

# Communication-Efficient Planning and Mapping for Multi-Robot Exploration in Large Environments

Micah Corah\*, Cormac O’Meadhra\*, Kshitij Goel, and Nathan Michael

**Abstract**—This work presents a framework for planning and perception for multi-robot exploration in large and unstructured 3D environments. We employ a Gaussian mixture model for global mapping to model complex environment geometries while maintaining a small memory footprint which enables distributed operation with a low volume of communication. We then generate a local occupancy grid for use in planning from the GMM using Monte Carlo ray tracing. Then, a finite-horizon, information-based planner uses this local map and optimizes sequences of observations locally while accounting for the global distribution of information in the robot state space which we model using a library of informative views. Simulation results demonstrate that the proposed system is able to maintain efficiency and completeness in exploration while only requiring a low rate of communication.

**Index Terms**—Aerial Systems: Perception and Autonomy, Mapping, Networked Robots, Robotic Exploration

## I. INTRODUCTION

RAPID exploration is critical in scenarios such as disaster response and search-and-rescue. Teams of aerial robots equipped with depth cameras can coordinate to rapidly explore complex 3D environments and can maintain efficiency even for deployments of large numbers compared to environment and sensor scales [1]. However, large environments and high volumes of sensor data pose challenges for communication and computation that limit the number of robots that can be effectively deployed for exploration tasks. Additionally, unstructured and uncertain environments pose challenges for both perception and planning as existing approaches often rely on dense approximations [2] and heuristics [3] that can be inappropriate for large and finely detailed environments.

As illustrated in Fig. 1, this work leverages techniques from information theory in both planning and perception to address challenges related to scale and complexity. The perception system uses a Gaussian mixture model (GMM) that accurately represents detailed geometry and efficiently captures empty volumes and surfaces [4, 5]. The resulting GMM has a small memory footprint and communicating updates across a network of robots requires relatively little bandwidth compared

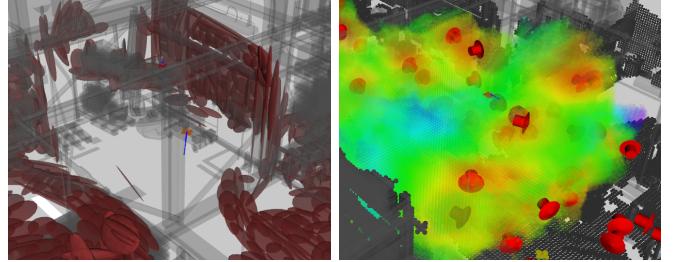
Manuscript received: September, 11, 2018; Revised December, 8, 2018; Accepted January, 16, 2019.

This paper was recommended for publication by Editor Jonathan Roberts upon evaluation of the Associate Editor and Reviewers’ comments. This work was supported by DOE (DE-EM0004067), DTRA (HDTRA1-13-0026), and industry.

The authors are affiliated with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA. {micahcorah, comeadhra, kshitij, nmichael}@cmu.edu

\*Authors contributed equally.

Digital Object Identifier (DOI): see top of this page.



(a) Gaussian mixture mapping (b) View library with shortest-path distance field

**Fig. 1:** In the proposed approach, lightweight data structures enable effective multi-robot exploration. (a) A Gaussian mixture model (red ellipsoids) serves as a global spatial representation of the environment geometry in the distributed mapping system. (b) A library of informative views (red arrows) represent the distribution of views of unmapped space. The receding-horizon planner then minimizes shortest-path distances (rainbow) to the views which draw the robot toward information-rich regions of the environment while the planner simultaneously maximizes information gain with respect to the partially mapped environment.

to the volume of novel data. Samples from the GMM are, in turn, used to maintain a dense local map for use in planning. A receding-horizon planner maximizes information gain over sequences of camera views [6], and a terminal cost based on distances to highly informative views provides global spatial reasoning and ensures complete exploration. Robots maintain a library of such views by sampling and updating views locally and share updates with the rest of the team.

## II. RELATED WORK

Most works in robotic exploration use occupancy grid models [3, 6, 7, 8] for mapping while some employ alternative representations such as Gaussian processes [9, 10]. Recently, Gaussian mixture models (GMMs) have been proposed for use in modeling point clouds [11] and for mapping applications [4, 5]. This work applies GMMs to multi-robot exploration and addresses challenges such as volumes of sensor data that exceed communication constraints and large memory footprints that arise with increasing spatial scales. Further, we reconstruct a dense, local occupancy grid from the global GMM and propose an exploration approach for this setting where only a local map is available.

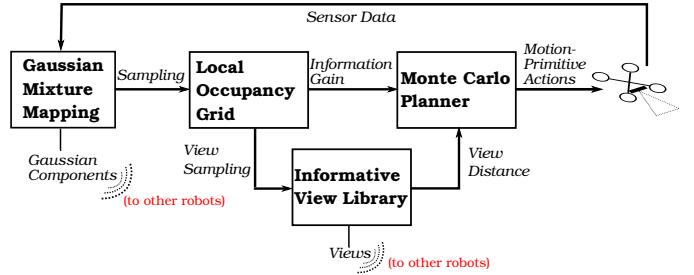
Reducing communication requirements for collaboration has also been considered in the context of distributed inter-robot localization. Techniques have been developed that enable efficient look up of overlapping measurements [12] and reduced number of measurements necessary for loop-closure [13]. Such approaches involve measurement sharing

and so are complementary to this work. Communication requirements can be reduced through the use of inference systems capable of encoding environmental information in compact latent spaces. However, inference in 3D is difficult and 2D inference approaches [14] still benefit from sharing information.

