

Riemannian Optimization for Active Mapping with Robot Teams

Arash Asgharivaskasi¹, Member, IEEE, Fritz Girke², Nikolay Atanasov¹, Senior Member, IEEE

Abstract—Autonomous exploration of unknown environments using a team of mobile robots demands distributed perception and planning strategies to enable efficient and scalable performance. Ideally, each robot should update its map and plan its motion not only relying on its own observations, but also considering the observations of its peers. Centralized solutions to multi-robot coordination are susceptible to central node failure and require a sophisticated communication infrastructure for reliable operation. Current decentralized active mapping methods consider simplistic robot models with linear-Gaussian observations and Euclidean robot states. In this work, we present a distributed multi-robot mapping and planning method, called Riemannian Optimization for Active Mapping (ROAM). We formulate an optimization problem over a graph with node variables belonging to a Riemannian manifold and a consensus constraint requiring feasible solutions to agree on the node variables. We develop a distributed Riemannian optimization algorithm that relies only on one-hop communication to solve the problem with consensus and optimality guarantees. We show that multi-robot active mapping can be achieved via two applications of our distributed Riemannian optimization over different manifolds: distributed estimation of a 3-D semantic map and distributed planning of $SE(3)$ trajectories that minimize map uncertainty. We demonstrate the performance of ROAM in simulation and real-world experiments using a team of robots with RGB-D cameras. Open-source software and videos supplementing this paper are available at <https://existentialrobotics.org/ROAM/>.

Index Terms—Distributed Robot Systems, Reactive and Sensor-Based Planning, Mapping, Distributed Riemannian Optimization

I. INTRODUCTION

The ability to explore unknown environments and discover objects of interest is a prerequisite for autonomous execution of complex tasks by mobile robots. Active mapping methods [1] consider joint optimization of the motion of a robot team and the fidelity of the map constructed by the team. The goal is to compute maximally informative robot trajectories under a limited exploration budget (e.g., time, energy, etc.).

Time-critical applications, such as search and rescue [2], [3] and security and surveillance [4], as well as large-scale operations, such as environmental monitoring [5], substantially benefit if exploration is carried out by a team of coordinating robots. This is traditionally done via multi-robot systems relying on centralized estimation and control [6], [7], where

The authors gratefully acknowledge support from NSF FRR CAREER 2045945, ONR N00014-23-1-2353, and ARL DCIST W911NF-17-2-0181.

¹A. Asgharivaskasi and N. Atanasov are with the Department of Electrical and Computer Engineering, University of California San Diego, La Jolla, CA 92093, USA (e-mails: {aasghari, natanasov}@ucsd.edu).

²F. Girke is with the School of Computation, Information and Technology, Technical University of Munich, Munich 80333, Germany (e-mail: fritz.girke@tum.de).

The authors thank Tianji Tang and Jason Stanley from the Department of Electrical and Computer Engineering at the University of California San Diego for their assistance with the simulation experiments in Gazebo.

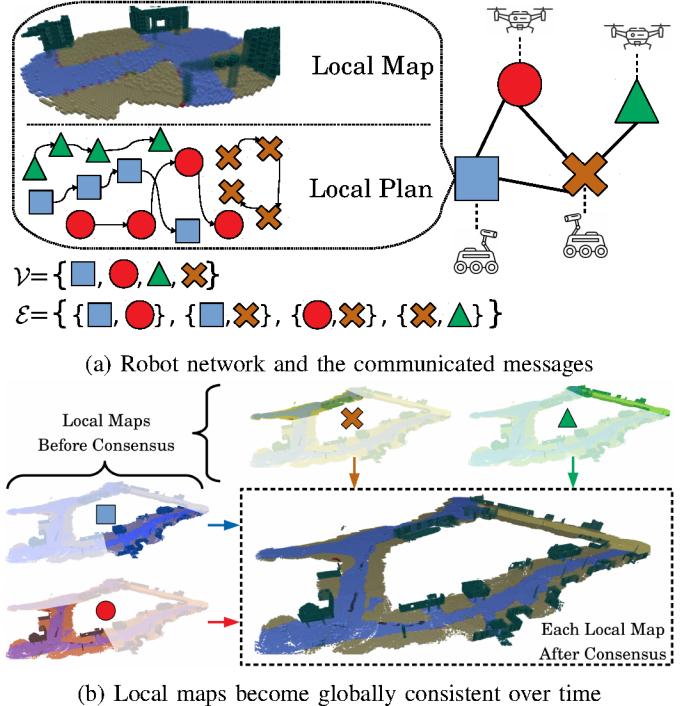


Fig. 1: Overview of our distributed multi-robot active mapping approach. (a) A team of robots, denoted by vertex set \mathcal{V} , collaboratively explores an unknown environment, such as an impacted area during disaster response or a hostile area during military reconnaissance. Each robot builds a local map using onboard sensor measurements and computes a local plan for the team, with the goal of maximizing the collective information gathered by the team. The local maps and plans are communicated over a peer-to-peer network whose connectivity is represented by the edge set \mathcal{E} . (b) As the robot team continues communication, the local maps stored by different robots become globally consistent in that they store similar information about the environment.

each robot receives local sensor observations, builds its own map, and sends it to a central node for map aggregation and team trajectory computation. The availability of powerful computation onboard small robot platforms makes it possible to develop autonomous exploration algorithms without the need for a central processing node [8]. Removing the central unit improves the resilience of a multi-robot system with respect to communication-based faults and central node failures [9] but brings up new challenges related to distributed storage, computation, and communication. How can one guarantee that the performance of decentralized active mapping would be on par with a centralized architecture in terms of global map consistency and team trajectory optimality?

To address this question, we propose ROAM: Riemannian

Optimization for Active Mapping with robot teams. ROAM is a decentralized Riemannian optimization algorithm that operates on a communication graph with node variables belonging to a Riemannian manifold and ensures consensus among the node variables. The graph nodes correspond to different robots, while the graph edges model the communication among the robots. In the context of mapping, the node variables are categorical probability mass functions representing probabilistic maps with different semantic classes (e.g., building, vegetation, terrain) at each robot. The consensus constraint requires that the local maps of different robots agree with each other. In the context of planning, the node variables are trajectories of $SE(3)$ robot poses. Each robot plans trajectories for the whole team using its local information, while the consensus constraint requires that the team trajectories computed by different robots agree. See Fig. 1 for an overview of ROAM.

We demonstrate the performance of ROAM in a variety of simulation and real-world experiments using a team of wheeled robots with on-board sensing and processing hardware. Specifically, each robot gathers range and semantic segmentation measurements using an RGBD sensor, and incrementally builds a local 3-D semantic grid map of the environment, where each map cell maintains a probability distribution over object classes. To achieve memory and communication efficiency, an octree data structure is employed to represent the 3-D semantic maps [10]. The robots cooperatively find the most informative set of $SE(3)$ paths for the team to efficiently improve the map and explore the unknown areas while avoiding obstacle collisions. Both multi-robot mapping and planning are performed in the absence of a central estimation and control node and only involve peer-to-peer communication among neighboring robots.

A. Related Work

1) *Distributed Optimization*: Multi-robot active mapping is in essence an optimization problem, with the goal of finding maximally informative robot trajectories, while simultaneously maintaining globally consistent map estimates. Thus, we begin our literature review with identifying relevant works in distributed optimization. The algorithms introduced in [11]–[13] provide a class of approaches for decentralized gradient-based optimization in the Euclidean space under a variety of constraints such as time variation or communication asymmetry between agents in the network. The survey by Halsted et al. [14] provides a comprehensive study of distributed optimization methods for multi-robot applications. In this work, we decompose the task of multi-robot active mapping to two consensus-constrained Riemannian optimization problems, i.e. distributed mapping and distributed path planning. However, naive utilization of the Euclidean optimization techniques in Riemannian manifolds might violate the structure of the optimization domain, leading to infeasible solutions. Therefore, it is required to employ a special family of distributed optimization methods specific to Riemannian manifolds.

2) *Optimization over Riemannian Manifolds*: Absil et al. [15] presents the foundations of optimization over matrix manifolds, giving rise to many centralized and distributed

algorithms in subsequent works. As examples, Chen et al. [16] and Wang et al. [17] devise decentralized optimization algorithms for Stiefel manifolds where a Lagrangian function is used to enforce consensus and maintain the manifold structure. Manifold optimization also allows designing efficient learning algorithms where model parameters can be learned using unconstrained manifold optimization as opposed to Euclidean space optimization with projection to the parameter manifold. Zhang et al. [18] and Li et al. [19] introduce stochastic learning algorithms for Riemannian manifolds in centralized and federated formats, respectively. Related to our work, Tian et al. [20] present a multi-robot pose-graph simultaneous localization and mapping (SLAM) algorithm which employs gradient-descent local to each robot directly over the $SE(3)$ space of poses. Our work is inspired by the distributed Riemannian gradient optimization method introduced by Shah [21]. We develop a distributed gradient-descent optimization method for general Riemannian manifolds, and derive specific instantiations for two particular cases, namely the space of probability distributions over semantic maps and the space of $SE(3)$ robot pose trajectories.

3) *Multi-Robot Mapping*: Distributed mapping is a special case of distributed estimation, where a model of the environment is estimated via sensor measurements. Distributed estimation techniques are used in multi-robot localization [22], multi-robot mapping [23], or multi-robot SLAM [24]. Paritosh et al. [25] define Bayesian distributed estimation as maximizing sensor data likelihood from all agents, while enforcing consensus in the estimates. The present work follows a similar methodology in that we achieve multi-robot Bayesian semantic mapping via distributed maximization of local sensor observation log-likelihood with a consensus constraint on the estimated maps. Regarding collaborative mapping, an important consideration is the communication of local map estimates among the robots. Corah et al. [26] propose distributed Gaussian mixture model (GMM) mapping, where a GMM map is globally estimated, and each robot uses this global map to extract occupancy maps for planning. The use of GMM environment representation for multi-robot exploration is motivated by its lower communication overhead compared to uniform resolution occupancy grid maps. Subsequent works in [27] and [28] have similarly used distributed GMM mapping for place recognition and relative localization alongside exploration. Alternative techniques for communication-efficient multi-robot mapping include sub-map-based grid mapping [29] and distributed truncated signed distance field (TSDF) estimation [30]. More recently, the work in [31] extends neural implicit signed distance mapping to a distributed setting via formulating multi-robot map learning as a consensus-constrained minimization of the loss function. In this case, the robots need to share the neural network parameters to achieve consensus. In our work, we use a semantic octree data structure introduced in our prior work [10] to alleviate the communication burden by using a lossless octree compression. Relevant to our work, the authors in [32] propose merging of two binary octree maps via summing the occupancy log-odds of corresponding octree leaves. Our work distinguishes itself from [32] through a different formulation of multi-

robot mapping as a consensus-based Riemannian optimization problem, which enables *a*) extension to multi-class octree representations, and *b*) combination of map merging with online map updates from local observations.

4) *Planning for Exploration*: Similar to multi-robot mapping, many multi-robot planning methods utilize distributed optimization techniques. The work in [33] outlines various trajectory planning methods used in multi-robot systems, including graph-based, sampling-based, model-based, and bio-inspired approaches. In particular, graph neural networks (GNNs) have been utilized in [34], [35] for learning to extract, communicate, and accumulate features from local observations in the context of collaborative multi-robot planning in a distributed way. Coordination and plan deconfliction for multi-robot cooperative tasks is discussed in [36], where robots are assigned priorities in a decentralized manner in order to reach a Pareto equilibrium. The paper by Freda et al. [37] uses a two-level deconfliction approach for multi-robot exploration to minimize redundant visitations (topological-level) and avoid inter-robot collisions (metric-level). In our work, we introduce a decentralized gradient-based negotiation mechanism to resolve $SE(3)$ path conflicts. Path planning for autonomous exploration has been extensively studied in the field of active SLAM. Atanasov et al. [38] propose a distributed active SLAM method for robots with linear-Gaussian observation models and a finite set of admissible controls. The authors exploit the conditional entropy formula for the Gaussian noise model to derive an open-loop control policy, called reduced value iteration (RVI), with the same performance guarantees as a closed-loop policy. An anytime version of RVI is proposed in [39] using a tree search that progressively reduces the suboptimality of the plan. In contrast to [38], [39], we use a probabilistic range-category observation model that accounts for occlusion in sensing. Sampling-based solutions to multi-robot active SLAM have been presented in [40] and [41], with asymptotic optimality guarantees. Cai et al. [42] consider collision safety and energy as additional factors in the cost function for active SLAM using a heterogeneous team of robots. Zhou and Kumar [43] propose robust multi-robot active target tracking with performance guarantees in regard to sensing and communication attacks, however, the estimation and control are carried out centrally. Tzes et al. [44] develop a learning-based approach for multi-robot target estimation and tracking, used a GNN to accumulate and process information communicated among one-hop neighbors. The works in [45], [46] aim to maintain multi-robot network connectivity and collision avoidance via control barrier functions. Another line of research [47], [48] uses decentralized Monte-Carlo tree search for multi-robot path planning for exploration. The interested reader is encouraged to refer to [1] for a comprehensive survey of active SLAM methods. Our work distinguishes itself by considering continuous-space planning on a Riemannian manifold, generalizing the previous works in terms of the finite number of controls and the Euclidean robot states. Related to active SLAM with continuous-space planning, Koga et al. [49], [50] introduce iterative covariance regulation, an $SE(3)$ trajectory optimization algorithm for single-robot active SLAM with a Gaussian observation model.

Model-based [51] and model-free [52] deep reinforcement learning techniques have been applied to similar single-robot active SLAM problems. Extending to a team of robots, Hu et al. [53] propose Voronoi-based decentralized exploration using reinforcement learning, where coordination among the robots takes place via distributed assignment of each Voronoi region to a robot, and the policy generates a 2-D vector of linear and angular velocities. In our work, we formulate multi-robot planning for exploration as a distributed optimization problem in $SE(3)$ space with a consensus constraint to enforce agreement among the robot plans.

B. Contributions

Compared to the works mentioned in the previous part, our distributed Riemannian optimization approach extends the scope of multi-robot estimation and planning to enable continuous non-Euclidean state and control spaces and non-linear non-Gaussian perception models. Our contributions include:

- 1) a distributed Riemannian optimization algorithm for multi-robot systems using only one-hop communication, with consensus and optimality guarantees,
- 2) a distributed semantic octree mapping approach utilizing local semantic point cloud observations at each robot,
- 3) a distributed collaborative planning algorithm for robot exploration, where the search domain is defined as the continuous space of $SE(3)$ robot pose trajectories,
- 4) an open-source implementation, achieving real-time performance onboard resource-constrained robots in simulation and real-world experiments.

We begin by formulating consensus-constrained Riemannian optimization for multi-agent systems in Sec. II. Next, in Sec. III, we introduce a distributed Riemannian optimization algorithm with consensus and optimality guarantees. In Sec. IV, we formulate distributed semantic octree mapping as a special case, where the optimization variables are probability mass functions over the set of possible semantic maps. Sec. V formulates distributed collaborative planning for robot exploration as another application of distributed Riemannian optimization, where robot trajectories in the $SE(3)$ manifold are the optimization variables. Lastly, in Sec. VI we evaluate the performance of our proposed distributed multi-robot exploration in several simulation and real-world experiments.

II. PROBLEM STATEMENT

Consider a network of agents represented by an undirected connected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, where \mathcal{V} denotes the set of agents and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ encodes the existence of communication links between pairs of agents. Each agent $i \in \mathcal{V}$ has state x^i which belongs to a compact Riemannian manifold \mathcal{M} . Let $T_{x^i}\mathcal{M}$ denote the tangent space of \mathcal{M} at x^i and let $\langle v, u \rangle_{x^i} \in \mathbb{R}$ with $u, v \in T_{x^i}\mathcal{M}$ be a Riemannian metric on \mathcal{M} [54, Ch.3]. The norm of a tangent vector $v \in T_{x^i}\mathcal{M}$ is defined by the Riemannian metric as $\|v\|_{x^i} = \sqrt{\langle v, v \rangle_{x^i}}$. Additionally, let $\text{Exp}_{x^i}(\cdot) : T_{x^i}\mathcal{M} \rightarrow \mathcal{M}$ denote the exponential map on \mathcal{M} at x^i , and denote its inverse as $\text{Exp}_{x^i}^{-1}(\cdot) : \mathcal{M} \rightarrow T_{x^i}\mathcal{M}$.

