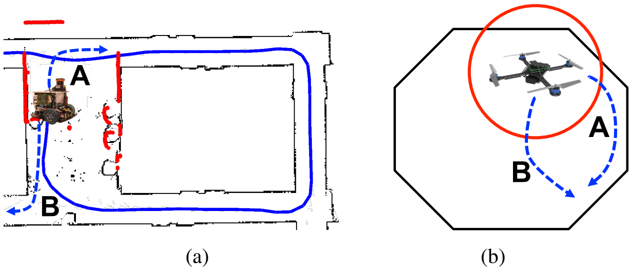


Fig. 10: **Limitations.** In (a) and (b) executing action A is preferable over action B as the robot's uncertainty will be much lower. However, our approach only considers the map's uncertainty and selects action B in both cases.

While modeling the robot's uncertainty in the control policy can be important, it is not always necessary to ensure good performance. For example, in our ground robot experiments, the accumulated state error was small. Although ground truth is not available, the final maps were consistent, meaning the corrections from the SLAM system provide insight into the robot's accumulated error. Over 8 experiments, the mean correction to the robot's 2D position and orientation was 0.28 m and 1.35°. These small corrections are due to the robot's accurate laser odometry [35] and demonstrate that it is possible to build good maps even when a robot does not explicitly take actions to reduce its own uncertainty.

Our work's second limitation is that we do not employ a principled termination condition, as is evident from the long flat tails of the entropy curves (Fig. 7b and Fig. 9b). Our experiments with the human operator provide insight into this process. It may be possible to take inspiration from prior work that uses reinforcement learning for active mapping [24] to determine appropriate termination conditions.

Finally, we use trajectory optimization for maximizing the CSQMI objective over a discrete occupancy grid. Apart from the lack of theoretical optimality guarantees, the optimization's performance also heavily depends on the initial trajectory [36].

VII. Conclusion

In this work, we present an information-theoretic planning approach for dense 3D mapping of unknown environments with mobile robots. In contrast to prior work that either use only global plans or local motion primitives, our approach combines the benefits of the two by considering a candidate set of trajectories consisting of global plans and local motion primitives. We select the most promising trajectory from this candidate set and further optimize it to increase information gain while satisfying the robot's motion model. Through a series of simulations and real world experiments, we study the benefits and drawbacks of our system and compare its end to end performance with several other approaches. Our results indicate that the proposed approach has the potential to increase the efficiency of 3D mapping, particularly for robots with limited battery life that are equipped with noisy sensors that have short sensing ranges and limited fields of view.

Acknowledgments

This work was supported in part by AFOSR grant FA9550-10-1-0567, ARL grant W911NF-08-2-0004, ONR grants N00014-07-1-0829 and N00014-09-1-1051, NSF grant IIS-1426840, and TerraSwarm, one of six centers of STARnet.

REFERENCES

[1] F. Amigoni and V. Caglioti. An Information-based Exploration Strategy for Environment Mapping with Mobile Robots. *Robotics and Autonomous Systems*, 58(5):684–699, 2010.

[2] F. Bourgault, A. Makarenko, S. Williams, B. Grocholsky, and H. Durrant-Whyte. Information-based Adaptive Robotic Exploration. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 540–545, 2002.

[3] W. Burgard, M. Moors, C. Stachniss, and F. E Schneider. Coordinated Multi-Robot Exploration. *IEEE Trans. on Robotics*, 21(3):376–386, 2005.

[4] B. Charrow, V. Kumar, and N. Michael. Approximate Representations for Multi-Robot Control Policies that Maximize Mutual Information. In *Robotics: Science and Systems (RSS)*, 2013.

[5] B. Charrow, S. Liu, V. Kumar, and N. Michael. Information-theoretic mapping using cauchy-schwarz quadratic mutual information. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2015.

[6] H. L. Choi and J. P. How. Continuous Trajectory Planning of Mobile Sensors for Informative Forecasting. *Automatica*, 46(8): 1266–1275, 2010.