A robot that has mapped a local region of the environment may have to travel a significant distance across previously mapped regions to reach more unmapped space. For this reason, techniques that model the distribution of informative observations in the environment or state space are central to robotic exploration. Yamauchi [3] called the boundary between unknown and known unoccupied space the *frontier* and proposed that the robot should navigate toward the nearest point on the frontier. In general, frontiers can be inaccessible or insignificant as the concept does not account for the robot's reachable space or sensor model. View-based methods reason in the robot's state space and often attempt to quantify the information gain from future observations [6, 8, 15]. Because a robot in unoccupied space observes unknown space via the frontier, highly informative views are generally near frontiers, and view-based methods often explicitly evaluate views on or near frontiers [6, 7]. Our work relies on a different principle: the information gain for hypothetical observations changes only near robots and their observations. We take advantage of this to maintain a global model of information gain while using only local updates. Witting et al. [16] take advantage of a similar principle by maintaining a graph connecting states near those that have already been visited to enable efficient search for distant yet informative views. In this work, we emphasize significantly informative views (called *informative views*) which form a superlevel set in the state space and sample and update views in the local map as robots move through the environment. By doing so, the robots maintain spatially global models of the distribution of information by communicating local updates to nearby views to the rest of the team.

Robots gain the ability to explore by application of appropriate planning techniques to the selected models of the environment and information content. One common approach is to select and navigate toward a frontier or view based on criteria such as distance and information content [3, 7, 17]. However, reasoning about the joint contributions of sequences of observations can improve performance [6, 18, 19, 20]. While some formulations use a variable time horizon and allow trajectories to cover the entire spatial extents of the environment [6, 18, 20], this work uses a fixed horizon [1, 19] to maintain compatibility with the local mapping approach. A terminal cost based on the shortest path distance to any known informative view ensures complete exploration of large environments when planning over a finite horizon. Distances are precomputed using a variant of the fast marching method as has been used previously in planning [21] and exploration [22]. Further, using views as sources allows for implicit rather than explicit [18] selection of goals during optimization.

Although we do not apply multi-robot coordination in planning, our proposed approach can be readily extended with coordination techniques at the level of the finite-horizon plan-



**Fig. 2:** A team of identical robots use *sensor data* from depth cameras to update a global **Gaussian mixture model** (GMM) while *sharing Gaussian components*. Robots then *sample points* from the GMM to produce a **local occupancy grid**. They then *sample views* locally and *share views* with the other robots to maintain an **informative view library** that represents the spatial distribution of information in the state space. The **Monte Carlo tree search planner** seeks to maximize *information gain* while minimizing *distances to informative views* and outputs the *motion-primitive actions* that the robots track.

ner as in our prior work [1] which uses sequential planning. Incorporating other common techniques for coordination would nominally involve more significant changes as these generally operate over potentially long-term goals via assignment [7, 23] or routing [24] and require global spatial representations of the geometry and information content.

### III. OVERVIEW OF THE MULTI-ROBOT EXPLORATION AND MAPPING SYSTEM

Consider a team of  $n_r$  identical multirotor robots. Each individual robot is equipped with: a forward-facing depth camera by which it can observe the surrounding environment; an onboard processor that can process sensor data for mapping and plan future exploration actions; and a communications apparatus—which may be bandwidth limited—that it can use to communicate data pertinent to planning and mapping to the rest of the multi-robot team. *This work seeks to develop a system for multi-robot exploration to enable the described robots to explore an unknown environment rapidly and completely.*

Figure 2 shows an overview of our proposed approach. The components of this system fall under two categories: mapping (Sec. IV) and planning for exploration (Sec. V). The Gaussian mixture map and informative view library, respectively, form global spatial representations of the geometry and distribution of information in the environment. In both cases, robots share their local updates with the rest of the team, and by incorporating these updates, each robot maintains a global model. Additionally, as discussed in the representative sections, we design these global representations to minimize memory requirements for communication and storage. The remaining two system components are spatially local: the sliding occupancy grid provides a dense, local representation of the environment geometry, and the Monte Carlo tree search planner seeks to maximize information gain with respect to the occupancy grid while minimizing distances to views in the view library, locally, over a finite-horizon.

### IV. MAPPING

When performing multi-robot exploration with high density sensors such as RGB-D cameras, network utilization can

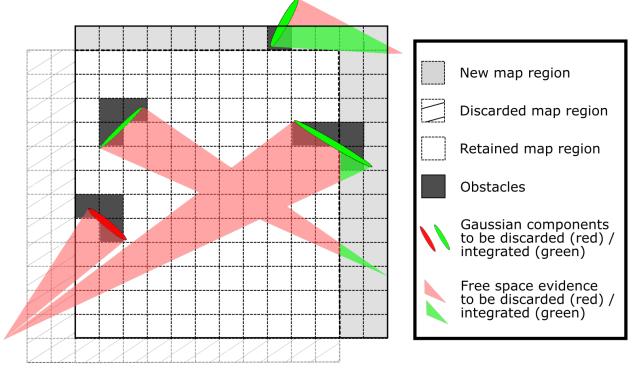
quickly saturate due to the high volume of data. In this work, Gaussian mixture models (GMMs) are employed as a high fidelity generative model of sensor measurements. Incoming sensor measurements are modeled as GMMs with a known sensor location (Section IV-A) and the compressed model is shared across the multi-robot team (Section IV-C). The reduced memory complexity of GMMs is leveraged to enable exploration in large environments where the maintenance of a high fidelity volumetric occupancy is intractable. Exploration requires a volumetric representation of the probability of occupancy, which is achieved through a sliding voxel map representation, recovered through Monte Carlo ray tracing of points resampled from the GMM (Section IV-B). Finally, the local occupancy representation is leveraged to discard low utility points (Section IV-B2).