We associate a local objective function $f^i(\cdot) : \mathcal{M} \rightarrow \mathbb{R}$ with each agent $i \in \mathcal{V}$. Our goal is to maximize the cumulative objective function over the joint agent state $\mathbf{x} = (x^1, \dots, x^{|\mathcal{V}|})$:

$$F(\mathbf{x}) = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} f^i(x^i). \quad (1)$$

The global objective can be maximized using $|\mathcal{V}|$ independent local optimizations. However, in many applications it is necessary to find a common solution among all agents. For example, in multi-robot mapping, the robots need to ensure that their local maps are consistent and take into account the observations from other robots. Therefore, the global optimization problem needs to be constrained such that the agents reach consensus on \mathbf{x} during optimization. For this aim, we define an aggregate distance function $\phi(\mathbf{x}) : \mathcal{M}^{|\mathcal{V}|} \rightarrow \mathbb{R}_{\geq 0}$:

$$\phi(\mathbf{x}) = \sum_{\{i,j\} \in \mathcal{E}} A_{ij} d^2(x^i, x^j), \quad (2)$$

where A is a symmetric weighted adjacency matrix corresponding to the graph \mathcal{G} , and $d(\cdot) : \mathcal{M} \times \mathcal{M} \rightarrow \mathbb{R}_{\geq 0}$ is a distance function on the Riemannian manifold \mathcal{M} , i.e., computes the length of the geodesic (shortest path) between pairs of elements in \mathcal{M} . The definition of the aggregate distance function in (2) implies that consensus will be reached if and only if $\phi(\mathbf{x}) = 0$. Hence, adding $\phi(\mathbf{x}) = 0$ as a constraint to (1) would require feasible joint states $\mathbf{x} = (x^1, \dots, x^{|\mathcal{V}|})$ to satisfy $x^i = x^j$ for all $i, j \in \mathcal{V}$.

Problem 1. Consider a connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where each node $i \in \mathcal{V}$ represents an agent with state $x^i \in \mathcal{M}$ and local objective function $f^i(x^i)$. Find a joint state \mathbf{x} that maximizes the following objective function:

$$\begin{aligned} \max_{\mathbf{x}} \quad & F(\mathbf{x}) = \frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} f^i(x^i), \\ \text{s.t.} \quad & x^i \in \mathcal{M}, \quad \forall i \in \mathcal{V}, \quad \text{and} \quad \phi(\mathbf{x}) = 0, \end{aligned} \quad (3)$$

where $\phi(\mathbf{x}) = 0$ is the consensus constraint defined in (2).

As we discuss in Sec. IV and Sec. V, both multi-robot mapping and multi-robot trajectory optimization can be formulated as consensus-constrained optimization problems as in (3). In mapping, the manifold \mathcal{M} is the probability simplex capturing map density functions while the local objective $f^i(x^i)$ is the log-likelihood of the observations made by robot i . In trajectory optimization, \mathcal{M} represents the space of 3-D pose (rotation and translation) trajectories in $SE(3)$, and $f^i(x^i)$ is a collision and perception-aware objective for the robot pose trajectories. In the next section, we develop a distributed gradient-based optimization algorithm to solve (3) using only local computation and single-hop communication.

III. DISTRIBUTED RIEMANNIAN OPTIMIZATION

The problem in (3) has a specific structure, maximizing a sum of local objectives subject to a consensus constraint among all x^i , $i \in \mathcal{V}$. We develop a distributed gradient-based algorithm to solve (3). The idea is to interleave gradient updates for the local objectives with gradient updates for the consensus constraint at each agent. Alg. 1 formalizes

Algorithm 1 Distributed Riemannian Optimization

Input: Network $\mathcal{G}(\mathcal{V}, \mathcal{E})$ and initial state $x^{i(0)}$

Output: Consensus optimal solution to (3)

```

1: for  $k \in \mathbb{Z}_{\geq 0}$  do
2:   for each agent  $i \in \mathcal{V}$  do
3:      $\triangleright$  Promote consensus with step size  $\epsilon$ :
4:      $\tilde{x}^{i(k)} = \text{Exp}_{x^{i(k)}}(-\epsilon \text{grad}_{x^i} \phi(\mathbf{x})|_{\mathbf{x}=\mathbf{x}^{(k)}})$ 
5:      $\triangleright$  Optimize local objective with step size  $\alpha^{(k)}$ :
6:      $x^{i(k+1)} = \text{Exp}_{\tilde{x}^{i(k)}}(\alpha^{(k)} \text{grad } f^i(x^i)|_{x^i=\tilde{x}^{i(k)}})$ 
7: return  $x^{i(k)}$ 

```

this idea. The update step in line 4 guides the local state x^i towards satisfaction of the consensus constraint, with a step size of ϵ . The gradient of $\phi(\mathbf{x})$ with respect to x^i , denoted as $\text{grad}_{x^i} \phi(\mathbf{x})$, lies in the tangent space $T_{x^i} \mathcal{M}$. Hence, the exponential map is used to retract the gradient update $-\epsilon \text{grad}_{x^i} \phi(\mathbf{x})|_{\mathbf{x}=\mathbf{x}^{(k)}}$ to the manifold \mathcal{M} . The gradient $\text{grad}_{x^i} \phi(\mathbf{x})$ can be expressed as a sum of gradients with respect to the neighbors $\mathcal{N}_i = \{j | A_{ij} > 0\}$ of agent i :

$$\text{grad}_{x^i} \phi(\mathbf{x}) = \sum_{j \in \mathcal{N}_i} A_{ij} \text{grad}_{x^i} d^2(x^i, x^j) = -2 \sum_{j \in \mathcal{N}_i} A_{ij} \text{Exp}_{x^i}^{-1}(x^j).$$

Therefore, line 4 requires *only single-hop communication* between agent i and its neighbors \mathcal{N}_i . Line 6 carries out an update with step size $\alpha^{(k)}$ in the direction of the gradient of the local objective $f^i(\cdot)$, computed at the updated state $\tilde{x}^{i(k)}$. Similar to the consensus update step, the exponential map is used in to retract $\text{grad } f^i(x^i)$ and apply it to the point $\tilde{x}^{i(k)}$. Line 6 is local to each agent i and does not require communication. The two update steps are continuously applied until a maximum number of iterations is reached or the update norm is smaller than a threshold.

Example. Consider a sensor network where several agents gather data that is not supposed to be shared over the network, due to either privacy reasons or bandwidth limitations. Our Riemannian optimization algorithm enables distributed processing of the global data, accumulated over all agents, without actually sharing the data. As an example, Fig. 2 illustrates applying Alg. 1 to compute the leading eigenvector of the covariance of data. Fig. 2a depicts the global data distribution $Z = [Z_1^\top Z_2^\top]^\top$, such that different segments of the data Z_1 and Z_2 are known to agent 1 and agent 2, respectively. This problem can be formulated as:

$$\begin{aligned} \max_{x^1, x^2} \quad & \sum_{i \in \{1, 2\}} (Z_i x^i)^\top Z_i x^i, \\ \text{s.t.} \quad & x^1, x^2 \in \mathbb{S}^1 \quad \text{and} \quad \text{arccos}(x^1 \top x^2) = 0, \end{aligned} \quad (4)$$

where the domain manifold is the unit circle \mathbb{S}^1 , and cosine distance is used as the distance function. Note that for all x^1 and x^2 that satisfy the consensus constraint $\text{arccos}(x^1 \top x^2) = 0$, the objective function is equivalent to the one for the centralized leading eigenvector problem. Hence, we expect to find the eigenvector for the covariance of the global data matrix Z by employing Alg. 1 to (4). Fig. 2b shows an initialization of x^1 and x^2 over the unit circle \mathbb{S}^1 . While the Riemannian

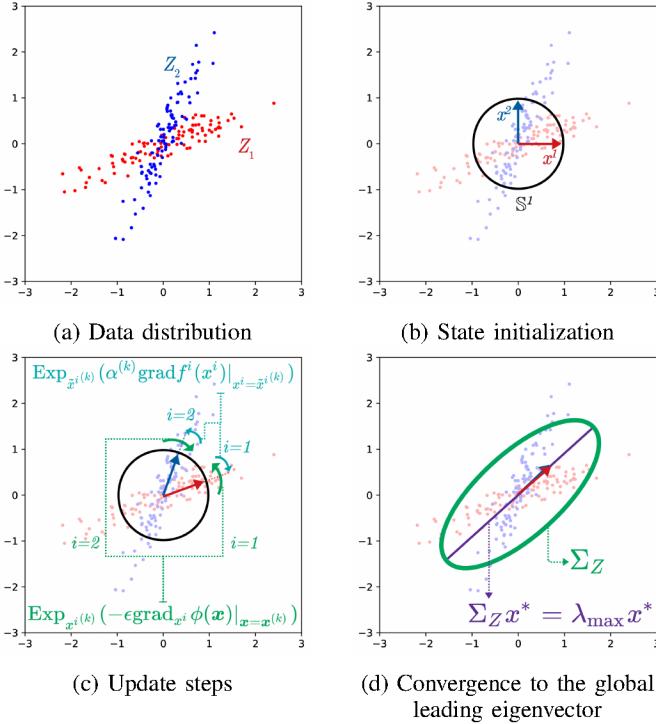


Fig. 2: Application of Alg. 1 to the leading eigenvector problem. (a) Data distribution $Z = [Z_1^\top Z_2^\top]^\top$, where Z_1 and Z_2 are separately known to agent 1 and agent 2. (b) State initialization, limited to the unit circle \mathbb{S}^1 since we are only interested in eigenvector directions. (c) Consensus and local objective function updates, shown in green and teal arcs, respectively. The exponential mapping of \mathbb{S}^1 maintains the manifold structure of the states throughout the update steps. (d) Level-set of the covariance matrix of the global data Z is shown, alongside its leading eigenvector x^* .

gradients of $\phi(\cdot)$ and $f^i(\cdot)$ are tangent vectors to \mathbb{S}^1 , the consensus and local objective function update steps keep the state on the \mathbb{S}^1 manifold thanks to the exponential map (see the circular arcs in Fig. 2c):

$$\text{Exp}_{x^i}(v) = \cos(\sqrt{v^\top v})x^i + \sin(\sqrt{v^\top v})\frac{v}{\sqrt{v^\top v}}. \quad (5)$$

Note that the consensus update (green arc) acts in the direction of agreement between x^1 and x^2 , whereas the local objective function gradient tries to steer the states x^i towards the leading eigenvector of their respective data Z_i . Although each agent has only partial access to Z , both x^1 and x^2 eventually converge to x^* , namely the leading eigenvector of the covariance for the global data matrix Z , as Fig. 2d suggests. •

Next, we study whether Alg. 1 achieves consensus and optimality. We make several assumptions to ensure that the problem is well-posed in accordance with prior work on distributed optimization [11]–[13], [55].

Definition 1. A differentiable function $f : \mathcal{M} \rightarrow \mathbb{R}$ is geodesically convex if and only if for any $x, y \in \mathcal{M}$:

$$f(x) \geq f(y) + \langle \text{grad } f(y), \text{Exp}_y^{-1}(x) \rangle_y.$$

The function f is geodesically concave if the above inequality is flipped.

Assumption 1. Assume the following statements hold for Problem 1 and Alg. 1.

- The Riemannian manifold \mathcal{M} is compact.
- The local objective functions f^i , $\forall i \in \mathcal{V}$, are smooth, geodesically concave, and their Riemannian gradients are bounded by some constant C :

$$\|\text{grad } f^i(x^i)\|_{x^i} \leq C, \forall x^i \in \mathcal{M}, \forall i \in \mathcal{V}.$$

- The weighted adjacency matrix A of the graph \mathcal{G} is row-stochastic, i.e., $\sum_{j \in \mathcal{V}} A_{ij} = 1$.
- The squared distance function $d^2 : \mathcal{M} \times \mathcal{M} \rightarrow \mathbb{R}_{\geq 0}$ is geodesically convex.
- The step sizes $\alpha^{(k)} > 0$ for the update step in line 6 satisfy the Robbins-Monro conditions:

$$\sum_{k=0}^{\infty} \alpha^{(k)} = \infty, \quad \sum_{k=0}^{\infty} \alpha^{(k)}^2 < \infty, \quad \forall k \geq 0.$$

In addition to the assumptions above, we require an additional condition to prove that Alg. 1 achieves consensus, i.e., $\phi(\mathbf{x}) = 0$. Let $\mathcal{T}_{y^i}^{x^i} : T_{y^i} \mathcal{M} \rightarrow T_{x^i} \mathcal{M}$ denote parallel transport [54, Ch.10] from the tangent space at y^i to the tangent space at x^i . For points $x^i, x^j, y^j, y^i \in \mathcal{M}$, consider the geodesic loop $x^i \rightarrow x^j \rightarrow y^j \rightarrow y^i \rightarrow x^i$ with corresponding tangent vectors $v_x^{ij}, v_y^{ij}, v_{xy}^{ij}, v_{yx}^{ij}$ defined as:

$$v_x^{ij} = \text{Exp}_{x^i}^{-1}(x^j), \quad v_{xy}^{ij} = \text{Exp}_{x^i}^{-1}(y^j),$$

with similar definitions for v_y^{ij} and v_{yx}^{ij} . Let $v_{xy}^{ij} \in T_{x^i} \mathcal{M}$ be the net tangent vector transported to $T_{x^i} \mathcal{M}$:

$$v_{xy}^{ij} = v_x^{ij} + \mathcal{T}_{x^j}^{x^i} v_{xy}^{ij} - v_{xy}^{ij} - \mathcal{T}_{y^i}^{x^i} v_y^{ij}. \quad (6)$$

Assumption 2. For a $\rho > 0$ and any 4-tuple $(x^i, x^j, y^j, y^i) \in \mathcal{M}$, assume the norm of the net tangent vector v_{xy}^{ij} is bounded by the lengths of the opposite geodesics along the loop:

$$\|v_{xy}^{ij}\|_{x^i} \leq \rho \min\{\|v_x^i\|_{x^i} + \|v_y^j\|_{x^j}, \|v_x^j\|_{x^i} + \|v_y^i\|_{x^i}\}. \quad (7)$$

In Euclidean space, the net tangent vector v_{xy}^{ij} is equivalent to zero linear displacement; hence, the assumption holds for any $\rho \geq 0$. Similarly, the manifold \mathbb{S}^1 of example (4) satisfies (7) due to zero angular displacement. For a general case, v_{xy}^{ij} can be non-zero due to the curvature of the manifold. This is dual to the fact that, for a zero net tangent vector v_{xy}^{ij} , the corresponding geodesics might not form a loop. The assumption in (7) essentially imposes a condition over curvature of the manifold so that the norm of v_{xy}^{ij} is limited by the length of the geodesic loop. Based on the above assumptions, we show consensus and optimality for Alg. 1.

Theorem 1. Consider the consensus-constrained Riemannian optimization problem in (3) and the distributed Riemannian optimization algorithm in Alg. 1. Suppose that Assumptions 1 and 2 hold and step size ϵ is chosen such that $\epsilon \in (0, 2/L)$ with $L = 4(1 + \rho)$. Then, Alg. 1 provides a solution to (3) with the following properties.

- I) The joint state $\mathbf{x}^{(k)}$ converges to $\mathbf{x}^{(\infty)} \in \mathcal{M}^{|\mathcal{V}|}$, where $\mathbf{x}^{(\infty)}$ is a consensus configuration, i.e., $x^{i(\infty)} = x^{j(\infty)}$ for all $i, j \in \mathcal{V}$.

2) Let \mathbf{x}^* be an optimal solution to (3). The optimal value $F(\mathbf{x}^*)$ is a lower-bound for the maximum of $F(\mathbf{x}^{(k)})$ across all iterations:

$$F(\mathbf{x}^*) \leq \lim_{k_{\max} \rightarrow \infty} \max_{0 \leq k \leq k_{\max}} F(\mathbf{x}^{(k)}). \quad (8)$$

Proof. See Appendix A. \square

We stress that, while the optimal solution \mathbf{x}^* and the convergence point $\mathbf{x}^{(\infty)}$ of Alg. 1 are both consensus configurations, the optimality bound of (8) can potentially admit a solution $\mathbf{x}^{(k)}$ that does not satisfy the consensus constraint. For the Euclidean case, Nedić [56, Ch.5] shows that $d^2(\mathbf{x}^*, \mathbf{x}^{(k)})$ is a Lyapunov function, and subsequently, $F(\mathbf{x}^*) = F(\mathbf{x}^{(\infty)})$ holds. However, a similar derivation for $d^2(\mathbf{x}^*, \mathbf{x}^{(k)})$ has not been found for a general Riemannian manifold, due to the complexity added by the curvature.

Alg. 1 establishes consensus and an optimality bound without requiring identical initial states $x^{i(0)} = x_0$ for all $i \in \mathcal{V}$ or parallel transport of the gradients between neighboring agents. Hence, our distributed Riemannian optimization provides an approach to solve multi-robot problems with communication constraints. The main requirement to use Alg. 1 is to express a multi-robot optimization problem in the form of (3), with local objectives $f^i(\cdot)$ and distance measure $\phi(\cdot)$ defined as smooth concave and convex functions in \mathcal{M} and $\mathcal{M}^{|\mathcal{V}|}$, respectively. In the absence of concavity for the objective functions or convexity for the consensus constraint, Alg. 1 can still be utilized to obtain a solution with local consensus and optimality guarantees as long as smoothness of $\phi(\mathbf{x})$ and boundedness of the objective function gradient hold.

In the next two sections, we apply Alg. 1 to achieve simultaneous multi-robot mapping and planning. We refer to our approach as *Riemannian Optimization for Active Mapping (ROAM)*. In Sec. IV, we apply Alg. 1 to multi-robot estimation of semantic octree maps, while in Sec. V we use Alg. 1 to achieve multi-robot motion planning for exploration and active estimation of semantic octree maps.