[7] T. Cover and J. Thomas. *Elements of Information Theory*. John Wiley & Sons, 2012.

[8] A. Dame and E. Marchand. Mutual Information-based Visual Servoing. *IEEE Trans. on Robotics*, 27(5):958–969, 2011.

[9] A. Domahidi. FORCES: Fast optimization for real-time control on embedded systems. http://forces.ethz.ch, October 2012.

[10] C. Dornhege and A. Kleiner. A Frontier-Void-based Approach for Autonomous Exploration in 3D. *Advanced Robotics*, 27(6): 459–468, 2013.

[11] R. Fabbri, L. Costa, J. C. Torelli, and O. M. Bruno. 2D Euclidean Distance Transform Algorithms: A Comparative Survey. *ACM Computing Surveys*, 40(1):2, 2008.

[12] C. Forster, M. Pizzoli, and D. Scaramuzza. Appearance-based Active, Monocular, Dense Reconstruction for Micro Aerial Vehicle. In *Robotics: Science and Systems (RSS)*, 2014.

[13] L. Freda, G. Oriolo, and F. Vecchioli. Sensor-based Exploration for General Robotic Systems. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 2157–2164, 2008.

[14] H. H González-Banos and J-C Latombe. Navigation Strategies for Exploring Indoor Environments. *Int. Journal of Robotics Research*, 21(10-11):829–848, 2002.

[15] B. Grocholsky. *Information-Theoretic Control of Multiple Sensor Platforms*. PhD thesis, University of Sydney, Sydney, 2002.

[16] G. M. Hoffmann and C. J. Tomlin. Mobile Sensor Network Control using Mutual Information Methods and Particle Filters. *IEEE Trans. on Automatic Control*, 55(1):32–47, 2010.

[17] G. A. Hollinger, B. Englot, F. S Hover, U. Mitra, and G. S Sukhatme. Active Planning for Underwater Inspection and the Benefit of Adaptivity. *Int. Journal of Robotics Research*, 32(1): 3–18, 2013.

[18] D. Holz, N. Basilico, F. Amigoni, and S. Behnke. A Comparative Evaluation of Exploration Strategies and Heuristics to Improve Them. In *European Conf. on Mobile Robots (ECMR)*, pages 25–30, 2011.

[19] V. Indelman, L. Carlone, and F. Dellaert. Planning Under Uncertainty in the Continuous Domain: A Generalized Belief Space Approach. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2014.

[20] B. J. Julian. *Mutual Information-based Gradient-Ascent Control for Distributed Robotics*. PhD thesis, Massachusetts Institute of Technology, 2013.

[21] B. J. Julian, S. Karaman, and D. Rus. On Mutual Information-based Control of Range Sensing Robots for Mapping Applications. *Int. Journal of Robotics Research*, 33(10):1375–1392, 2014.

[22] A. Kim and R. M. Eustice. Perception-driven Navigation: Active Visual SLAM for Robotic Area Coverage. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 3196–3203, 2013.

[23] N. Koenig and A. Howard. Design and Use Paradigms for Gazebo, An Open-Source Multi-Robot Simulator. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 2149–2154, 2004.

[24] T. Kollar and N. Roy. Trajectory Optimization using Reinforcement Learning for Map Exploration. *Int. Journal of Robotics Research*, 27(2):175–196, 2008.

[25] H. Kretzschmar and C. Stachniss. Information-Theoretic Compression of Pose Graphs for Laser-based SLAM. *Int. Journal of Robotics Research*, 31(11):1219–1230, 2012.

[26] J. Leonard and H. F Durrant-Whyte. Simultaneous Map Building and Localization for an Autonomous Mobile Robot. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 1442–1447, 1991.

[27] M. Likhachev and D. Ferguson. Planning Long Dynamically Feasible Maneuvers for Autonomous Vehicles. *Int. Journal of Robotics Research*, 28(8):933–945, 2009.

[28] W. Maddern, A. Harrison, and P. Newman. Lost in Translation (and Rotation): Rapid Extrinsic Calibration for 2D and 3D LIDARs. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 3096–3102, 2012.