### A. Gaussian Mixture Modeling

A point cloud  $\mathcal{Z}$  is modeled as a set of *i.i.d.* samples drawn from an underlying density function in  $\mathbb{R}^3$ , which is assumed to be well modeled by a Gaussian mixture model (GMM). A GMM,  $\mathcal{G}$ , is a weighted mixture of Gaussian components, parameterized by  $\{\pi_j, \mu_j, \Sigma_j\}$ , which are the mixing parameters, component means and covariances. Specifically, a GMM is a density function which describes the likelihood of a point  $z$  as  $\mathcal{G}(z) = \sum_{j=1}^J \pi_j \mathcal{N}(z|\mu_j, \Sigma_j)$ . In this application,  $\mathcal{G}$  is learned via the expectation-maximization (EM) algorithm. For details on the specifics of EM, the reader is directed to [25, Ch. 9]. Free space information is incorporated into the model by storing the sensor location,  $s$ , from which  $\mathcal{Z}$  was observed. For each point in  $\mathcal{Z}$ , the free space information is given by the beam connecting the point to  $s$ . The number of mixtures components,  $J$ , controls a trade-off between model fidelity and complexity. Approaches for selecting  $J$  include a range of information criteria [26] and hierarchical strategies with stopping heuristics [11]. In this work, a density based heuristic was used to achieve a minimum compression of the input point cloud. Specifically,  $J = |\mathcal{Z}|/R_c$ , where the  $R_c$  is a parameter that controls the point-to-component ratio. Setting  $R_c$  to 160 was found to yield good performance.

### B. Sliding Voxel Map

A voxel map is used to encode a volumetric representation of the probability of occupancy in the environment. The cost of maintaining a voxel map scales with the volume of the environment of interest. Each robot maintains an egocentric sliding voxel map, with bounding box  $\mathcal{B}$  centered on the robot's location, to overcome this scaling issue. The map is derived from a local GMM,  $\mathcal{G}_L$ , which is generated using a relevant subset of components from the global GMM based on  $\mathcal{B}$ . The free and occupied space evidence are extracted from  $\mathcal{G}_L$  using a Monte Carlo ray tracing technique [5], detailed in Section IV-B1. The probability of occupancy for an individual voxel is then found by combining the occupied and free space evidence as is standard in occupancy grid mapping, see O'Meadhra *et al.* for more information. The local map is used to identify the novel subset of the incoming measurement as detailed in Section IV-B2. The sliding voxel map is updated



**Fig. 3:** Sliding grid map and resampled ray trimming. After robot motion, the hatched region becomes out of bounds and is mapped to the new region, shown in gray. Beams sampled from the GMM are intersected with the new map region to prevent double counting their contributions to occupancy probability. The trimmed portions of the resampled beams are shown in red and the portions used to update the map are shown in green.

each time a depth measurement is received. The high frame rate of common depth sensors limits motion between sensor measurements. Therefore, the new voxel map after updating the robot location typically contains a substantial portion of the previous voxel map. The cost associated with memory copies is avoided by implementing the sliding map using a circular buffer, see Fig 3.

**1) Monte Carlo ray tracing:** Conversion between the GMM and the voxel map is achieved via Monte Carlo ray tracing [5]. In principle, the GMM provides an analytic representation of both the occupied and free space evidence in each measurement. However, in  $\mathbb{R}^3$ , it is not possible to extract this evidence for a given volume (e.g. a voxel) in closed form. Accordingly, Monte Carlo sampling and ray tracing is employed to extract the occupied and free space evidence from the GMM. This amounts to drawing samples  $\mathcal{Z}_R$  from  $\mathcal{G}_L$  and performing ray tracing with  $\mathcal{Z}_R$ .

As the bounding box tracks the moving robot, only the new section of the map (gray region in Fig. 3) needs to be updated using beams sampled from the GMM—whereas new sensor data is integrated into the full local map. The segments used to update the map (green beams in Fig. 3) are identified by intersecting sampled beams against the extent of the new map region (grey region in Fig. 3).

**2) Point novelty check:** In typical robotic mapping applications, only a subset of each measurement will be novel. In order to increase the time and memory efficiency of the model, points below a predefined novelty threshold are discarded. There exist many strategies for computing point-wise novelty [15]. In this application, the novelty of a point is defined as the entropy reduction achieved by the ray from the sensor location to the point. To increase efficiency, beams are considered independent, which yields an overestimate of the entropy reduction. The entropy reduction is computed for point  $z$  by ray tracing beam  $r$  from the sensor location  $s$  to  $z$  through the local occupancy map to identify the voxel set  $V$ . The cell-wise entropy change is then  $E = \sum_{v \in V} (E_v^+ - E_v^-)$ , where superscript denotes before and after the update from ray  $r$  and

$$E_v = -p(v) \log(p(v)).$$

3) *Component updating*: The robustness of the occupancy map to measurement noise is increased by requiring several agreeing measurements to drive a voxel to an occupied or free state. As a result, beams that are considered novel may have been previously observed. In order to reduce the complexity of the model, the GMM is queried to determine whether existing components well-model the novel beams.