IV. MULTI-ROBOT SEMANTIC OCTREE MAPPING

In this section, we design a decentralized multi-robot octree mapping algorithm using the results from Sec. III. We consider a team of robots gathering local sensor measurements and communicating map estimates with one-hop neighbors in order to build a globally consistent common map. The robots are navigating in an environment consisting of disjoint sets $\mathcal{S}_c \subset \mathbb{R}^3$, each associated with a semantic category $c \in \mathcal{C} := \{0, 1, \dots, C\}$. Let \mathcal{S}_0 represent the free space and let each \mathcal{S}_c for $c > 0$ represent a different category, such as building, vegetation, terrain. Each robot $i \in \mathcal{V}$ is equipped with a mounted sensor that provides a stream of semantically-annotated point clouds in the sensor frame. Such information may be obtained by processing the measurements of an RGBD camera [57] or a LiDAR with a semantic segmentation algorithm [58]. We model a point cloud as a set $\mathbf{z}_t^i = \{(r_{t,b}^i, y_{t,b}^i)\}_{b=1}^B$ of B rays with known direction at time t , containing the distance $r_{t,b}^i \in \mathbb{R}_{\geq 0}$ from the sensor's position to the closest obstacle along the ray in addition to object category $y_{t,b}^i \in \mathcal{C}$ of the obstacle (see Fig. 3).

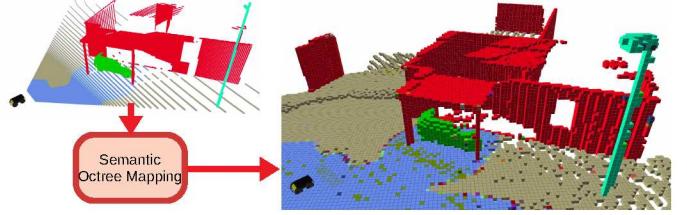


Fig. 3: Semantically annotated point cloud, with object categories in different colors, integrated in multi-class octree map.

We represent the map \mathbf{m} as a 3-D grid of N independent cells, where each individual cell m is labeled with a category in \mathcal{C} . To model measurement noise, we use a probability density function (PDF) $q^i(\mathbf{z}_t^i | \mathbf{m})$ as the observation model of each robot. The observation model $q^i(\mathbf{z}_t^i | \mathbf{m})$ depends on the sensor pose as well but we assume that accurate sensor poses are available from localization and calibration between the robot body frame and the sensor frame. We intend to perform probabilistic mapping, which requires maintaining a PDF of the map, and updating it based on sensor observations. For this aim, we maximize the sum of expected log-likelihood of the measurements up to time t , i.e. local observations $\mathbf{z}_{1:t}^i$ collected from each robot $i \in \mathcal{V}$ ¹:

$$\max_{p \in \mathcal{P}} \sum_{i \in \mathcal{V}} \mathbb{E}_{\mathbf{m} \sim p} [\log q^i(\mathbf{z}_{1:t}^i | \mathbf{m})], \quad (9)$$

where \mathcal{P} is the space of all probability mass functions (PMF) over the set of possible maps:

$$\mathcal{P} = \{p(\cdot) | \sum_{\mathbf{m}} p(\mathbf{m}) = 1, p(\mathbf{m}) \geq 0 \ \forall \mathbf{m} \in \mathcal{C}^N\} \quad (10)$$

The map cell independence assumption allows for decomposing the measurement log-likelihood as a sum over individual map cells m , as indicated in the following lemma.

Lemma 1. *The objective function in (9) can be expressed as a sum over all map cells and all observations:*

$$\sum_{i \in \mathcal{V}} \sum_{\tau=1}^t \left(\log q^i(\mathbf{z}_{\tau}^i) + \sum_{n=1}^N \mathbb{E}_{m \sim p_n} [\log \frac{q^i(m | \mathbf{z}_{\tau}^i)}{p_n(m)}] \right), \quad (11)$$

where $q^i(\mathbf{z}_{\tau}^i)$ is the marginal density of the observation, $p_n(\cdot)$ denotes the PMF of the n -th map cell, and $q^i(m | \mathbf{z}_{\tau}^i)$ is an inverse observation model that represents the sensor noise properties (see (10) in [10]).

Proof. See Appendix B. \square

The log-density term $\log q^i(\mathbf{z}_{\tau}^i)$ in (11) does not depend on any of the map probabilities $p_n(\cdot)$, $n \in \{1, \dots, N\}$; hence, it can be removed from the objective without affecting the solution. Moreover, each term in the innermost summation in (11) only depends on a single map cell probability $p_n(\cdot)$. Therefore, the maximization of the objective can be carried out separately for each cell m :

$$\max_{p_n \in \mathcal{P}_c} \sum_{i \in \mathcal{V}} \mathbb{E}_{m \sim p_n} [\log \frac{q_t^i(m)}{p_n(m)}], \quad (12)$$

¹It can be shown that maximizing the sum of expected log-likelihood of the data is equivalent to minimizing the KL-divergence between the true and the evaluated observation models. See [25] for more details.

where $\mathcal{P}_{\mathcal{C}}$ is the space of categorical distributions over \mathcal{C} and

$$q_t^i(m) = \prod_{\tau=1}^t q^i(m|\mathbf{z}_\tau^i)^{1/t}. \quad (13)$$

In order to remove the constraint $p_n \in \mathcal{P}_{\mathcal{C}}$, we utilize multi-class log-odds ratio of the categorical distribution [10]:

$$\mathbf{h}_n := \begin{bmatrix} \log \frac{p_n(m=0)}{p_n(m=0)} & \dots & \log \frac{p_n(m=C)}{p_n(m=0)} \end{bmatrix}^\top \in \mathbb{R}^{C+1}. \quad (14)$$

A PMF and its log-odds representation have a one-to-one correspondence through the softmax function $\sigma : \mathbb{R}^{C+1} \rightarrow \mathbb{R}^{C+1}$:

$$p_n(m=c) = \sigma_{c+1}(\mathbf{h}_n) := \frac{\mathbf{e}_c^\top \exp(\mathbf{h}_n)}{\mathbf{1}^\top \exp(\mathbf{h}_n)},$$

where \mathbf{e}_c is the standard basis vector with c -th element equal to 1 and 0 elsewhere, $\mathbf{1}$ is the vector with all elements equal to 1, and $\exp(\cdot)$ is applied element-wise to the vector \mathbf{h}_n . In order to enable distributed optimization of the objective (12) via the framework of Sec. III, we introduce a constraint that requires the robots to agree on a common map estimate using only one-hop communication.

Problem 2. Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be a network of robots, where each robot $i \in \mathcal{V}$ collects semantic point cloud observations \mathbf{z}_i^i . Construct local estimates of the map log-odds \mathbf{h}^i at each robot i that are consistent across \mathcal{V} via the following optimization:

$$\begin{aligned} & \max_{\mathbf{h}^{1:|\mathcal{V}|} \in \mathbb{R}^{(C+1) \times |\mathcal{V}|}} \sum_{i \in \mathcal{V}} f^i(\mathbf{h}^i), \\ \text{s.t. } & \phi(\mathbf{h}^{1:|\mathcal{V}|}) = \sum_{\{i,j\} \in \mathcal{E}} A_{ij} \|\mathbf{h}^j - \mathbf{h}^i\|_2^2 = 0, \end{aligned} \quad (15)$$

where $f^i(\mathbf{h}^i) = \sum_{c \in \mathcal{C}} \sigma_{c+1}(\mathbf{h}^i) \log \frac{q_t^i(c)}{\sigma_{c+1}(\mathbf{h}^i)}$ and $q_t^i(c)$ is defined in (13).

The multi-robot mapping problem in (15) has the same structure as the general distributed optimization in (3). Therefore, the distributed Riemannian optimization algorithm (Alg. 1) can be employed to perform multi-robot semantic mapping. Note that $\phi(\cdot)$ is globally convex because of the flatness of Euclidean space. Thus, Theorem 1 guarantees that Alg. 1 can achieve consensus in the map estimates of all robots. The application of Alg. 1 to solve (15) in a distributed manner is presented in Alg. 2. The update step in line 4 guides the local log-odds towards satisfaction of the consensus constraint, which only requires single-hop communication between neighboring robots $j \in \mathcal{N}_i$. Line 9 incorporates the local observations via γ^i and β^i , where \odot is element-wise multiplication. This step is local to each robot i and does not require communication. Note that lines 4 and 11 resemble the log-odds equivalent of Bayes rule for updating multi-class probabilities (see (8) in [10]).

The distributed semantic mapping algorithm we developed assumes a regular grid representation of the environment. To reduce the storage and communication requirements, we may utilize a semantic octree data structure which provides a lossless compression of the original 3-D multi-class map. In this case, the update rules in Alg. 2 should be applied to all

Algorithm 2 Distributed Semantic Mapping

Input: Local observations $\mathbf{z}_{1:t}^i$ and initial multi-class map estimate $\mathbf{h}^{i(0)}$
Output: Globally consistent semantic map

- 1: **for** $k \in \mathbb{Z}_{\geq 0}$ **do**
- 2: **for** each cell in \mathbf{m} **do**
- 3: ▷ Promote consensus with step size ϵ_m :
- 4: $\tilde{\mathbf{h}}^{i(k)} = \mathbf{h}^{i(k)} + \epsilon_m \sum_{j \in \mathcal{N}_i} A_{ij} (\mathbf{h}^{j(k)} - \mathbf{h}^{i(k)})$
- 5: ▷ Local gradient computation:
- 6: $\Delta^i = \tilde{\mathbf{h}}^{i(k)} - \log \mathbf{q}_t^i \quad \triangleright \log \mathbf{q}_t^i = [\log q_t^i(c)]_{c=0}^C$
- 7: $\gamma^i = (\exp(\tilde{\mathbf{h}}^{i(k)})^\top \Delta^i) \mathbf{1}$
- 8: $\beta^i = (\exp(\tilde{\mathbf{h}}^{i(k)})^\top \mathbf{1}) \Delta^i$
- 9: $\mathbf{g}^i = (\gamma^i - \beta^i) \odot \frac{\exp(\tilde{\mathbf{h}}^{i(k)})}{(\exp(\tilde{\mathbf{h}}^{i(k)})^\top \mathbf{1})^2}$
- 10: ▷ Apply gradient with step size $\alpha_m^{(k)}$:
- 11: $\mathbf{h}_1^{i(k+1)} = \tilde{\mathbf{h}}^{i(k)} + \alpha_m^{(k)} \mathbf{g}^i$
- 12: $h_1^{i(k+1)} = 0 \quad \triangleright h_1^{i(k+1)} = \log \frac{p(m=0)}{p(m=0)} = 0$
- 13: **return** $\mathbf{h}^{i(k)}$

leaf nodes in the semantic octree map of each robot i . Refer to Alg. 3 in [10] for the semantic octree equivalents of the update steps in lines 4 and 11.

In this section, we presented the mapping component of ROAM as distributed construction of semantic octree maps given local semantic point cloud observations at each robot. In the next section, we introduce the multi-robot planning component of ROAM, where robots cooperatively find trajectories along which their observations are maximally informative. Employing ROAM for simultaneous distributed mapping and planning closes the loop for autonomously exploring an unknown environment with a team of robots.

V. MULTI-ROBOT PLANNING

We discussed the case where observations are collected passively along the robot trajectories and used for distributed mapping. In this section, we consider planning the motion of the robots to collect observations that reduce map uncertainty and uncover an unknown environment. This active mapping process prevents redundant observations that may not improve the map accuracy or increase the overall covered area.

Let $\mathbf{X}_t^i \in SE(3)$ be the pose of robot $i \in \mathcal{V}$, at time t :

$$\mathbf{X}_t^i = \begin{bmatrix} \mathbf{R}_t^i & \mathbf{p}_t^i \\ \mathbf{0}^\top & 1 \end{bmatrix},$$

where $\mathbf{R}_t^i \in SO(3)$ and $\mathbf{p}_t^i \in \mathbb{R}^3$ are the robot's orientation and position, respectively. The Lie algebra $\mathfrak{se}(3)$ corresponding to the Lie group $SE(3)$ is defined as follows:

$$\mathfrak{se}(3) = \left\{ \hat{\xi} := \begin{bmatrix} \hat{\theta} & \rho \\ \mathbf{0}^\top & 0 \end{bmatrix} \in \mathbb{R}^{4 \times 4} \mid \xi = \begin{bmatrix} \rho \\ \theta \end{bmatrix} \in \mathbb{R}^6 \right\},$$

with $(\hat{\cdot})$ used to denote the mapping from a vector $\xi \in \mathbb{R}^6$ to a 4×4 twist matrix in $\mathfrak{se}(3)$. The matrix exponential $\exp(\cdot) : \mathfrak{se}(3) \rightarrow SE(3)$ relates a twist in $\mathfrak{se}(3)$ to a pose in $SE(3)$ via the Rodrigues' formula:

$$\exp(\hat{\xi}) = I + \hat{\xi} + \frac{(1 - \cos \|\theta\|)}{\|\theta\|^2} \hat{\xi}^2 + \frac{(\|\theta\| - \sin \|\theta\|)}{\|\theta\|^3} \hat{\xi}^3.$$

The exponential mapping at an arbitrary pose $\mathbf{X} \in SE(3)$ with perturbation $\xi \in \mathbb{R}^6$ (in the robot frame) can be expressed as:

$$\text{Exp}_{\mathbf{X}}(\xi) = \mathbf{X} \exp(\hat{\xi}).$$

The distance between two poses \mathbf{X}_t^i and $\mathbf{X}_{t'}^j$ is defined as:

$$d^2(\mathbf{X}_t^i, \mathbf{X}_{t'}^j) = \xi_{\mathbf{X}_t^i, \mathbf{X}_{t'}^j}^\top \Gamma \xi_{\mathbf{X}_t^i, \mathbf{X}_{t'}^j}, \quad \xi_{\mathbf{X}_t^i, \mathbf{X}_{t'}^j} = \log(\mathbf{X}_t^{i-1} \mathbf{X}_{t'}^j)^\vee,$$

where the functions $\log(\cdot) : SE(3) \rightarrow \mathfrak{se}(3)$ and $(\cdot)^\vee : \mathfrak{se}(3) \rightarrow \mathbb{R}^6$ denote the inverse mappings associated with $\exp(\cdot)$ and (\cdot) , respectively. Also, $\Gamma \in \mathbb{R}^{6 \times 6}$ is a diagonal matrix with positive diagonal entries that account for the difference in scale between the linear and angular elements of $\xi_{\mathbf{X}_t^i, \mathbf{X}_{t'}^j}$. For more details, please refer to [59, Ch.7].

To enable gradient-based pose trajectory optimization, we introduce differentiable cost functions to quantify the safety and the informativeness of a pose trajectory. We use a distance field $D(\mathbf{X}_t^i, p_t^i(\mathbf{m}))$ as a measure of path safety derived from the map $p_t^i(\mathbf{m})$ of robot i given observations up to time t . To obtain the distance field, we extract a maximum likelihood occupancy map from $p_t^i(\mathbf{m})$ and compute the distance transform. Furthermore, we use semantic Shannon mutual information (SSMI) [10, Eq.(4)] denoted by $I(\mathbf{m}; \mathbf{z} | \mathbf{X}_t^i, p_t^i(\mathbf{m}))$ to quantify the informativeness of a range-category observation \mathbf{z} made from pose \mathbf{X}_t^i with respect to the current multi-class map $p_t^i(\mathbf{m})$ of robot i . Having SSMI in the objective function allows explicit consideration of the class probabilities, stored in the map. As detailed in our previous work [10], the planning method can prioritize searching for specific objects of interest by tuning the information-theoretic objective. However, in the case of semantic octree mapping with a range sensor, mutual information is not differentiable with respect to the pose \mathbf{X}_t^i . As a solution, we use the approach in [60] to obtain a differentiable approximation of mutual information by interpolating its values at nearby poses $\mathbf{V} \in SE(3)$. Specifically, the collision and informativeness score of \mathbf{X}_t^i is expressed as a convex combination of poses \mathbf{V} on a grid $\mathcal{X}(\mathbf{X}_t^i)$ inside a geodesic ball around \mathbf{X}_t^i with radius ξ_{\max} :

$$\begin{aligned} \mathfrak{f}(\mathbf{X}_t^i, p_t^i(\mathbf{m})) &= \sum_{\mathbf{V} \in \mathcal{X}(\mathbf{X}_t^i)} \lambda_{\mathbf{V}}(\mathbf{X}_t^i) s^i(\mathbf{V}), \\ s^i(\mathbf{V}) &= I(\mathbf{m}; \mathbf{z} | \mathbf{V}, p_t^i(\mathbf{m})) + \gamma_c \log D(\mathbf{V}, p_t^i(\mathbf{m})), \end{aligned}$$

where the safety constant $\gamma_c > 0$ trades off informativeness with collision avoidance and the convex combination coefficients $\lambda_{\mathbf{V}}(\mathbf{X}_t^i)$ adjust the influence of the terms corresponding to \mathbf{V} based on distance to \mathbf{X}_t^i :

$$\begin{aligned} \lambda_{\mathbf{V}}(\mathbf{X}_t^i) &= \frac{1 + \cos(\bar{d}(\mathbf{X}_t^i, \mathbf{V}))}{\sum_{\mathbf{U} \in \mathcal{X}(\mathbf{X}_t^i)} (1 + \cos(\bar{d}(\mathbf{X}_t^i, \mathbf{U}))),} \\ \bar{d}(\mathbf{X}_t^i, \mathbf{V}) &= \frac{\pi}{\xi_{\max}} d(\mathbf{X}_t^i, \mathbf{V}). \end{aligned}$$

Fig. 4 illustrates the collision and informativeness score \mathfrak{f} .