[29] R. Marchant and F. Ramos. Bayesian Optimisation for Informative Continuous Path Planning. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2014.

[30] S. Martínez and F. Bullo. Optimal Sensor Placement and Motion Coordination for Target Tracking. *Automatica*, 42(4):661–668, 2006.

[31] S. J. Moorehead. *Autonomous Surface Exploration for Mobile Robots*. PhD thesis, Carnegie Mellon University, 2001.

[32] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer Verlag, 1999.

[33] J. L. Ny and G. J Pappas. On Trajectory Optimization for Active Sensing in Gaussian Process Models. In *Proc. IEEE Int. Conf. on Decision and Control (CDC)*, pages 6286–6292, 2009.

[34] E. Olson. *Robust and Efficient Robotic Mapping*. PhD thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, June 2008.

[35] E. Olson. Real-time correlative scan matching. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 4387–4393. IEEE, June 2009.

[36] J. Pan, Z. Chen, and P. Abbeel. Predicting Initialization Effectiveness for Trajectory Optimization. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2014.

[37] S. Patil, Y. Duan, J. Schulman, K. Goldberg, and P. Abbeel. Gaussian Belief Space Planning with Discontinuities in Sensing Domains. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, 2014.

[38] J. C. Principe. *Information Theoretic Learning: Renyi's Entropy and Kernel Perspectives*. Springer, New York, August 2010.

[39] R. Rocha, J. Dias, and A. Carvalho. Cooperative Multi-Robot Systems: A Study of Vision-based 3D Mapping using Information Theory. *Robotics and Autonomous Systems (RAS)*, 53(3):282–311, 2005.

[40] A. Ryan and J. K. Hedrick. Particle Filter based Information-Theoretic Active Sensing. *Robotics and Autonomous Systems (RAS)*, 58(5):574–584, 2010.

[41] M. Schwager, P. Dames, D. Rus, and V. Kumar. A Multi-Robot Control Policy for Information Gathering in the Presence of Unknown Hazards. In *Int. Symp. on Robotics Research (ISRR)*, 2011.

[42] R. Shade and P. Newman. Choosing Where to Go: Complete 3D Exploration with Stereo. In *Proc. IEEE Int. Conf. Robotics*

and Automation (ICRA), pages 2806–2811, 2011.

[43] S. Shen, N. Michael, and V. Kumar. Stochastic Differential Equation-based Exploration Algorithm for Autonomous Indoor 3D Exploration with a Micro-Aerial Vehicle. *Int. Journal of Robotics Research*, 31(12):1431–1444, 2012.

[44] S. Soatto. Steps Towards a Theory of Visual Information: Active Perception, Signal-to-Symbol Conversion and the Interplay between Sensing and Control. *arXiv preprint arXiv:1110.2053*, 2011.

[45] C. Stachniss. *Robotic Mapping and Exploration*, volume 55. Springer, 2009.

[46] C. Stachniss, G. Grisetti, and W. Burgard. Information Gain-based Exploration Using Rao-Blackwellized Particle Filters. In *Robotics: Science and Systems (RSS)*, volume 2, 2005.

[47] J. Strom and E. Olson. Occupancy Grid Rasterization in Large Environments for Teams of Robots. In *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 4271–4276, 2011.

[48] S. Thrun. Robotic Mapping: A Survey. *Exploring Artificial Intelligence in the New Millennium*, pages 1–35, 2002.

[49] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2008.

[50] R. Valencia, M. Morta, J. Andrade-Cetto, and J. M Porta. Planning Reliable Paths with Pose SLAM. *IEEE Trans. on Robotics*, 29(4):1050–1059, 2013.

[51] P. Whaite and F. P Ferrie. Autonomous Exploration: Driven by Uncertainty. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 19(3):193–205, 1997.

[52] B. Yamauchi. A Frontier-based Approach for Autonomous Exploration. In *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pages 146–151, July 1997.