Points that do not contribute free space information can be checked directly against the GMM components. Specifically, for each novel point  $z_n$  and each candidate component  $k$ , the Mahalanobis distance is computed,  $d_{n,k} = (z_n - \mu_k)^T \Sigma_k^{-1} (z_n - \mu_k)$ . The distance  $d_{n,k}$  is compared against a threshold  $d_{ll}$ . Each component with a distance below the threshold is updated with the fractional point  $d_{n,k}^{-1} / \sum_{d_{n,i} < d_{ll}} d_{n,i}^{-1}$ .

In the case that the beam contributes free space information, the set of components to be updated is further constrained to satisfy  $\|s_k - s\|_2 < d_s$ , i.e. that the current sensor location  $s$  is near the sensor location  $s_k$  from which the component was observed. Components are updated by modifying their support size (the number of points modeled by a component) by the number of matched points. As component shapes are not updated, component accuracy can not be increased. However, inaccuracies in the model will contradict sensor observations leading to large numbers of novel points and accumulation of new components until the fidelity stabilizes. While iterative EM approaches exist for updating components, changing the shape of the component would make global GMM synchronization more difficult and is left as future work.

### C. Distributed Mapping

The principle motivation for the use of GMMs is their substantial reduction in complexity compared to raw point clouds. Leveraging this property enables a simple distributed mapping framework whereby each robot shares all novel components and all component updates computed after each new local depth measurement is received. The result of this paradigm is that, barring communication failures, each robot maintains a complete global representation of the environment.

Novel data is shared in two forms. In the case of previously unseen components, the message consists of six fields: a unique ID, the number of points supporting the component, sensor location, component mean, component covariance, and a flag indicating whether the points are within the maximum sensor range. However, only the ID and the new number of support points are shared to update an existing component.

## V. PLANNING FOR EXPLORATION

The exploration approach consists of two subsystems: a local planner that seeks to maximize mutual information and a library of views that models the distribution of information in the environment.

The robot system model consists of states  $x \in \mathcal{X}$ , observations  $y$ , and control inputs  $u$  according to a generic model,

$$x_t = f(x_{t-1}, u_t) \quad y_t = h(x_t). \quad (1)$$

A finite-horizon trajectory over time 1 through  $T$  has associated sequences of states  $X = (x_1, \dots, x_T)$ , control inputs  $U = (u_1, \dots, u_T)$ , and observations  $Y = (y_1, \dots, y_T)$ . The observations are also a function of the occupancy grid environment, represented by the random variable  $M$  and approximated by the local sliding grid described in Sec. IV-B.

We use a kinematic quadrotor model for planning with a state space consisting of yaw and translation  $\mathcal{X} \subseteq \mathbb{R}^3 \times \mathbb{S}$ . Uncertainty in the environment induces disjoint safe  $\mathcal{X}_{\text{safe}}$  and unsafe  $\mathcal{X}_{\text{unsafe}}$  subsets of the state-space which are computed from truncated distances to unknown and occupied voxels in the local map given a fixed collision radius.

### A. Information-Theoretic Objective

The proposed objective consists of two components: a local reward for information gain  $R_{\text{inf}}(\cdot)$ , and a global reward on the terminal state  $R_{\text{dist}}(\cdot)$  for decreasing distance to informative views in the view library, which leads to the following optimization problem

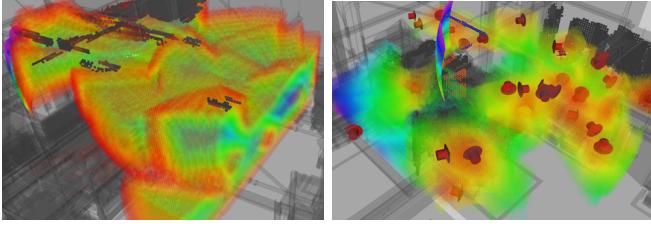
$$\begin{aligned} \max_U R_{\text{inf}}(X) + \alpha R_{\text{dist}}(x_T) \\ \text{s.t. } x_t \in \mathcal{X}_{\text{safe}} \quad \forall t \in \{1, \dots, T\} \end{aligned} \quad (2)$$

whereas  $R_{\text{inf}}(X) = I(Y; M)$  and  $R_{\text{dist}}(x) = \max(d(x_0) - d(x), 0)$ . Here,  $I(\cdot; \cdot)$  represents the chosen measure of information gain between observations and the map, specifically the Cauchy-Schwarz quadratic mutual information [6, 27]. Further,  $d(x)$  is the shortest-path distance to the nearest state in  $\mathcal{X}_{\text{safe}}$  having information reward of at least a minimum value—as will be defined in more detail, later. Thus, the two components of the objective serve to maximize information gain and to draw the robot toward information-rich regions of the environment; the coefficient  $\alpha \geq 0$  controls the trade-off between these two behaviors.

The key properties of the information objective for this work are the incorporation of camera geometry (via ray-tracing) and the ability to represent the joint contribution of sequences of observations where some sensor data can be redundant. Other objectives that have been proposed for use in exploration and active perception [8, 15] can be substituted into the proposed exploration framework, given slight modification, if desired.

### B. Receding-Horizon Planning with Monte Carlo Tree Search

The objective (2) is a non-convex function of sequences of states and observations. To address this, we extend the approach described in our prior work [1] which uses a sampling-based planner, Monte Carlo tree search (MCTS) [28], which is frequently used in active perception and exploration [19, 29]. Here, the actions available to the robot at each time-step correspond to polynomial trajectories from a library of motion-primitives [30]. Repeated concatenation of these motion-primitives forms a tree. MCTS samples finite-horizon plans—the concatenated sequences of motion-primitives—through a combination of informed search, seeking to maximize reward, and random sampling. MCTS evaluates the objective of (2) once per each such finite-horizon sample and, by doing so, learns the distribution of reward in the tree.