Cooperative planning requires the robots to take into account the plans of their peers in order to avoid actions that provide redundant information. Let $\mathfrak{X} = [\mathbf{X}_{t+1:t+T}^1, \dots, \mathbf{X}_{t+1:t+T}^{|\mathcal{V}|}]^\top \in SE(3)^{|\mathcal{V}| \times T}$ be the concatenated T -length trajectories of all robots in \mathcal{V} , where T is the planning horizon. In the remainder of this section, we use $\mathfrak{X}_{i,\tau}$

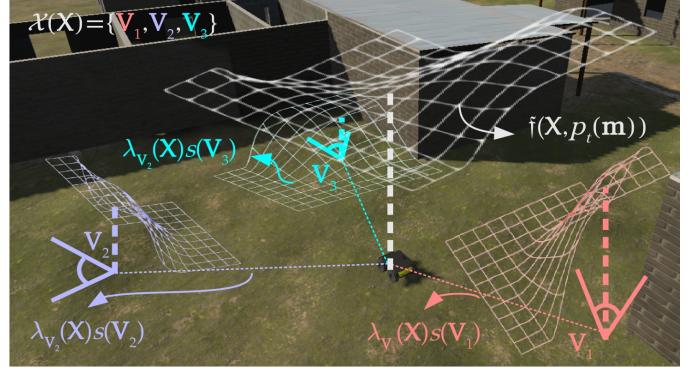


Fig. 4: Collision and informativeness score for robot pose \mathbf{X} . Each sampled viewpoint $\mathbf{V}_l \in \mathcal{X}(\mathbf{X})$ is colored differently. For each viewpoint, the field of view and the distance from the nearest obstacle determine the Shannon mutual information $I(\mathbf{m}; \mathbf{z} | \mathbf{V}_l, p_t(\mathbf{m}))$ and the log-distance $\log D(\mathbf{V}_l, p_t(\mathbf{m}))$, respectively. The weight $\lambda_{\mathbf{V}_l}(\mathbf{X})$ dictates the contribution of \mathbf{V}_l to the total score function $\mathfrak{f}(\mathbf{X}, p_t(\mathbf{m}))$, colored white.

as an alternative notation for $\mathbf{X}_{t+\tau}^i$, namely the $SE(3)$ pose of robot i at time $t+\tau$. The function $q(\mathfrak{X}_{i,\tau}, \mathfrak{X}_{j,\tau'})$ quantifies the observation redundancy as the overlap between sensor field of views (FoVs) for two poses $\mathfrak{X}_{i,\tau}$ and $\mathfrak{X}_{j,\tau'}$:

$$q(\mathfrak{X}_{i,\tau}, \mathfrak{X}_{j,\tau'}) = \max \{0, 2d_q - \|\mathbf{Q}(\mathfrak{X}_{i,\tau} - \mathfrak{X}_{j,\tau'})\mathbf{e}\|_2\}^2,$$

where:

$$d_q = |\mathcal{F}| + \xi_{\max}, \quad \mathbf{Q} = \begin{bmatrix} \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 3} & 0 \end{bmatrix}, \quad \mathbf{e} = \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ 1 \end{bmatrix},$$

and $|\mathcal{F}|$ is the diameter of the sensor FoV.

The local objective function for robot i is defined using the collision and informativeness score \mathfrak{f} and the FoV overlap q :

$$\begin{aligned} f^i(\mathfrak{X}, p_t^i(\mathbf{m})) &= \sum_{\tau=1}^T \left[\mathfrak{f}(\mathfrak{X}_{i,\tau}, p_t^i(\mathbf{m})) \right. \\ &\quad \left. - \gamma_q \sum_{j \in \mathcal{V}} \sum_{\tau'=1}^T [1 - \delta_{ij} \delta_{\tau\tau'}] [1 - \frac{\delta_{ij}}{2}] q(\mathfrak{X}_{i,\tau}, \mathfrak{X}_{j,\tau'}) \right], \end{aligned} \quad (16)$$

where δ_{ij} is the Kronecker delta which takes value 1 if and only if $i = j$, and 0 otherwise. Also, the constant $\gamma_q > 0$ trades off trajectory collision avoidance and informativeness with sensor FoV overlap.

The goal of multi-robot planning is to maximize the sum of local objective functions f^i over \mathcal{V} . Since we intend to perform the maximization in a distributed manner, we consider *local plans* $\mathfrak{X}^i \in SE(3)^{|\mathcal{V}| \times T}$ for each robot $i \in \mathcal{V}$, representing an individual robot's plan for the collective trajectories of the team. Eventually, these local plans should reach consensus so that the team members act in agreement. To quantify the disagreement among the robot plans, we define an aggregate distance function $\phi(\cdot) : SE(3)^{|\mathcal{V}| \times T \times |\mathcal{V}|} \rightarrow \mathbb{R}_{\geq 0}$ that accumulates the pairwise distances between all local plans \mathfrak{X}^i , $i \in \mathcal{V}$:

$$\phi(\mathfrak{X}^{1:|\mathcal{V}|}) = \sum_{\{i,j\} \in \mathcal{E}} A_{ij} d^2(\mathfrak{X}^i, \mathfrak{X}^j), \quad (17)$$

where $d : SE(3)^{|\mathcal{V}| \times T \times 2} \rightarrow \mathbb{R}_{\geq 0}$ is defined via extension of the distance in $SE(3)$ to the product manifold $SE(3)^{|\mathcal{V}| \times T}$.

Algorithm 3 Distributed Planning for Exploration

Input: Local map $p_t^i(\mathbf{m})$ of robot i
Output: Collaborative robot team plan for exploration

- 1: $\mathfrak{X}^{i(0)} = \text{FRONTIER}(p_t^i(\mathbf{m})) \quad \forall i \in \mathcal{V} \quad \triangleright \text{Initialization}$
- 2: **for** $k \in \mathbb{Z}_{\geq 0}$ **do**
- 3: $\triangleright \text{Promote consensus with step size } \epsilon_p:$
- 4: **for** every $l \in \mathcal{V}$ and $\tau \in \{1, \dots, T\}$ **do**
- 5: $\tilde{\mathfrak{X}}_{l,\tau}^{i(k)} = \mathfrak{X}_{l,\tau}^{i(k)} \exp \left(\epsilon_p \sum_{j \in \mathcal{N}_i} A_{ij} \right.$

$$\left. \times (J_L^{-\top}(\boldsymbol{\xi}_{\tilde{\mathfrak{X}}_{l,\tau}^{i(k)}, \tilde{\mathfrak{X}}_{l,\tau}^{i(k)}}) \Gamma \boldsymbol{\xi}_{\tilde{\mathfrak{X}}_{l,\tau}^{i(k)}, \tilde{\mathfrak{X}}_{l,\tau}^{i(k)}})^\wedge \right)$$
- 6: $\triangleright \text{Local gradient computation:}$
- 7: $\mathbf{g}_{l,\tau} = 0 \quad \triangleright \text{Initialize for all } l \in \mathcal{V}, \tau \in \{1, \dots, T\}$
- 8: **for** every $\tau' \in \{1, \dots, T\}$ **do**
- 9: $c_{\text{set}} = |\mathcal{X}(\tilde{\mathfrak{X}}_{i,\tau'})| + \sum_{\mathbf{V} \in \mathcal{X}(\tilde{\mathfrak{X}}_{i,\tau'})} \cos(\bar{d}(\tilde{\mathfrak{X}}_{i,\tau'}, \mathbf{V}))$
- 10: $\mathbf{g}_{i,\tau'} += \sum_{\mathbf{V} \in \mathcal{X}(\tilde{\mathfrak{X}}_{i,\tau'})} [(s(\mathbf{X}) - \mathfrak{f}(\tilde{\mathfrak{X}}_{i,\tau'}, p_t^i(\mathbf{m})))$

$$\times \frac{\sin(\bar{d}(\tilde{\mathfrak{X}}_{i,\tau'}, \mathbf{V}))}{c_{\text{set}} \bar{d}(\tilde{\mathfrak{X}}_{i,\tau'}, \mathbf{V})} J_L^{-\top}(\boldsymbol{\xi}_{\tilde{\mathfrak{X}}_{i,\tau'}, \mathbf{V}}) \Gamma \boldsymbol{\xi}_{\tilde{\mathfrak{X}}_{i,\tau'}, \mathbf{V}}]$$
- 11: **for** every $l \in \mathcal{V}$ and $\tau \in \{1, \dots, T\}$ **do**
- 12: $\mathbf{p}_{i,\tau'} = \mathbf{Q} \tilde{\mathfrak{X}}_{i,\tau'}^{i(k)} \mathbf{e}, \quad \mathbf{p}_{l,\tau} = \mathbf{Q} \tilde{\mathfrak{X}}_{l,\tau}^{i(k)} \mathbf{e}$
- 13: $\mathbf{R}_{i,\tau'} = \mathbf{Q} \tilde{\mathfrak{X}}_{i,\tau'}^{i(k)} \mathbf{E}, \quad \mathbf{R}_{l,\tau} = \mathbf{Q} \tilde{\mathfrak{X}}_{l,\tau}^{i(k)} \mathbf{E}$
- 14: $\mathbf{c}_{\text{disp}} = [1 - \delta_{il} \delta_{\tau\tau'}] [1 - \frac{\delta_{il}}{2}] (\mathbf{p}_{i,\tau'} - \mathbf{p}_{l,\tau})$
- 15: $\mathbf{c}_{\text{tot}} = \gamma_{\mathbf{q}} \mathbf{c}_{\text{disp}} \max \left\{ 0, \frac{2d_{\mathbf{q}}}{\|\mathbf{p}_{i,\tau'} - \mathbf{p}_{l,\tau}\|_2} - 1 \right\}$
- 16: $\mathbf{g}_{i,\tau'} += \begin{bmatrix} \mathbf{R}_{i,\tau'}^\top \mathbf{c}_{\text{tot}} \\ \mathbf{0}_{3 \times 1} \end{bmatrix}, \quad \mathbf{g}_{l,\tau} -= \begin{bmatrix} \mathbf{R}_{l,\tau}^\top \mathbf{c}_{\text{tot}} \\ \mathbf{0}_{3 \times 1} \end{bmatrix}$
- 17: $\triangleright \text{Apply gradient with step size } \alpha_p^{(k)}$
- 18: **for** every $l \in \mathcal{V}$ and $\tau \in \{1, \dots, T\}$ **do**
- 19: $\tilde{\mathfrak{X}}_{l,\tau}^{i(k+1)} = \tilde{\mathfrak{X}}_{l,\tau}^{i(k)} \exp(\alpha_p^{(k)} \mathbf{g}_{l,\tau})$
- 20: **return** $\mathfrak{X}^{i(k)}$

Problem 3. Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be a network of robots, where each robot $i \in \mathcal{V}$ maintains a local map $p_t^i(\mathbf{m})$ obtained by solving (15). Determine $SE(3)$ pose trajectories for all robots that maximize the cost function in (16) subject to the consensus constraint in (17):

$$\begin{aligned} \max_{\mathfrak{X}^{1:|\mathcal{V}|} \in SE(3)^{|\mathcal{V}| \times T \times |\mathcal{V}|}} & \sum_{i \in \mathcal{V}} f^i(\mathfrak{X}^i, p_t^i(\mathbf{m})), \\ \text{s.t.} & \phi(\mathfrak{X}^{1:|\mathcal{V}|}) = 0. \end{aligned} \quad (18)$$

The structure of the planning problem in (18) is compatible with the distributed Riemannian optimization method of Sec. III. We formulate a version of Alg. 1 specialized for the $SE(3)$ manifold. Due to the positive curvature of the $SE(3)$ manifold, the aggregate distance function $\phi(\cdot)$ has local minima (see Appendix A.3 in [61]). Thus, if the initial trajectories $\mathfrak{X}^{i(0)}$, $i \in \mathcal{V}$, are not similar, the algorithm may converge to a local optimum of the consensus constraint (17). Furthermore, the local objective functions f^i , $i \in \mathcal{V}$, are only locally concave [62]. Hence, Theorem 1 guarantees only a locally optimal consensus solution.

Our distributed planning algorithm for solving (18) is presented in Alg. 3. Given its current local map $p_t^i(\mathbf{m})$, each

robot $i \in \mathcal{V}$ computes an initial plan $\mathfrak{X}^{i(0)}$ for the whole team using frontier-based exploration [63]. In line 5, each pose in the local plan $\mathfrak{X}^{i(k)}$ is guided towards consensus with the plans of neighboring robots $j \in \mathcal{N}_i$. The update in this line is carried out via a perturbation in the robot frame, where $J_L(\cdot)$ denotes the left Jacobian of $SE(3)$ [59, Ch.7], and only involves communication between neighbors. To compute the local objective function gradients with respect to each pose in the local plan, we first initialize the gradients with zero in line 7, and then populate them with proper values in lines 10 and 16. The gradient of the collision and informativeness score $\mathfrak{f}(\tilde{\mathfrak{X}}_{i,\tau'}, p_t^i(\mathbf{m}))$ is computed in lines 9-10, while lines 12-16 derive the gradient of the sensor overlap $\mathfrak{q}(\tilde{\mathfrak{X}}_{i,\tau'}, \tilde{\mathfrak{X}}_{l,\tau'})$ with respect to both inputs, where $\mathbf{E} = [\mathbf{I}_{3 \times 3} \ \mathbf{0}_{3 \times 1}]^\top$. Note that, for the \mathfrak{f} terms, we only need to compute the gradient with respect to robot i 's own trajectory $\tilde{\mathfrak{X}}_{i,\tau'}$, $\tau' \in \{1, \dots, T\}$, whereas for the \mathfrak{q} terms, robot i should locally obtain gradients with respect to both its own trajectory as well as the trajectories of all other robots in \mathcal{V} . Since each robot stores the trajectory of the whole team, the computation for the gradients of the \mathfrak{q} terms does not require any communication among the robots. Lastly, in line 19, we apply the computed gradients to each pose in the local plan, using a right perturbation in the robot frame.

Solving (18) via Alg. 3 leads to two types of behaviors.

- 1) Locally, the robots attempt to maximize information and distance from obstacles along their trajectories. This encourages each robot to visit unvisited parts of the environment, and corresponds to the \mathfrak{f} terms of (16).
- 2) Within each neighborhood, the robots *negotiate* with their peers to minimize redundant observations. This prevents the trajectories to amass at certain regions of the map, and corresponds to the \mathfrak{q} terms of (16).

We emphasize that each local plan \mathfrak{X}^i stores the paths for all robots in \mathcal{V} , instead of only robot i 's and its immediate neighbors. This is because storing all $|\mathcal{V}|$ paths in each robot allows propagation of the mentioned behaviors on a global scale, due to the consensus constraint of (18). Therefore, the global solution of (18) corresponds to a Pareto optimum where agents find an optimal trade-off between their own information and safety maximization on one hand and avoiding observation overlap with their peers on the other hand.

In this section, we developed the distributed planning component of ROAM. The robot trajectories are chosen to maximize information and safety for cooperative estimation of a semantic octree map. Combined with the distributed mapping method of Sec. IV, the overall system can be utilized for efficient multi-robot exploration of an unknown environment. In the next section, we demonstrate the performance of ROAM in a variety of simulation and real-world experiments.

VI. EXPERIMENTS

This section describes the implementation of ROAM on multi-robot systems. Next, we evaluate the performance of ROAM using several measures that quantify optimality, convergence to consensus, and communication overhead. The evaluations are done in both simulation and real-world.

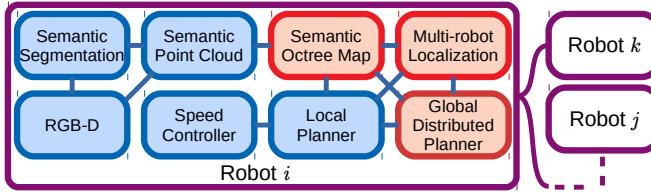


Fig. 5: Software stack for multi-robot distributed active mapping. The blue blocks are local to each robot, whereas the red blocks require communication with neighboring robots. In particular, the robots broadcast their local octree map, computed team path, and estimated pose. The communication links between pairs of robots are represented by violet lines.

TABLE I: Parameter set for multi-robot exploration.