(a) Frontier distance      (b) View library and distance

**Fig. 4:** This figure examines frontier and view distances with red signifying least and blue greatest distances. (a) Frontier distance accounts for locations of unobserved parts of the map but not for the sensor model. Frontier distances are strongly influenced by frontier voxels associated with relatively uninformative camera views such as those at the top or bottom of known free space (shown in the figure) which are difficult to observe using a forward-facing camera. (b) The view library and distance directly account for camera views and expected information gain and more accurately represent how best to observe unexplored regions of the environment.

#### Algorithm 1 Update view values and sample views

```

1: Input:  $\hat{\mathcal{X}}_{\text{inf}}$   $\triangleright$  informative views,  $\hat{R}_{\text{inf}}$   $\triangleright$  pre-computed
   values
2:  $\triangleright$  Filter views below a threshold ( $\epsilon_{\text{view}}$ )
3: for  $x \in \hat{\mathcal{X}}_{\text{inf}}$  do
4:   if  $\hat{R}_{\text{inf}}(x) < \epsilon_{\text{view}}$  then
5:      $\hat{\mathcal{X}}_{\text{inf}} \leftarrow \hat{\mathcal{X}}_{\text{inf}} \setminus \{x\}$ 
6:  $\triangleright$  Update values for nearby views in the library
7:  $\hat{\mathcal{X}}_{\text{near}} \leftarrow$  subset of  $\hat{\mathcal{X}}_{\text{inf}}$  wholly contained in the local map
8: for  $x \in \hat{\mathcal{X}}_{\text{near}}$  do
9:    $\hat{R}_{\text{inf}}(x) \leftarrow R_{\text{inf}}(x)$ 
10:  $\triangleright$  Sample new candidate views
11: for  $|\hat{\mathcal{X}}_{\text{near}}| \dots n_{\text{budget}}$  do  $\triangleright$  Limit total evaluations
12:    $x \leftarrow$  sample uniformly from the subset of  $\mathcal{X}_{\text{safe}}$  with
      views wholly contained in the local map
13:    $\triangleright$  Check membership in  $\hat{\mathcal{X}}_{\text{inf}}$  and require novelty
14:   if  $R_{\text{inf}}(x) \geq \epsilon_{\text{view}}$  then
15:     if  $R_{\text{inf}}(\{x\} \cup \hat{\mathcal{X}}_{\text{near}}) - R_{\text{inf}}(\hat{\mathcal{X}}_{\text{near}}) \geq \epsilon_{\text{novelty}}$  then
16:        $\hat{\mathcal{X}}_{\text{inf}} \leftarrow \hat{\mathcal{X}}_{\text{inf}} \cup \{x\}$ ,  $\hat{R}_{\text{inf}}(x) \leftarrow R_{\text{inf}}(x)$ 
```

Further, the planning horizon has a fixed length, and the last action in the plan is constrained to terminate with zero velocity. The latter condition guarantees safety by requiring that the robot is able to stop within  $\mathcal{X}_{\text{safe}}$ . The plans are executed in a receding-horizon fashion and are continually updated as the robot replans. If planning fails, the robot may continue to execute the current plan, indefinitely, until a new plan is found.

#### C. Modeling Travel Distances

The receding-horizon planner reasons about sequences of actions and observations locally in space and time. The view library provides the spatially global representation of the distribution of information from which we approximate the distance reward on the terminal state in the objective of (2). Conceptually, the view library seeks to model a superlevel set

of  $\mathcal{X}_{\text{safe}}$  associated with sufficiently informative observations according to a threshold  $\epsilon_{\text{view}} > 0$  such that

$$\mathcal{X}_{\text{inf}} = \{x \mid R_{\text{inf}}(X) \geq \epsilon_{\text{view}}, x \in \mathcal{X}_{\text{safe}}\}. \quad (3)$$

Rather than explicitly selecting a point in  $\mathcal{X}_{\text{inf}}$  the objective seeks to reduce the distance to *any* point in  $\mathcal{X}_{\text{inf}}$ . The minimum travel distance according to this planning problem can be represented using a differential equation, called an Eikonal equation, with boundary values on  $\mathcal{X}_{\text{inf}}$  so that

$$\begin{aligned} |\nabla d(x)| &= 1 \quad \forall x \in \mathcal{X}_{\text{safe}} \setminus \mathcal{X}_{\text{inf}}, \quad |\nabla d(x)| = 0 \quad \forall x \in \mathcal{X}_{\text{unsafe}}, \\ d(x) &= 0 \quad \forall x \in \mathcal{X}_{\text{inf}}. \end{aligned} \quad (4)$$

Such equations can be solved efficiently in robotics applications using variants of the fast marching method [21] which can be thought of as a generalization of Dijkstra's algorithm.

The fast marching solver pre-computes values of  $d(\cdot)$  over  $\mathcal{X}_{\text{safe}}$  or a subset thereof so that the MCTS planner (Sec. V-B) can efficiently look up the value of  $d(\cdot)$  at the end of the planning horizon to compute the distance reward  $R_{\text{dist}}(\cdot)$ .

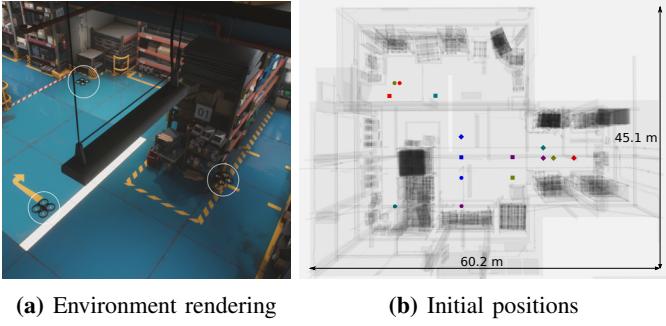
However, challenges related to computation of  $d(\cdot)$  require some approximations. Dense reconstruction of  $\mathcal{X}_{\text{inf}}$  is intractable due to evolving belief in environment occupancy, large spatial scale, and cost of evaluation of mutual information. Instead,  $\mathcal{X}_{\text{inf}}$  is approximated by a subset, the library of informative views, as discussed later. Further, as described in Sec. IV-B, occupancy probability is only evaluated densely in the local map. Therefore, rather than solving for  $d(\cdot)$  globally, we solve over local map and approximate values of  $d(\cdot)$  on the boundaries of the local map using the scaled Euclidean distances to the nearest view in the view library outside of the local map<sup>1</sup>. Last, the fast marching solver uses the  $\ell_1$  norm for  $|\nabla d(x)|$  in (4) for simple computation and implementation.

Figure 4 illustrates distances computed using the proposed approach and, analogously, using frontiers. As shown, frontiers on their own—which we compare to in the simulation results—do not exclusively capture regions of the state space associated with significant information gain and instead include small, isolated clusters of frontier voxels and frontier voxels at the top and bottom of the free space that are difficult to observe using forward-facing cameras.

#### D. Modeling Distribution of Information with a View Library

The view library uses a set of sampled views  $\hat{\mathcal{X}}_{\text{inf}}$  to approximate  $\mathcal{X}_{\text{inf}}$  (described in (3)). Algorithm 1 details the approach used to update the library and sample new views and consists of three basic components. Beginning on line 2, views whose values have decreased below  $\epsilon_{\text{view}}$  are removed from the view library. Starting on line 6, values for views contained in the local map are updated and stored in  $\hat{R}_{\text{inf}}$ . Most importantly, new views are sampled in the local map starting on line 10, and the number of information gain evaluations, including updates,

<sup>1</sup>This heuristic works well for the environment studied in this work which is not “maze-like” despite being expansive and cluttered. A more general approach could maintain a global roadmap with connections to views in the view library to provide an upper-bound for  $d(\cdot)$  at the boundary of the local map.



**Fig. 5:** (a) Three robots (circled in white) explore in the 3D warehouse environment and (b) in each of the five trials—shown in different colors—start from different initial positions in order to ensure a variety of initial conditions.

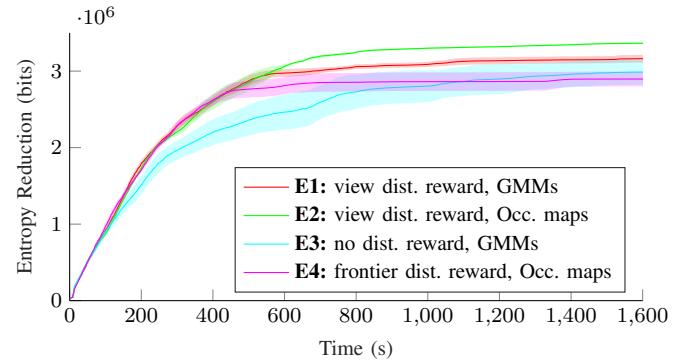
is limited to  $n_{\text{budget}}$ . Sampled views are added if they belong to  $\mathcal{X}_{\text{inf}}$  and satisfy a novelty check which requires new views to provide additional information gain of at least  $\epsilon_{\text{novelty}}$  with respect to other nearby views. The novelty check limits the growth of the view library based on how the views represent information content in the map.

Additionally, robots communicate changes to  $\hat{\mathcal{X}}_{\text{inf}}$  and  $\hat{R}_{\text{inf}}$  for views added or updated during execution of Alg. 1. When updating a view, robots transmit the associated pose and information gain along with a unique identifier created by the originating robot. This ensures that each robot can update views created by any other. Note that information gain for views does not vary across robots because the robots are identical.

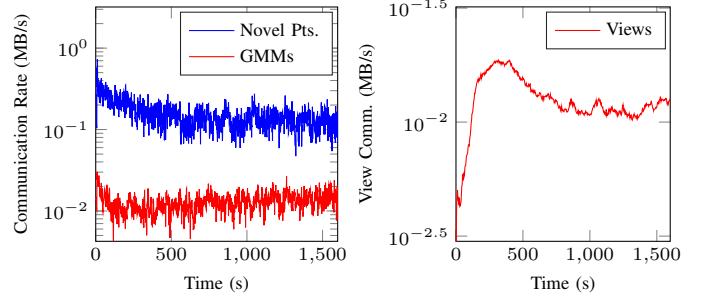
## VI. EXPERIMENT DESIGN

We evaluate exploration performance using reduction in Shannon entropy of the map [31] which quantifies uncertainty. Average times to reach levels of entropy reduction capture variations in initial performance, performance in the partially explored environment, and completeness of exploration. To ensure fair comparisons despite varying environment representations, we report entropy reduction from a separate occupancy grid, and experiments are run slower than real time to reject effects of overheads of different mapping configurations. Evaluation of communication rates is based the total quantities of data produced by the multi-robot team.