Planning			Mapping	
ϵ_p	0.1	$\alpha_p^{(k)}$	$\frac{0.1}{k+1}$	ϵ_m
d_q	20m	ξ_{\max}	16m	$\alpha_m^{(k)}$
γ_c	10^{-3}	γ_q	10^{-2}	t_m^{pub}
T	5	k_p	20	t_m^{int}
t_p^{pub}	1sec	$thresh_p$	0.4	Voxel
Γ	$diag(1, 1, 1, 0.1, 0.1, 0.1)$		side length	0.2m

A. Implementation of ROAM for Distributed Active Mapping

We deploy our approach on a team of ground wheeled robots, each equipped with an RGB-D sensor. Fig. 5 shows an overview of the software stack, implemented using the *Robot Operating System* (ROS). The RGB-D sensor provides synchronized RGB and depth images. The RGB image is processed with a semantic segmentation algorithm to label each pixel with an object category. The segmented image is fused with the depth image to obtain a 3-D semantically annotated point cloud in the sensor frame of robot i .

Multi-robot localization: It is required to perform multi-robot localization in order to find the transformation a) from robot i 's sensor frame to a static *world* frame \mathcal{W}_i for point cloud registration, and b) from \mathcal{W}_i to \mathcal{W}_j for distributed multi-robot mapping and planning. Our implementation of multi-robot localization in the simulation and real-world experiments is explained in Sec. VI-B and Sec. VI-D, respectively.

Multi-robot mapping: The semantic point cloud is used to build and update a semantic octree map for each robot $i \in \mathcal{V}$ via lines 6-12 of Alg. 2. The semantic map of each robot i is broadcasted to its neighboring robots \mathcal{N}_i once every t_m^{pub} seconds. Moreover, each robot pushes any newly received map to a local buffer memory, and performs line 4 of Alg. 2 every t_m^{int} seconds to integrate neighbors' maps into its local map. The buffer is cleared after each successful iteration of Alg. 2.

Multi-robot viewpoint planning: To decouple low-frequency informative planning from high-frequency planning for collision-avoidance, we perform two separate planning stages, namely on global viewpoint level and on local trajectory level. On the viewpoint level, the distributed collaborative planning in Alg. 3 is employed to find informative viewpoints for each robot in \mathcal{V} . To coordinate viewpoint planning across all robots, every robot $i \in \mathcal{V}$ maintains a ledger \mathcal{L} composed of $|\mathcal{V}|$ binary values each indicating whether the corresponding robot in the team is ready for planning. Due to the decentral-

Algorithm 4 Distributed Ledger Synchronization

Input: Incoming ledger \mathcal{L}_{inc}

Output: Synchronized ledger

```

1:  $\mathcal{L}_i = \mathcal{L}_{inc}$ 
2: if CHECKREADY() then
3:    $\mathcal{L}_i[i] = 1$ 
4: if MEAN( $\mathcal{L}_i$ )  $\geq thresh_p$  then
5:   STARTPLANNING()  $\triangleright$  Viewpoint planning via Alg. 3
6: else
7:    $\mathcal{L}_i[i] = ISPLANNING()$ 
8: return  $\mathcal{L}_i$ 

```

ized nature of our method, each robot sends its own copy of the ledger \mathcal{L}_i to its neighbors every t_p^{pub} second, and updates \mathcal{L}_i using the incoming ledgers, as well as its status with respect to the current plan. Alg. 4 details the process of decentralized ledger synchronization for each robot i . In line 1 robot i makes a copy of the incoming ledger \mathcal{L}_{inc} . Then, in line 2, the function CHECKREADY() determines whether or not the robot is ready to compute a new plan. A robot would declare ready to plan only when it has finished its previous plan and also it is currently not planning. The global distributed planning of Alg. 3 would start only after a minimum fraction of robots, denoted by $thresh_p$, are ready to plan. Line 7 is used to stabilize the ledger synchronization process. During the global distributed planning of Alg. 3, each robot i broadcasts its local plan \mathfrak{X}^i after each optimization iteration. Incoming local plans \mathfrak{X}^j , $j \in \mathcal{N}_i$, are pushed to a local buffer memory to be used during the consensus step (line 5 of Alg. 3). The buffer is cleared after each optimization iteration. Lastly, the planning terminates after reaching k_p iterations.

Local trajectory optimization: After computing a sequence of viewpoints $\mathfrak{X}^{1:|\mathcal{V}|}$, each robot i locally computes a trajectory to visit its portion of the viewpoints $\mathfrak{X}_{i,1:T}^i$. The separation of the viewpoint planning from the trajectory optimization allows the robots to rapidly react to environment changes or mapping errors via local path re-planning, without the need to coordinate with their peers in viewpoint planning via Alg. 3. Furthermore, the two stage planning allows accounting for dynamical constraints of each robot in heterogeneous robot teams, such that the low-level trajectory optimizer takes the viewpoint set $\mathfrak{X}_{i,1:T}^i$ and computes a dynamically feasible path. In our experiments, each robot i projects its own semantic octree map onto a 2-D plane to obtain an occupancy grid map of the environment. Given its viewpoint set $\mathfrak{X}_{i,1:T}^i$, the robot computes a sequence of collision-free positions and orientations that connect its current pose to $\mathfrak{X}_{i,1}^i$, and each $\mathfrak{X}_{i,\tau}^i$ to $\mathfrak{X}_{i,\tau+1}^i$ for $\tau \in \{1, \dots, T-1\}$. For this purpose, the trajectory optimizer uses A^* graph search over the 2-D occupancy map. If a collision is detected during execution of the path, the corresponding path segment is re-planned using another A^* call. The local trajectory is then used by a low-level speed controller to generate velocity commands.

An open-source implementation of ROAM is available on GitHub². The rest of this section describes the simulation and

²<https://github.com/ExistentialRobotics/ROAM>.



Fig. 6: Simulation environment for multi-robot distributed active mapping: A Husky robot receiving RGB, depth, semantic segmentation images. A top-down view of the simulated environment, where the numbered circles show the starting positions of six robots is shown on the right.

real-world experiments. Table I summarizes the parameters used across all experiments.

B. Outdoor Simulation Experiments in Unity

We carry out experiments in a photo-realistic 3-D simulation powered by the Unity engine. The environment resembles an outdoor village area with various types of terrain (e.g., grass, dirt road, asphalt, etc.) and object classes, such as buildings, cars, and street lighting. Our experiments utilize $|\mathcal{V}| = 6$ ClearPath Husky wheeled robots, each equipped with an RGB-D sensor. We assume known robot poses and perfect semantic segmentation over the RGB input in the simulation experiments. Fig. 6 shows the simulation setup.

Each robot uses its local semantic octree map to extract traversable regions, while other object and terrain classes are considered as obstacles. In particular, *Asphalt* and *Dirt road* classes are selected as traversable terrain classes. Fig. 7 visualizes a time lapse of the distributed multi-robot active mapping experiment. The consistency between the local map of robot 1 and the combined map of all robots can be seen as a qualitative example of the map consensus achieved by the distributed mapping method in Alg. 2. Analogously, Fig. 8 illustrates consensus achieved by the distributed multi-robot planning in Alg. 3. As described in Sec. V, each robot computes its local plan based on its local map. Hence, differences in the local maps can cause variation across the local plans, as seen in Fig. 8a. However, during each iteration of distributed planning, line 5 in Alg. 3 steers the local plans towards a consensus plan, as is evident in Fig. 8d.

The performance of ROAM is evaluated quantitatively under various robot network configurations and planning parameters. We consider 3 different network topologies: 1) *Full*, where all robots can communicate with each other in a fully-connected network, 2) *Hierarchical*, where robots can only communicate with their team leaders, and 3) *Ring*, where each robot has exactly 2 neighbors. Fig. 9 depicts the 3 network configurations. For each network configuration, we perform exploration under 3 variants of Alg. 3: 1) *Collaborative*, which is the original version of Alg. 3, 2) *Egocentric*, where each robot only maximizes its own path informativeness and safety by

choosing $\epsilon_p = \gamma_q = 0$, and 3) *Frontier*, where robots perform frontier-based exploration by choosing $k_p = 0$.

Fig. 10 quantifies the coverage achieved by each network topology and planning parameter set. For *Collaborative* and *Egocentric* planning configurations, *Full* network configuration leads to faster coverage while traveling less distance compared to *Hierarchical* and *Ring* topologies. This is expected since *Full* is the only network topology that allows one-hop exchange of information between any pair of robots. On the other hand, the network configuration does not play a significant role for *Frontier* exploration in terms of total covered area, since robots usually choose a frontier that is nearby their current position, and do not utilize information coming from their peers' local maps. The most interesting takeaway from Fig. 10 is the similar performance of *Collaborative* planning with *Hierarchical* and *Ring* topologies, compared to *Egocentric* planning with *Full* topology. This observation suggests that effective coordination among agents via *Collaborative* planning can alleviate the longer multi-hop communication routes caused by the sparse connectivity of *Hierarchical* and *Ring* topologies.

Similar insights can be obtained from Fig. 11, where normalized map entropy is measured against elapsed time and distance traveled, for each network topology and planning mode. Normalized map entropy for robot $i \in \mathcal{V}$ is defined as the sum of Shannon entropies of all map voxels divided by the number of voxels:

$$H_{norm}^i = \frac{-1}{N^i} \sum_{n=1}^{N^i} \sum_{c \in \mathcal{C}} p_n^i(m=c) \log p_n^i(m=c),$$

where N^i denotes the number of voxels in the local map of robot i , and \mathcal{C} as well as $p_n^i(m)$ are defined in the previous sections. Unlike total map entropy, normalized entropy can increase as the robots register unvisited voxels into their map. As Fig. 11 shows, for each planning mode, *Full* network topology outperforms *Hierarchical* and *Ring* configurations. Also, *Collaborative* planning with *Hierarchical* and *Ring* configurations have similar performance to *Egocentric* planning with *Full* network topology. The same reasoning used for Fig. 10 can be utilized to justify these observations. However, unlike coverage, network topology plays a more significant role in terms of normalized map entropy for *Frontier* planning mode. This is due to the relatively more distributed mapping consensus steps for the *Full* topology that lead to more certainty in the map estimation and, hence, smaller entropy compared to *Hierarchical* and *Ring*. Since coverage does not take map uncertainty into account, such behavior is only noticeable in the right column of Fig. 11 but not in Fig. 10.

Additional quantitative metrics specific to multi-robot exploration are reported in Fig. 12. The first row of Fig. 12 shows the aggregate distance $\phi(\mathbf{h}^{1:6})$, which represents the total discrepancy across all local maps. Despite robots discovering distinct unexplored regions during exploration, which can increase the difference among the local maps, it can be seen that the map discrepancy tends to decrease overall. The long-term value of the map discrepancy depends on the ratio of exploration rate and information exchange rate. Hence, the

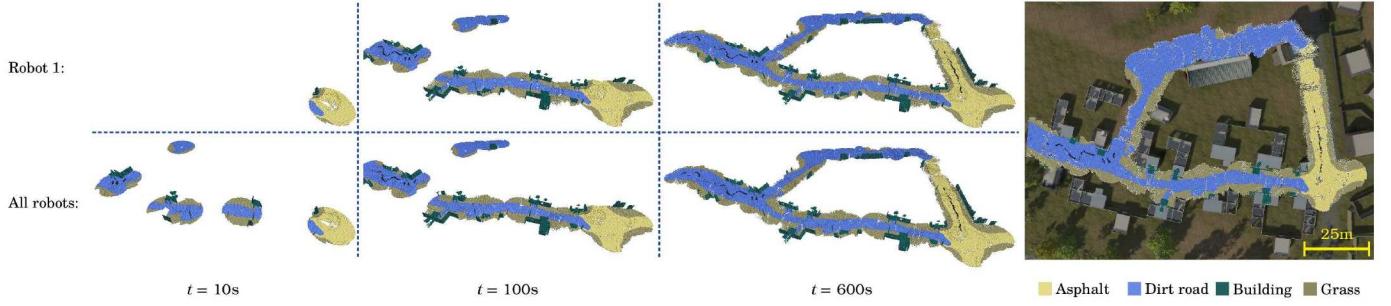


Fig. 7: Time lapse of the multi-robot active mapping experiment. The local map of robot 1 (in Fig. 6) is compared against the combined maps of all robots. The right sub-figure shows the estimated semantic octree map of robot 1 overlaid on the ground-truth simulation environment. The exploration is carried out using a fully-connected network of robots.

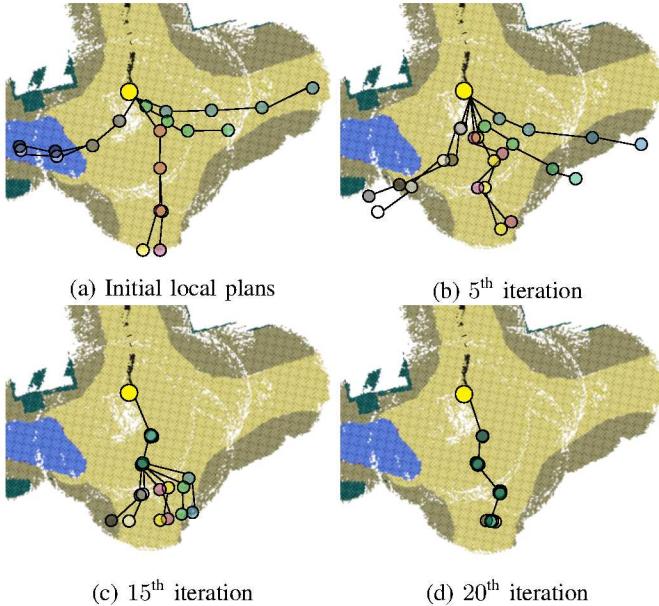


Fig. 8: Time lapse of viewpoint planning for robot 1 from Fig. 6. Each color corresponds to a robot $j \in \mathcal{V}$ computing a local plan $\mathfrak{X}_{1,1:T}^j$ for robot 1. The planned trajectories contain both position and orientation, however only the positions are visualized for clarity. The planning is carried out over a fully-connected network of robots.

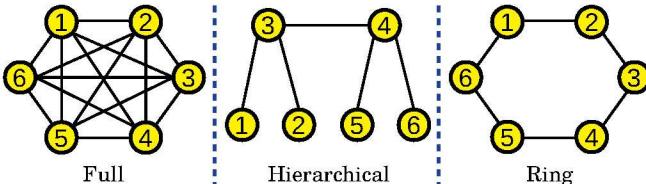


Fig. 9: Network topologies used in the simulation experiments.

Full topology yields the closest performance to map consensus due to its relatively faster rate of exchanging the local maps amongst the robots. The second row of Fig. 12 displays the bandwidth required for communicating the local maps within the robot network. Since the simulation environment uses a centralized network scheme to register the broadcasted local maps, there is no significant variation in terms of bandwidth use across different network topologies and planning modes. Nevertheless, the results show scalability of the semantic

octree mapping for multi-robot applications, where an average 97 Bytes/sec of bandwidth is needed for each 1 m^2 of covered area for voxel dimensions of $0.2 \times 0.2 \times 0.2 \text{ m}^3$.

The quantitative results of Fig. 10 and Fig. 11 demonstrate the effective performance of ROAM, while Fig. 12 showcases the consensus and communication properties of our method. In the next subsection, we evaluate the distributed multi-robot active mapping in real-world experiments.

C. Indoor Simulation Experiments in Gazebo

We also evaluated ROAM in a Gazebo simulation of an office environment with corridors and dead-ends, shown in Fig. 13. A team of $|\mathcal{V}| = 4$ ClearPath Jackal robots was used for multi-robot active mapping. We compared ROAM versus the 3D Multi-Robot (3DMR) exploration method [37]. Multi-robot planning in 3DMR is carried out using a two-level next-best-view approach, where redundant visitations and inter-robot collisions are handled in separate stages. The multi-robot mapping in 3DMR is based on occupancy octree mapping [64], and synchronization is done by sending missing sensor observations between robots when local map estimates diverge more than a pre-defined threshold.

A comparison of exploration trajectories is shown in Fig. 14, where both methods use *Full* network configuration, and *Collaborative* planning parameters are used for ROAM. In terms of exploration performance, we observe that ROAM tends to provide more sophisticated trajectories with diverging paths among the robots, ensuring every unexplored region of the environment is visited once. On the other hand, the paths taken by 3DMR are smoother, primarily oriented towards the frontiers of the known regions, and have more overlapping observations among the robots. These differences can be explained by the fact that ROAM is a continuous-space trajectory optimization method capable of optimizing small trajectory perturbations to maximize coverage, while 3DMR selects the best viewpoints using a discrete-space sampling method.

Quantitative evaluation of ROAM and 3DMR is reported in Fig. 15. In terms of coverage, ROAM achieves the same level of exploration by traversing relatively shorter distances. Similar to the analysis above, this can be attributed to the high information gain of ROAM paths in which each viewpoint along the trajectory is optimized for maximal coverage, whereas in 3DMR only the end pose of the trajectory maximizes visibility. Moreover, as seen in the bottom row

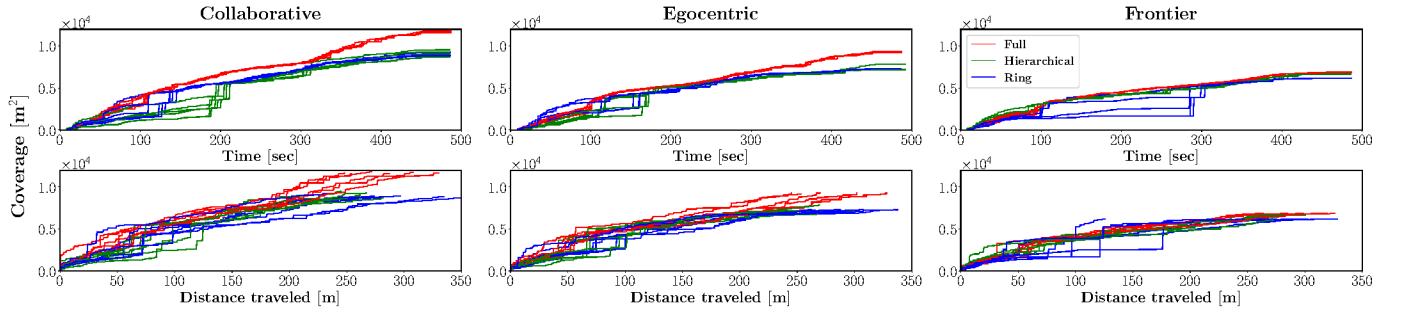


Fig. 10: Coverage versus time (top row) and distance traveled (bottom row) for the simulation experiments. In each plot, lines with the same color correspond to robots participating in the same multi-robot exploration experiment, while the experiments are separated by the type of network topology in Fig. 9.

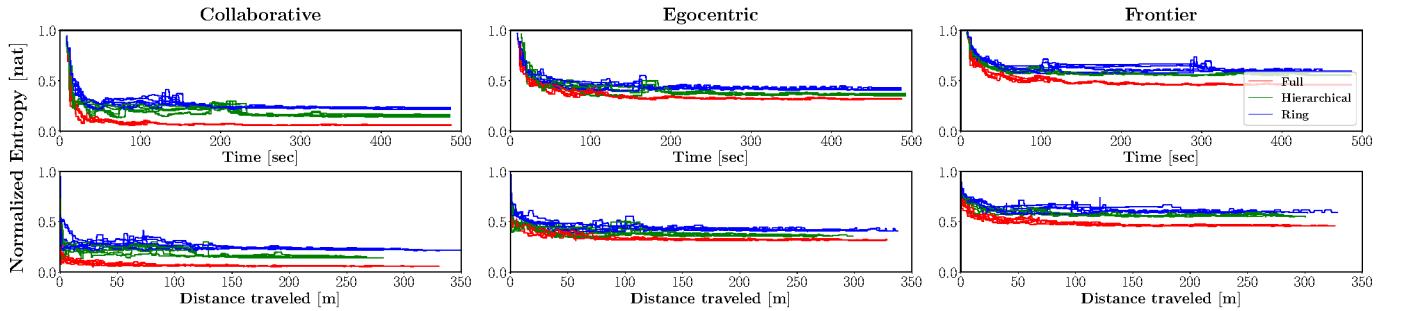


Fig. 11: Normalized map entropy versus time (top row) and distance traveled (bottom row) for the simulation experiments. In each plot, lines with the same color correspond to robots participating in the same multi-robot exploration experiment, while the experiments are separated by the type of network topology in Fig. 9.

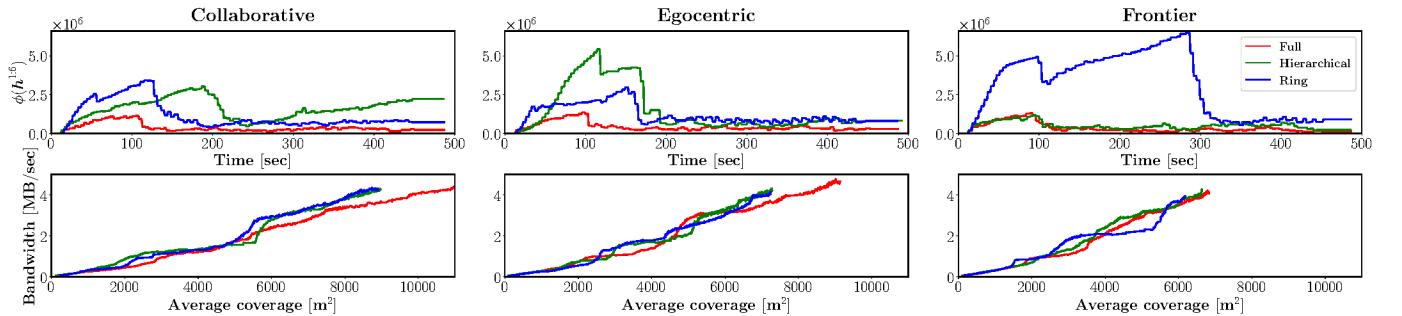


Fig. 12: Multi-robot exploration performance metrics in the simulation experiments. The top row shows evolution of the map discrepancy $\phi(h^{1:6})$ across the robot networks \mathcal{G} in Fig. 9 over time, and the bottom row displays bandwidth requirements for the distributed mapping with respect to average coverage. The average coverage is computed by averaging the area covered at each timestamp over all robots participating in an experiment.

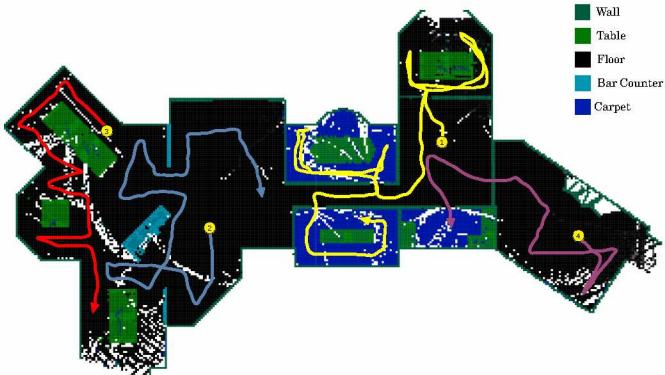
of Fig. 15, the threshold-based map synchronization rule of 3DMR leads to a large initial map discrepancy among the robots. With regards to bandwidth requirements, 3DMR utilizes a consistent level of usage, which is at the beginning significantly larger than ROAM, but is eventually surpassed by ROAM's bandwidth usage. This is due to the fact that 3DMR directly broadcasts sensor measurements, which have large but near constant memory footprint. Conversely, in ROAM each robot broadcasts its local octree map, which has small memory footprint but grows with the covered area. Using event-triggered and difference-based map synchronization can alleviate the almost linear bandwidth utilization of ROAM but this is outside the scope of this work.

D. Real-World Experiments

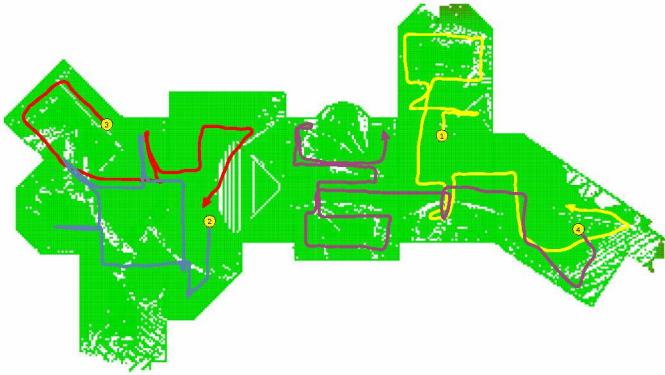
We deployed ROAM on a team of ground robots to achieve autonomous exploration and mapping of an unknown indoor area. The robot team was comprised of two ClearPath Jackal robots (robot 1 and robot 2), and an F1/10 race car robot (robot 3). The Jackals were each equipped with an Ouster OS1-32 3D LiDAR, an Intel RealSense D455 RGB-D camera, and an NVIDIA GTX 1650 GPU. The F1/10 race car was equipped with a Hokuyo UST-10LX 2D LiDAR, an Intel RealSense D455 RGB-D camera, and an NVIDIA Xavier NX computer. Fig. 16 shows the three robots participating in the experiment. We utilize a ResNet18 [65] neural network architecture pre-trained on the SUN RGB-D dataset [66] for semantic segmentation. To achieve real-time segmentation, we



Fig. 13: Indoor office environment in the Gazebo simulator used in our comparative analysis.



(a) Multi-class octree map estimated by ROAM



(b) Occupancy octree map estimated by 3DMR

Fig. 14: Estimated map and exploration paths taken by each robot for ROAM (top) and 3DMR (bottom). For both methods, multi-robot exploration was run for $t = 300$ s.

employed the deep learning inference ROS nodes provided by NVIDIA [67], which are optimized for NVIDIA GPUs via TensorRT acceleration. The semantic segmentation module processes the RGB image stream from the D455 camera, and fuses the segmentation results with the depth image stream, to publish semantic point cloud ROS topics. For localization, the Jackal robots used the direct LiDAR odometry of [68], while the F1/10 race car used iterative closest point (ICP) scan matching [69]. In order to align the world frames \mathcal{W}_i , $i \in \{1, 2, 3\}$, we used AprilTag detection [70], where the estimated $SE(3)$ transformation between the RGB sensor frame of a robot and the detected tag is defined as the *world-to-*

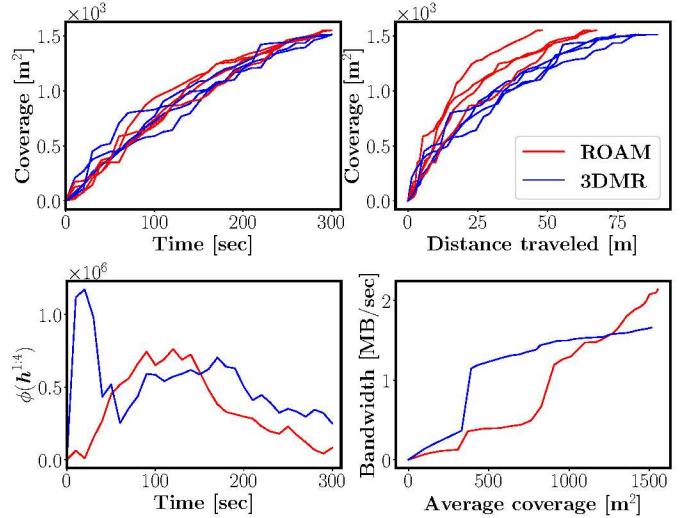


Fig. 15: Quantitative evaluation of multi-robot exploration for ROAM and 3DMR in the simulated office environment in Fig. 13. The top row show map coverage versus time and distance traveled, while the bottom row shows map discrepancy $\phi(h^{1:4})$ and bandwidth utilization.



Fig. 16: Ground robot team used in our real-world multi-robot active mapping experiments.

sensor transformation. Communication was handled via a Wi-Fi network and multi-master ROS architecture, such that each robot i runs its own ROS master, and shares its planning ledger \mathcal{L}_i , local plan \mathfrak{X}_i , local semantic octree map \mathbf{m}^i , and estimated pose with respect to \mathcal{W}_i . With all the mapping and planning computations carried out using the on-board robot computers, we obtain an average frame rate of 2.61Hz, 2.58Hz, and 2.13Hz for robots 1-3, respectively, for distributed semantic octree mapping. We observed an average distributed planning iteration time of 0.009s, 0.009s, and 0.024s for robots 1-3.

Our experiments took place in a basement area, consisting of a lobby room connected to a corridor and a large laboratory, shown in Fig. 17. Similar to the simulation experiments, the local semantic octree maps were utilized to analyze the terrain traversability, where in this case the *Ground* object class was selected as the traversable region. Exploration was performed using the *Collaborative* planning parameter set. The communication network topology was *Full*, however, there were intermittent disconnections due to signal attenuation and occlusion by the walls. Fig. 18 shows a time lapse of the real-world multi-robot active mapping experiment. The robots started at nearby positions, all facing the same AprilTag to align their world frames. During the first 400s of exploration,

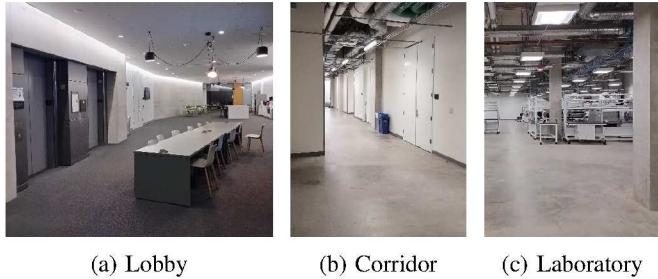


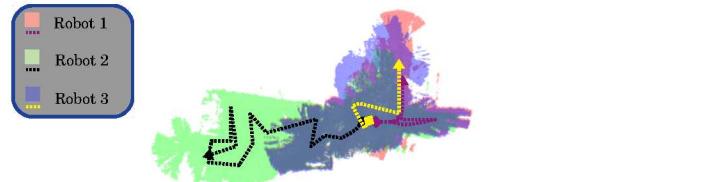
Fig. 17: Indoor environment used in our real-world multi-robot active mapping experiments.

the robots mostly explored the same areas in their immediate vicinity. Each robot gradually separated from the others after $t = 400$ s, and focused on a specific part of the environment, as it can be seen in Fig. 18a. In particular, robot 1 explored the corridor area, while robot 2 and robot 3 visited the laboratory and the lobby, respectively. Around $t = 500$ s, as the robots got farther away from each other, they temporarily lost communication. During the disconnection period, the robots could not plan collaboratively and relied only on their local maps for planning. Such cutoffs caused delayed consensus steps, leading to deviation of the map estimations and the planned paths across the robot team. Extended periods of communication interruption could lead to inefficient exploration because the robots may re-visit already explored areas when they are unaware of their peers' map estimates. Nonetheless, this did not deteriorate the exploration performance in our case since the robots were so far away from each other that they would not revisit regions already explored by a peer. Fig. 18b is a good example of such situations. At $t = 1200$ s, the team was ordered to return to base, where the robots individually planned paths from their current positions back to their initial positions. Communication was automatically re-established as soon as the robots arrived near the starting locations, and distributed mapping was resumed leading to agreement in the semantic octree maps, as shown in Fig. 18c. First-person views of the semantic octree map of robot 1 overlaid on the real-world environment is illustrated in Fig. 18d.

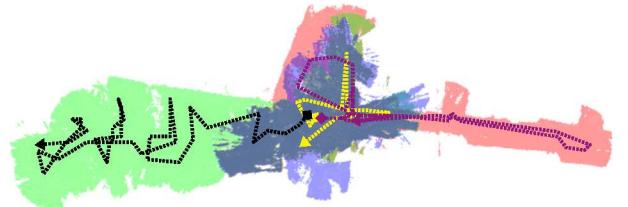
Fig. 19 and Fig. 20 show the quantitative results of the same real-world experiment. Note the sudden jump in the map coverage plot of Fig. 19, corresponding to reaching map consensus amongst the robots when network connection was re-established. This can be clearly observed in Fig. 20 where the map discrepancy $\phi(h^{1:3})$ shrinks to zero as the robots reach consensus. Unlike the simulation experiments, the bandwidth use of the robots can be measured separately by probing the communication packages sent to/from each robot, reported in Fig. 20. The flat lines represent the jump in map coverage due to the eventual connection after a period of intermittent disconnections during the exploration. Overall, the real-world experiments show the practicality of ROAM for autonomous mapping of large unstructured areas using a team of robots and consumer-grade communication infrastructure.

VII. CONCLUSIONS

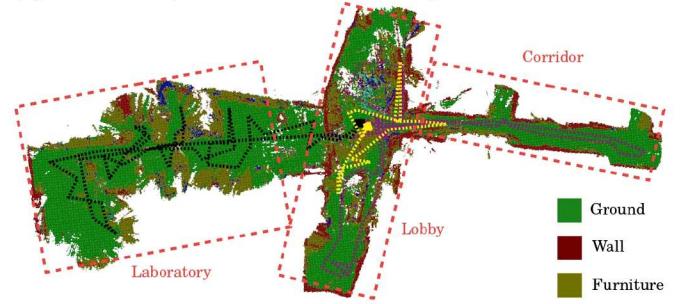
We developed a distributed Riemannian optimization method that achieves consensus among the variables estimated



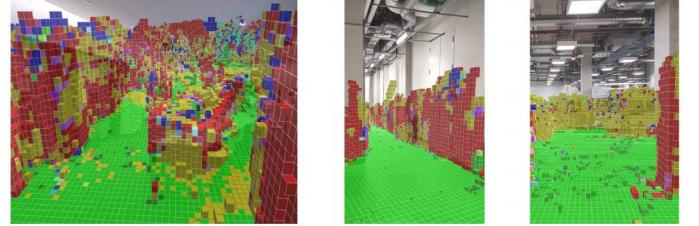
(a) At $t = 400$ s, the robots effectively choose different sections of the environment to explore.



(b) At $t = 900$ s, the robots continue to explore the environment.



(c) At $t = 1450$ s, the robots are back to their initial locations and perform a final map exchange to ensure consensus.



(d) First-person view of the semantic octree map of robot 1.