Simulation experiments are conducted in a high fidelity 3D warehouse environment (Fig. 5) with a team of three quadrotors equipped with forward facing depth cameras. Robots begin exploring near the ground for each of five trials with varied initial positions (Fig. 5) and continue for 1600 seconds. We conducted experiments with the following configurations: **E1**: View distance reward with the proposed GMM mapping approach, **E2**: View distance reward with a local submap from a global occupancy grid derived from all sensor data, **E3**: Distance rewards disabled with the proposed GMM mapping approach, and **E4**: Frontier distance reward and planning using a global occupancy map. Each configuration also uses the local information reward, even when distance rewards vary. The experiments validate the combined system (**E1**), compare to standard occupancy grid mapping (**E2**) and frontier



**Fig. 6:** Entropy reduction variation over time for the simulation experiments. Shaded regions show standard error.



**Fig. 7:** (Left) Communication rates for the proposed approach (Red) which uses GMMs are substantially less than if communicating novel points (Blue). Note that channel properties are assumed to be the same for each approach and accordingly, communication rate is equal to the rate of data produced across the team. (Right) View transmission requires a similar communication bandwidth to the GMM component messages.

exploration (**E4**), and establish the significance of distance rewards in the exploration scenario (**E3**).

## VII. RESULTS

Figure 6 shows entropy reduction results for each experiment, and Tab. I expands on time taken for various levels of completion. Note that these results account for differences in mapping implicitly through differences in exploration actions and by evaluating entropy reduction on a separate occupancy grid. We observe largely similar performance in terms of entropy reduction while using the view distance reward and varying the underlying mapper (**E1** and **E2**) except for slightly reduced entropy reduction for exploration with the GMM mapper. The variance (rendered as standard error) in entropy reduction remains low during each of these two experiments so we suggest that the difference is a matter of disagreement

**TABLE I:** Time time taken (sec.) for mean entropy reduction to reach different levels with respect to the maximum mean entropy reduction (Fig. 6, **E2**). Final completion percentage replaces time for experiments that do not reach a given level of completion (**E3**, **E4**).

Percent Explored/Experiment	10%	30%	60%	90%
View dist., GMMs ( <b>E1</b> )	31.9	<b>111.3</b>	<b>242.6</b>	743.4
View dist., Occ. Maps ( <b>E2</b> )	32.1	116.1	244.5	<b>583.5</b>
No dist., GMMs ( <b>E3</b> )	<b>30.2</b>	129.2	323.1	88%
Frontier dist., Occ. Maps ( <b>E4</b> )	33.1	113.3	262.2	86%

between the GMM map and the occupancy grid used to report entropy reduction rather than a matter of the representation negatively affecting performance. However, when exploring with frontiers and occupancy grids (**E4**), the steady final entropy reduction only reaches 86% of the maximum mean value over all trials (see Table I) with significantly increased variance. This indicates that failure to accurately model the distribution of information, such as due to unobservable frontiers, becomes significant when exploration is nearly complete. When view distance rewards are disabled (**E3**), the exploration performance degrades in comparison to the scenarios when distance rewards are taken into account (**E1**, **E2**, **E4**). This establishes the significance of the distance reward and spatially global reasoning in the experimental trials.

Figure 7 examines communication rates for **E1**, comparing communication costs for sending GMMs to novel points (introduced in Sec. IV-B2), and presents communication costs for view sharing. The communication cost for each approach is based on the total production of data by the multi-robot team and does not model messaging protocols or lossy transmission. Together, Fig. 7 and Fig. 6 demonstrate that GMM mapping produces substantially less data compared to communicating novel points while achieving similar (albeit slightly lower) entropy reduction in the model compared to occupancy mapping. Additionally, the cost of communicating novel points already represents a significant reduction over complete point clouds generated by the depth cameras. The additional cost for communicating views is comparable to that of the mapping system and does not dominate the total cost. Overall, the proposed approach provides a shared model of occupancy, a distribution of information that is suitable for exploration, and for a team of three robots, generates only tens of kilobytes of information per second.

### VIII. CONCLUSION

This work has presented a framework that enables mapping and exploration with teams of robots in three-dimensional environments at large spatial scales. The planning and mapping approaches take advantage of lightweight representations at the global spatial scale by representing geometry using a GMM and the distribution of information using sampled informative views. Robots explore using a finite-horizon planner with an information-theoretic objective that uses local, dense representations of occupancy and distances to views. Simulation results in a warehouse environment demonstrate that the system is able to provide complete and efficient exploration while maintaining low rates of communication.

### REFERENCES

- [1] M. Corah and N. Michael, “Distributed matroid-constrained submodular maximization for multi-robot exploration: theory and practice,” *Auton. Robots*, 2018.
- [2] A. Elfes, “Using occupancy grids for mobile robot perception and navigation,” *IEEE Computer Society*, vol. 22, no. 6, pp. 46–57, 1989.
- [3] B. Yamauchi, “A frontier-based approach for autonomous exploration,” in *Proc. of the Intl. Sym. on Comput. Intell. in Robot. and Autom.*, Monterey, CA, Jul. 1997.
- [4] S. Srivastava and N. Michael, “Approximate continuous belief distributions for precise autonomous inspection,” in *IEEE Intl. Symp. on Safety, Security, and Rescue Robotics*, Lausanne, Switzerland, Dec. 2016.
- [5] C. O’Meadhra, W. Tabib, and N. Michael, “Variable resolution occupancy mapping using Gaussian mixture models,” *IEEE Robot. Autom. Letters*, 2018.
- [6] B. Charrow, S. Liu, V. Kumar, and N. Michael, “Information-theoretic mapping using Cauchy-Schwarz quadratic mutual information,” in *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, Seattle, WA, May 2015.
- [7] W. Burgard, M. Moors, C. Stachniss, and F. E. Schneider, “Coordinated multi-robot exploration,” *IEEE Trans. Robotics*, vol. 21, no. 3, pp. 376–386, 2005.
- [8] B. J. Julian, S. Karaman, and D. Rus, “On mutual information-based control of range sensing robots for mapping applications,” *Intl. Journal of Robotics Research*, vol. 33, no. 10, pp. 1357–1392, 2014.
- [9] K. Yang, S. Keat Gan, and S. Sukkarieh, “A Gaussian process-based RRT planner for the exploration of an unknown and cluttered environment with a UAV,” *Advanced Robotics*, vol. 27, no. 6, pp. 431–443, 2013.
- [10] M. G. Jadidi, J. V. Miro, and G. Dissanayake, “Mutual information-based exploration on continuous occupancy maps,” in *Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst.*, Hamburg, Germany, Sep. 2015.
- [11] B. Eckart, K. Kim, A. J. Troccoli, A. Kelly, and J. Kautz, “Accelerated generative models for 3D point cloud data,” in *Proc. of the IEEE Conf. on Comput. Vision and Pattern Recognition*, Las Vegas, NV, 2016.
- [12] T. Cieslewski, S. Choudhary, and D. Scaramuzza, “Data-efficient decentralized visual SLAM,” in *Proc. of the IEEE Intl. Conf. on Robot. and Autom.*, Brisbane, Australia, May 2018.
- [13] Y. Tian, K. Khosoussi, M. Giamou, J. P. How, and J. Kelly, “Near-optimal budgeted data exchange for distributed loop closure detection,” in *Proc. of Robot.: Sci. and Syst.*, Pittsburgh, PA, 2018.
- [14] A. J. Smith and G. A. Hollinger, “Distributed inference-based multi-robot exploration,” *Auton. Robots*, 2018.
- [15] J. Delmerico, S. Isler, R. Sabzevari, and D. Scaramuzza, “A comparison of volumetric information gain metrics for active 3D object reconstruction,” *Auton. Robots*, vol. 42, no. 2, pp. 197–208, 2018.
- [16] C. Witting, M. Fehr, R. Bähnemann, H. Oleynikova, and R. Siegwart, “History-aware autonomous exploration in confined environments using MAVs,” *arXiv preprint arXiv:1803.10558*, 2018.
- [17] T. Cieslewski, E. Kaufmann, and D. Scaramuzza, “Rapid exploration with multi-rotors: A frontier selection method for high speed flight,” in *Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst.*, Vancouver, Canada, Sep. 2017.
- [18] B. Charrow, G. Kahn, S. Patil, S. Liu, K. Goldberg, P. Abbeel, N. Michael, and V. Kumar, “Information-theoretic planning with trajectory optimization for dense 3D mapping,” in *Proc. of Robot.: Sci. and Syst.*, Rome, Italy, Jul. 2015.
- [19] M. Lauri and R. Ritala, “Planning for robotic exploration based on forward simulation,” *Robot. Auton. Syst.*, vol. 83, 2016.
- [20] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, “Receding horizon path planning for 3D exploration and surface inspection,” *Auton. Robots*, vol. 42, no. 2, pp. 291–306, 2018.
- [21] A. Valero-Gómez, J. V. Gómez, S. Garrido, and L. Moreno, “Fast marching methods in path planning,” *IEEE Robot. Autom. Mag.*, 2013.
- [22] S. Garrido, L. Moreno, and D. Blanco, “Exploration of 2D and 3D environments using Voronoi transform and fast marching method,” *J. Intell. & Robot. Syst.*, vol. 55, no. 1, pp. 55–80, 2009.
- [23] J. Butzke and M. Likhachev, “Planning for multi-robot exploration with multiple objective utility functions,” in *Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst.*, San Francisco, CA, Sep. 2011.
- [24] L. Klodt and V. Willert, “Equitable workload partitioning for multi-robot exploration through pairwise optimization,” in *Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst.*, Hamburg, Germany, Sep. 2015.
- [25] C. M. Bishop, *Pattern Recognition and Machine Learning*. Berlin, Heidelberg: Springer-Verlag, 2006.
- [26] G. J. McLachlan and S. Rathnayake, “On the number of components in a gaussian mixture model,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 4, no. 5, pp. 341–355, 2014.
- [27] J. C. Príncipe, *Information Theoretic Learning: Rényi’s Entropy and Kernel Perspectives*. New York, NY: Springer Science & Business Media, 2010.
- [28] G. Chaslot, “Monte-Carlo tree search,” Ph.D. dissertation, Universiteit Maastricht, 2010.
- [29] G. Best, O. M. Cliff, T. Patten, R. R. Mettu, and R. Fitch, “Decentralised Monte Carlo tree search for active perception,” in *Algorithmic Found. Robot.*, San Francisco, CA, Dec. 2016.
- [30] W. Tabib, M. Corah, N. Michael, and R. Whittaker, “Computationally efficient information-theoretic exploration of pits and caves,” in *Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst.*, Daejeon, Korea, Oct. 2016.
- [31] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York, NY: John Wiley & Sons, 2012.