Fig. 18: Qualitative results from a real-world multi-robot active mapping experiment. (a)-(b): Snapshots of exploration at $t = 400$ s and $t = 900$ s. The explored region and the path of each robot are identified by a distinct color (legend at the top left). (c): The final semantic octree map of robot 1 at $t = 1450$ s. The three sections of the environment (i.e. lobby, laboratory, and corridor) have been marked on the map. (d) First-person views of the map overlaying the ground-truth environment.

by different nodes in a communication graph. We used this method to formulate distributed techniques for multi-robot semantic mapping and information-theoretic viewpoint planning. The resulting Riemannian Optimization for Active Mapping (ROAM) enables fully distributed collaborative active mapping of an unknown environment without the need for central estimation and control. Our experiments demonstrated scalability and efficient performance even with sparse communication, and corroborated the theoretical guarantees of ROAM to achieve consensus and convergence to an optimal solution. ROAM offers the possibility to generalize many single-robot non-Euclidean problems to distributed multi-robot applica-

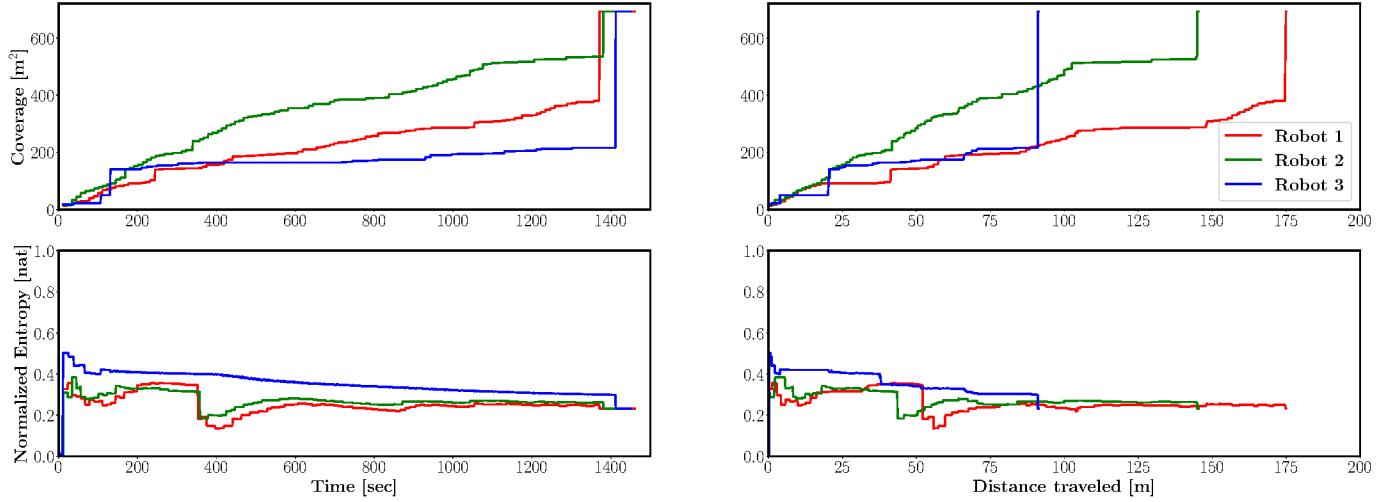


Fig. 19: Coverage (top row) and normalized map entropy (bottom row) versus time (left column) and distance traveled (right column) for the real-world multi-robot active mapping experiment. Each color corresponds to one robot in the team.

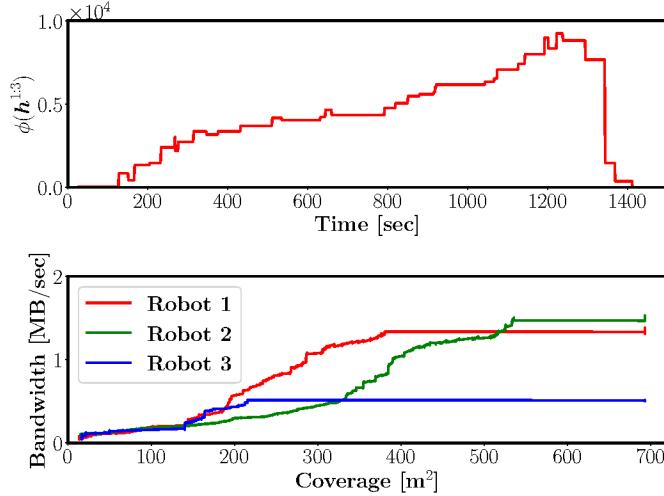


Fig. 20: Multi-robot exploration performance metrics for the real-world experiment. Top: Evolution of the map discrepancy $\phi(h^{1:3})$ across the robot network \mathcal{G} over time. Bottom: Bandwidth requirement of each robot for the distributed mapping with respect to coverage.

tions, for example formulating multi-robot control for $SE(3)$ robot dynamics [71]. An important future research direction involves analyzing the distance to consensus as well as the sub-optimality bounds for ROAM in the case of non-convex distance measures and non-concave objective functions. Other important future research directions include using Nesterov accelerated [72] and second-order [73] gradient methods in ROAM, extending ROAM to handle time-varying communication [12] and unknown robot localization, and handling cases where some robots act adversarially by intentionally sharing incorrect maps and plans.

APPENDIX A PROOF OF THEOREM 1

We organize the proof into three main steps. First, we show that the aggregate distance function $\phi(\cdot)$ is geodesically L -smooth. Second, we show that Alg. 1 converges to a consensus configuration. Last, optimality properties of Alg. 1 are derived.

Step 1. We begin the proof by showing that the aggregate distance function $\phi(\cdot)$ is geodesically L -smooth. Namely, for any pair of joint states $\mathbf{x}, \mathbf{y} \in \mathcal{M}^{|\mathcal{V}|}$ we prove:

$$\|g_\phi(\mathbf{x}) - \mathcal{T}_{\mathbf{y}}^{\mathbf{x}} g_\phi(\mathbf{y})\|_{\mathbf{x}} \leq L d(\mathbf{x}, \mathbf{y}),$$

with $g_\phi(\mathbf{x})$ as a shorthand notation for $\text{grad } \phi(\mathbf{x})$.

For two joint states $\mathbf{x}, \mathbf{y} \in \mathcal{M}^{|\mathcal{V}|}$ we have:

$$\text{grad}_{x^i} \phi(\mathbf{x}) - \mathcal{T}_{y^i}^{x^i} \text{grad}_{y^i} \phi(\mathbf{y}) = -2 \sum_{j \in \mathcal{V}} A_{ij} (v_x^{ij} - \mathcal{T}_{y^i}^{x^i} v_y^{ij}),$$

where the vectors v_x^{ij} and v_y^{ij} follow the notation in Assumption 2. Using row-stochasticity of A and the fact that $\mathcal{M}^{|\mathcal{V}|}$ is the product manifold of \mathcal{M} leads to:

$$\|g_\phi(\mathbf{x}) - \mathcal{T}_{\mathbf{y}}^{\mathbf{x}} g_\phi(\mathbf{y})\|_{\mathbf{x}}^2 \leq 4 \sum_{(i,j) \in \mathcal{V}^2} A_{ij} \|v_x^{ij} - \mathcal{T}_{y^i}^{x^i} v_y^{ij}\|_{x^i}^2, \quad (19)$$

Adding and subtracting tangent terms v_{xy}^i and v_{xy}^j as well as utilizing the Cauchy-Schwarz inequality in $T_{x^i} \mathcal{M}$ results in the following decomposition:

$$\begin{aligned} \|v_x^{ij} - \mathcal{T}_{y^i}^{x^i} v_y^{ij}\|_{x^i}^2 &\leq \|v_{xy}^i - \mathcal{T}_{x^j}^{x^i} v_{xy}^j\|_{x^i}^2 + \|v_{xy}^j\|_{x^i}^2 + \\ &\quad 2\|v_{xy}^i - \mathcal{T}_{x^j}^{x^i} v_{xy}^j\|_{x^i} \|v_{xy}^j\|_{x^i}, \end{aligned} \quad (20)$$

such that the vector $v_{xy}^i \in T_{x^i} \mathcal{M}$ is defined in (6). The vector v_{xy}^i contains sum of 4 vectors corresponding to a geodesic loop. Assumption 2, in addition to applying the triangle inequality, allows finding an upper bound for (20) that does not involve v_x^{ij} and v_y^{ij} terms:

$$\|v_x^{ij} - \mathcal{T}_{y^i}^{x^i} v_y^{ij}\|_{x^i}^2 \leq (\rho + 1)^2 (\|v_{xy}^i\|_{x^i} + \|v_{xy}^j\|_{x^i})^2.$$

Plugging into (19) and summing over $(i, j) \in \mathcal{V}^2$ yields:

$$\begin{aligned} \|g_\phi(\mathbf{x}) - \mathcal{T}_{\mathbf{y}}^{\mathbf{x}} g_\phi(\mathbf{y})\|_{\mathbf{x}}^2 &\leq \\ &8(\rho + 1)^2 \left(d^2(\mathbf{x}, \mathbf{y}) + \sum_{(i,j) \in \mathcal{V}^2} A_{ij} \|v_{xy}^i\|_{x^i} \|v_{xy}^j\|_{x^j} \right), \end{aligned}$$

The summation term in the above inequality is upper-bounded by the induced norm of A , i.e. its largest eigenvalue. Using

the fact that the largest eigenvalue of row-stochastic matrices is 1, we derive the following:

$$\frac{\|g_\phi(\mathbf{x}) - \mathcal{T}_\mathbf{y}^\mathbf{x} g_\phi(\mathbf{y})\|_\mathbf{x}}{d(\mathbf{x}, \mathbf{y})} \leq 4(1 + \rho) := L. \quad (21)$$

Therefore, the aggregate distance function $\phi(\mathbf{x})$ is geodesically L -smooth, with $L = 4(1 + \rho)$.

Step 2. This step proves convergence of Alg. 1 to a consensus configuration. We use the L -smoothness of $\phi(\cdot)$ to find a bound for the values of $\phi(\mathbf{x})$. Consider two points \mathbf{x} and \mathbf{y} in $\mathcal{M}^{|\mathcal{V}|}$, and the geodesic $\mathbf{s}(\cdot) : [0, 1] \rightarrow \mathcal{M}^{|\mathcal{V}|}$ connecting \mathbf{x} to \mathbf{y} , i.e. $\mathbf{s}(0) = \mathbf{x}$ and $\mathbf{s}(1) = \mathbf{y}$. Using the fundamental theorem of calculus for line integrals we have:

$$\phi(\mathbf{y}) - \phi(\mathbf{x}) - \langle g_\phi(\mathbf{x}), \text{Exp}_\mathbf{x}^{-1}(\mathbf{y}) \rangle_\mathbf{x} = \int_0^1 \langle \mathcal{T}_{\mathbf{s}(t)}^\mathbf{x} g_\phi(\mathbf{s}(t)) - g_\phi(\mathbf{x}), \text{Exp}_\mathbf{x}^{-1}(\mathbf{y}) \rangle_\mathbf{x} dt.$$

Applying the Cauchy-Schwarz inequality and using the L -smoothness of $\phi(\mathbf{x})$ results in:

$$\phi(\mathbf{y}) - \phi(\mathbf{x}) - \langle g_\phi(\mathbf{x}), \text{Exp}_\mathbf{x}^{-1}(\mathbf{y}) \rangle_\mathbf{x} \leq L d^2(\mathbf{x}, \mathbf{y}) \int_0^1 t dt = \frac{L}{2} d^2(\mathbf{x}, \mathbf{y}). \quad (22)$$

The above bound helps to analyze the dynamics of the joint state \mathbf{x} over the iterations of Alg. 1. Consider line 4 of Alg. 1. Using (22) leads to:

$$\phi(\tilde{\mathbf{x}}^{(k)}) \leq \phi(\mathbf{x}^{(k)}) + \langle g_\phi(\mathbf{x}^{(k)}), -\epsilon g_\phi(\mathbf{x}^{(k)}) \rangle_{\mathbf{x}^{(k)}} + \frac{L\epsilon^2}{2} \|g_\phi(\mathbf{x}^{(k)})\|_{\mathbf{x}^{(k)}}^2. \quad (23)$$

Similarly, for line 6 of Alg. 1 we have:

$$\phi(\mathbf{x}^{(k+1)}) \leq \phi(\tilde{\mathbf{x}}^{(k)}) + \langle g_\phi(\tilde{\mathbf{x}}^{(k)}), \alpha^{(k)} g_F(\tilde{\mathbf{x}}^{(k)}) \rangle_{\tilde{\mathbf{x}}^{(k)}} + \frac{L\alpha^{(k)}^2}{2} \|g_F(\tilde{\mathbf{x}}^{(k)})\|_{\tilde{\mathbf{x}}^{(k)}}^2,$$

where, analogous to $g_\phi(\mathbf{x}^{(k)})$, $g_F(\tilde{\mathbf{x}}^{(k)})$ is shorthand notation for $\text{grad } F(\mathbf{x})|_{\mathbf{x}=\tilde{\mathbf{x}}^{(k)}}$. Using the positive definiteness of the Riemannian metric $\langle v, u \rangle_\mathbf{x}$, we have $2\langle v, u \rangle_\mathbf{x} \leq \eta\|v\|_\mathbf{x}^2 + \|u\|_\mathbf{x}^2/\eta$ for any $v, u \in T_\mathbf{x} \mathcal{M}^{|\mathcal{V}|}$ and $\eta > 0$. Hence:

$$\phi(\mathbf{x}^{(k+1)}) \leq \phi(\tilde{\mathbf{x}}^{(k)}) + \frac{\eta}{2} \|g_\phi(\tilde{\mathbf{x}}^{(k)})\|_{\tilde{\mathbf{x}}^{(k)}}^2 + \frac{\alpha^{(k)}^2}{2} (L + \frac{1}{\eta}) \|g_F(\tilde{\mathbf{x}}^{(k)})\|_{\tilde{\mathbf{x}}^{(k)}}^2.$$

By adding and subtracting $\mathcal{T}_{\mathbf{x}^{(k)}}^\mathbf{x} g_\phi(\mathbf{x}^{(k)})$ from $g_\phi(\tilde{\mathbf{x}}^{(k)})$, and using the fact that $\|v + u\|_\mathbf{x}^2/2 \leq \|v\|_\mathbf{x}^2 + \|u\|_\mathbf{x}^2$, we have:

$$\phi(\mathbf{x}^{(k+1)}) \leq \phi(\tilde{\mathbf{x}}^{(k)}) + \frac{\alpha^{(k)}^2}{2} (L + \frac{1}{\eta}) \|g_F(\tilde{\mathbf{x}}^{(k)})\|_{\tilde{\mathbf{x}}^{(k)}}^2 + \eta \|g_\phi(\tilde{\mathbf{x}}^{(k)}) - \mathcal{T}_{\mathbf{x}^{(k)}}^\mathbf{x} g_\phi(\mathbf{x}^{(k)})\|_{\tilde{\mathbf{x}}^{(k)}}^2 + \eta \|g_\phi(\mathbf{x}^{(k)})\|_{\mathbf{x}^{(k)}}^2.$$

Using the bound for gradients of the local objective functions $f^i(x^i)$ alongside utilizing the L -smoothness of $\phi(\mathbf{x})$ and plugging (23) into the above inequality, yields:

$$\phi(\mathbf{x}^{(k+1)}) \leq \phi(\mathbf{x}^{(k)}) + \frac{\alpha^{(k)}^2}{2|\mathcal{V}|} (L + \frac{1}{\eta}) C^2 + [\epsilon(\frac{L\epsilon}{2} - 1) + \eta(1 + L^2\epsilon^2)] \|g_\phi(\mathbf{x}^{(k)})\|_{\mathbf{x}^{(k)}}^2.$$

Choosing $\eta = \frac{\epsilon(2-L\epsilon)}{4(1+L^2\epsilon^2)}$ we have:

$$\frac{\epsilon}{2} (1 - \frac{L\epsilon}{2}) \|g_\phi(\mathbf{x}^{(k)})\|_{\mathbf{x}^{(k)}}^2 \leq \phi(\mathbf{x}^{(k)}) - \phi(\mathbf{x}^{(k+1)}) + \alpha^{(k)}^2 (\frac{1 + \epsilon^2 L^2}{\epsilon(1 - \frac{L\epsilon}{2})} + \frac{L}{2}) \frac{C^2}{|\mathcal{V}|}. \quad (24)$$

Summing over k_{\max} first iterations of Alg. 1 yields:

$$\frac{\epsilon}{2} (1 - \frac{L\epsilon}{2}) \sum_{k=0}^{k_{\max}} \|g_\phi(\mathbf{x}^{(k)})\|_{\mathbf{x}^{(k)}}^2 \leq \phi(\mathbf{x}^{(0)}) - \phi(\mathbf{x}^{(k_{\max}+1)}) + \frac{C^2}{|\mathcal{V}|} (\frac{1 + \epsilon^2 L^2}{\epsilon(1 - \frac{L\epsilon}{2})} + \frac{L}{2}) \sum_{k=0}^{k_{\max}} \alpha^{(k)}^2.$$

Because $\phi(\mathbf{x}^{(k_{\max}+1)})$ is always non-negative, we have:

$$\frac{\epsilon}{2} (1 - \frac{L\epsilon}{2}) \sum_{k=0}^{k_{\max}} \|g_\phi(\mathbf{x}^{(k)})\|_{\mathbf{x}^{(k)}}^2 \leq \phi(\mathbf{x}^{(0)}) + \frac{C^2}{|\mathcal{V}|} (\frac{1 + \epsilon^2 L^2}{\epsilon(1 - \frac{L\epsilon}{2})} + \frac{L}{2}) \sum_{k=0}^{k_{\max}} \alpha^{(k)}^2.$$

As a consequence of the compactness of \mathcal{M} , $\phi(\mathbf{x}^{(0)})$ will be bounded for any choice of $\mathbf{x}^{(0)}$. Moreover, setting $k_{\max} \rightarrow \infty$, due to the convergence property for the sum of the squared step sizes $\alpha^{(k)}^2$, we have:

$$\frac{\epsilon}{2} \left(1 - \frac{L\epsilon}{2}\right) \sum_{k=0}^{\infty} \|g_\phi(\mathbf{x}^{(k)})\|_{\mathbf{x}^{(k)}}^2 \leq \phi(\mathbf{x}^{(0)}) + \frac{C^2}{|\mathcal{V}|} \left(\frac{1 + \epsilon^2 L^2}{\epsilon(1 - \frac{L\epsilon}{2})} + \frac{L}{2}\right) \sum_{k=0}^{\infty} \alpha^{(k)}^2 < \infty. \quad (25)$$

Therefore, for $\epsilon \in (0, 2/L)$, $g_\phi(\mathbf{x}^{(k)})$ shrinks to zero as k goes to infinity. Since $\alpha^{(k)}$ is a decaying sequence and $\|g_F(\tilde{\mathbf{x}}^{(k)})\|_{\tilde{\mathbf{x}}^{(k)}} \leq \frac{C}{\sqrt{|\mathcal{V}|}}$ for all $k \geq 0$, Alg. 1 converges to a first-order critical point of $\phi(\mathbf{x})$.

Let $\mathbf{x}^{(\infty)}$ be the joint state that Alg. 1 converges to. Also, let \mathbf{x}_c be a consensus state, which can be constructed by setting all $x^i, i \in \mathcal{V}$, to an arbitrary state $x_c \in \mathcal{M}$. Since the squared distance $d^2(\cdot)$ is geodesically convex, and the adjacency matrix A is symmetric and row stochastic, it is possible to show that $\phi(\cdot)$ is also geodesically convex. Thus, we have:

$$\phi(\mathbf{x}_c) \geq \phi(\mathbf{x}^{(\infty)}) + \langle g_\phi(\mathbf{x}^{(\infty)}), \text{Exp}_{\mathbf{x}^{(\infty)}}^{-1}(\mathbf{x}_c) \rangle_{\mathbf{x}^{(\infty)}}.$$

Since $\phi(\mathbf{x}_c)$ is zero, $\phi(\mathbf{x}^{(\infty)})$ is non-negative, and (25) indicates that $\mathbf{x}^{(\infty)}$ is a first-order critical point of $\phi(\cdot)$, we arrive at $\phi(\mathbf{x}^{(\infty)}) = 0$ for Alg. 1. Note that (25) and $\phi(\mathbf{x}^{(\infty)}) = 0$ hold even for a disconnected graph \mathcal{G} . However, for the global asymptotic consensus, \mathcal{G} should be a connected graph; otherwise, consensus occurs separately for each connected component of \mathcal{G} .

Step 3. In the last step, we show the optimality properties of Alg. 1. We utilize the Riemannian manifold version of the law of cosines, which can be expressed for a geodesic triangle with side lengths a, b , and c as follows (Lemma 5, [74]):

$$a^2 \leq \frac{c\sqrt{|\kappa_{\min}|}}{\tanh(c\sqrt{|\kappa_{\min}|})} b^2 + c^2 - 2bc \cos(\angle bc), \quad (26)$$

where κ_{\min} is a lower bound on the sectional curvature of the manifold. Now, consider a geodesic triangle specified by $x^{i(k+1)}, \tilde{x}^{i(k)}$, and x^* , where $x^* \in \mathcal{M}$ is an element of the centralized optimal solution of (3), denoted by $\mathbf{x}^* \in \mathcal{M}^{|\mathcal{V}|}$:

$$\begin{aligned} d^2(x^{i(k+1)}, x^*) &\leq d^2(\tilde{x}^{i(k)}, x^*) + \zeta \alpha^{(k)} \|\mathbf{g}_{f^i}(\tilde{x}^{i(k)})\|_{\tilde{x}^{i(k)}}^2 \\ &\quad - 2\alpha^{(k)} \langle \mathbf{g}_{f^i}(\tilde{x}^{i(k)}), \text{Exp}_{\tilde{x}^{i(k)}}^{-1}(x^*) \rangle_{\tilde{x}^{i(k)}}, \end{aligned}$$

where $\zeta = \frac{d_{\max} \sqrt{|\kappa_{\min}|}}{\tanh(d_{\max} \sqrt{|\kappa_{\min}|})}$ with $d_{\max} \geq \max_{x,y \in \mathcal{M}} d(x,y)$. Since \mathcal{M} is compact, d_{\max} is well-defined. Also, the law of cosines still holds using d_{\max} instead of side length $d^2(\tilde{x}^{i(k)}, x^*)$ due to the strict monotonicity of the function $\frac{y}{\tanh(y)}$ for $y \in \mathbb{R}_{\geq 0}$. As a result of the local objective function concavity and gradient boundedness, we have:

$$\begin{aligned} 2\alpha^{(k)}(f^i(x^*) - f^i(\tilde{x}^{i(k)})) &\leq d^2(\tilde{x}^{i(k)}, x^*) - d^2(x^{i(k+1)}, x^*) \\ &\quad + \zeta C^2 \alpha^{(k)} \alpha^{(k)}. \end{aligned}$$

Summing over all agents in \mathcal{V} yields:

$$\begin{aligned} 2\alpha^{(k)}|\mathcal{V}|(F(\mathbf{x}^*) - F(\tilde{\mathbf{x}}^{(k)})) &\leq \\ d^2(\tilde{\mathbf{x}}^{(k)}, \mathbf{x}^*) - d^2(\mathbf{x}^{(k+1)}, \mathbf{x}^*) + \zeta |\mathcal{V}| C^2 \alpha^{(k)} \alpha^{(k)}. &\quad (27) \end{aligned}$$

Now, we repeat the same steps for the geodesic triangle specified by $\tilde{\mathbf{x}}^{(k)}, \mathbf{x}^{(k)}$, and \mathbf{x}^* :

$$\begin{aligned} d^2(\tilde{\mathbf{x}}^{(k)}, \mathbf{x}^*) &\leq d^2(\mathbf{x}^{(k)}, \mathbf{x}^*) + \zeta \epsilon^2 \|\mathbf{g}_\phi(\mathbf{x}^{(k)})\|_{\mathbf{x}^{(k)}}^2 \\ &\quad + 2\epsilon \langle \mathbf{g}_\phi(\mathbf{x}^{(k)}), \text{Exp}_{\mathbf{x}^{(k)}}^{-1}(\mathbf{x}^*) \rangle_{\mathbf{x}^{(k)}}. \end{aligned}$$

Using the convexity of $\phi(\cdot)$, the fact that $\phi(\mathbf{x}^*)$ is zero by definition, and $\phi(\mathbf{x}^{(k)})$ is always non-negative, we have:

$$d^2(\tilde{\mathbf{x}}^{(k)}, \mathbf{x}^*) \leq d^2(\mathbf{x}^{(k)}, \mathbf{x}^*) + \zeta \epsilon^2 \|\mathbf{g}_\phi(\mathbf{x}^{(k)})\|_{\mathbf{x}^{(k)}}^2.$$

Plugging (24) into the above inequality yields:

$$\begin{aligned} d^2(\tilde{\mathbf{x}}^{(k)}, \mathbf{x}^*) &\leq d^2(\mathbf{x}^{(k)}, \mathbf{x}^*) + \frac{2\zeta\epsilon}{1 - \frac{L\epsilon}{2}} [\phi(\mathbf{x}^{(k)}) - \phi(\mathbf{x}^{(k+1)})] \\ &\quad + \frac{\alpha^{(k)} \alpha^{(k)} C^2}{|\mathcal{V}|} \left[\frac{1 + \epsilon^2 L^2}{\epsilon(1 - \frac{L\epsilon}{2})} + \frac{L}{2} \right]. \end{aligned}$$

Plugging once more into (27) leads to:

$$\begin{aligned} 2\alpha^{(k)}|\mathcal{V}|(F(\mathbf{x}^*) - F(\tilde{\mathbf{x}}^{(k)})) &\leq d^2(\mathbf{x}^{(k)}, \mathbf{x}^*) - d^2(\mathbf{x}^{(k+1)}, \mathbf{x}^*) \\ &\quad + \zeta |\mathcal{V}| C^2 \alpha^{(k)} \alpha^{(k)} + \frac{2\zeta\epsilon}{1 - \frac{L\epsilon}{2}} [\phi(\mathbf{x}^{(k)}) - \phi(\mathbf{x}^{(k+1)})] \\ &\quad + \frac{\alpha^{(k)} \alpha^{(k)} C^2}{|\mathcal{V}|} \left[\frac{1 + \epsilon^2 L^2}{\epsilon(1 - \frac{L\epsilon}{2})} + \frac{L}{2} \right]. \end{aligned}$$

Summing over k_{\max} first iterations of Alg. 1 yields:

$$\begin{aligned} 2|\mathcal{V}| \sum_{k=0}^{k_{\max}} \alpha^{(k)}(F(\mathbf{x}^*) - F(\tilde{\mathbf{x}}^{(k)})) &\leq d^2(\mathbf{x}^{(0)}, \mathbf{x}^*) + \frac{4\zeta\phi(\mathbf{x}^{(0)})}{2/\epsilon - L} \\ &\quad + \zeta C^2 \left[\frac{2\epsilon}{|\mathcal{V}|(1 - \frac{L\epsilon}{2})} \left[\frac{1 + \epsilon^2 L^2}{\epsilon(1 - \frac{L\epsilon}{2})} + \frac{L}{2} \right] + |\mathcal{V}| \right] \sum_{k=0}^{k_{\max}} \alpha^{(k)} \alpha^{(k)}. \end{aligned}$$

It is straightforward to show:

$$\begin{aligned} \min_{0 \leq k' \leq k_{\max}} \{F(\mathbf{x}^*) - F(\tilde{\mathbf{x}}^{(k')})\} \sum_{k=0}^{k_{\max}} \alpha^{(k)} &\leq \\ \sum_{k=0}^{k_{\max}} \alpha^{(k)} (F(\mathbf{x}^*) - F(\tilde{\mathbf{x}}^{(k)})). \end{aligned}$$

Therefore, we have:

$$\begin{aligned} F(\mathbf{x}^*) &\leq \max_{0 \leq k' \leq k_{\max}} \{F(\tilde{\mathbf{x}}^{(k')})\} \\ &\quad + \frac{1}{2|\mathcal{V}| \sum_{k=0}^{k_{\max}} \alpha^{(k)}} \left[d^2(\mathbf{x}^{(0)}, \mathbf{x}^*) + \frac{4\zeta\phi(\mathbf{x}^{(0)})}{2/\epsilon - L} \right. \\ &\quad \left. + \zeta C^2 \left(\frac{2\epsilon}{|\mathcal{V}|(1 - \frac{L\epsilon}{2})} \left(\frac{1 + \epsilon^2 L^2}{\epsilon(1 - \frac{L\epsilon}{2})} + \frac{L}{2} \right) + |\mathcal{V}| \right) \sum_{k=0}^{k_{\max}} \alpha^{(k)} \alpha^{(k)} \right]. \end{aligned}$$

Since $d^2(\mathbf{x}^{(0)}, \mathbf{x}^*)$, $\phi(\mathbf{x}^{(0)})$, and $\sum_{k=0}^{\infty} \alpha^{(k)} \alpha^{(k)}$ are bounded and $\sum_{k=0}^{\infty} \alpha^{(k)} = \infty$, the term inside the brackets in the right hand side of the above inequality vanishes as $k_{\max} \rightarrow \infty$. Therefore, $\max_{0 \leq k' \leq k_{\max}} \{F(\tilde{\mathbf{x}}^{(k')})\}$ will be asymptotically lower-bounded by $F(\mathbf{x}^*)$. Moreover, since $\|\mathbf{g}_\phi(\mathbf{x}^{(k)})\|_{\mathbf{x}^{(k)}}$ and $\alpha^{(k)}$ both shrink to zero as $k \rightarrow \infty$, we derive the same asymptotic lower bound for $F(\mathbf{x}^{(k)})$:

$$F(\mathbf{x}^*) \leq \lim_{k_{\max} \rightarrow \infty} \max_{0 \leq k \leq k_{\max}} \{F(\mathbf{x}^{(k)})\}. \quad (28)$$

In summary, (25) with step size $\epsilon \in (0, 2/L)$ leads to convergence to a consensus configuration $\phi(\mathbf{x}^{(\infty)}) = 0$, where $L = 4(1 + \rho)$. Furthermore, (28) expresses the optimality of Alg. 1 with respect to a centralized solution. \square

APPENDIX B PROOF OF LEMMA 1

The conditional independence of observations given the map \mathbf{m} allows decomposing the observation model $q^i(\mathbf{z}_{1:t}^i | \mathbf{m})$ into a product of single observation models $q^i(\mathbf{z}_\tau^i | \mathbf{m})$, $\tau \in \{1, \dots, t\}$. By applying Bayes rule to the decomposed observation model, the objective function in (9) can be written as:

$$\sum_{i \in \mathcal{V}} \sum_{\tau=1}^t \left(\log q^i(\mathbf{z}_\tau^i) + \mathbb{E}_{\mathbf{m} \sim p} \left[\log \frac{q^i(\mathbf{m} | \mathbf{z}_\tau^i)}{p(\mathbf{m})} \right] \right).$$

Using the map independence assumption, the log term inside the expectation can be expressed as the sum of log terms with respect to single cells. The additivity of expected value yields:

$$\sum_{i \in \mathcal{V}} \sum_{\tau=1}^t \left(\log q^i(\mathbf{z}_\tau^i) + \sum_{n=1}^N \mathbb{E}_{\mathbf{m} \sim p} \left[\log \frac{q^i(m_n | \mathbf{z}_\tau^i)}{p_n(m_n)} \right] \right).$$

Since every term inside the expectation only depends on a single cell $m_n \sim p_n$, $n \in \{1, \dots, N\}$, the expectation can thus be simplified to only incorporate one cell instead of the joint distribution $\mathbf{m} \sim p$. This leads to the expression in (11). \square

REFERENCES

- [1] J. A. Placed, J. Strader, H. Carrillo, N. Atanasov, V. Indelman, L. Carbone, and J. A. Castellanos, "A survey on active simultaneous localization and mapping: State of the art and new frontiers," *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 1686–1705, 2023.

[2] J. P. Queraltà, J. Taipalmaa, B. C. Pullinen, V. K. Sarker, T. N. Gia, H. Tenhunen, M. Gabouj, J. Raitoharju, and T. Westerlund, “Collaborative multi-robot search and rescue: Planning, coordination, perception, and active vision,” *IEEE Access*, vol. 8, pp. 191 617–191 643, 2020.

[3] T. Rouček, M. Pecka, P. Čížek, T. Petříček, J. Bayer, V. Šalanský, D. Heřt, M. Petrlík, T. Báča, V. Spurný *et al.*, “DARPA subterranean challenge: Multi-robotic exploration of underground environments,” in *Modelling and Simulation for Autonomous Systems: 6th International Conference, MESAS 2019, Palermo, Italy, October 29–31, 2019, Revised Selected Papers 6*. Springer, 2020, pp. 274–290.

[4] B. Grocholsky, J. Keller, V. Kumar, and G. Pappas, “Cooperative air and ground surveillance,” *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 16–25, 2006.

[5] X. Huang, M. Sun, H. Zhou, and S. Liu, “A multi-robot coverage path planning algorithm for the environment with multiple land cover types,” *IEEE Access*, vol. 8, pp. 198 101–198 117, 2020.

[6] D. Morilla-Cabello, L. Bartolomei, L. Teixeira, E. Montijano, and M. Chli, “Sweep-your-map: Efficient coverage planning for aerial teams in large-scale environments,” *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 810–10 817, 2022.

[7] A. Dutta, A. Ghosh, and O. P. Kreidl, “Multi-robot informative path planning with continuous connectivity constraints,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019, pp. 3245–3251.

[8] X. Liu, A. Prabhu, F. Cladera, I. D. Miller, L. Zhou, C. J. Taylor, and V. Kumar, “Active metric-semantic mapping by multiple aerial robots,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 3282–3288.

[9] A. Prorok, M. Malencia, L. Carloni, G. S. Sukhatme, B. M. Sadler, and V. Kumar, “Beyond robustness: A taxonomy of approaches towards resilient multi-robot systems,” *arXiv preprint arXiv:2109.12343*, 2021.

[10] A. Asgharivaskasi and N. Atanasov, “Semantic octree mapping and Shannon mutual information computation for robot exploration,” *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 1910–1928, 2023.

[11] B. Gharesifard and J. Cortés, “Distributed continuous-time convex optimization on weight-balanced digraphs,” *IEEE Transactions on Automatic Control*, vol. 59, no. 3, pp. 781–786, 2014.

[12] A. Nedić and A. Olshevsky, “Distributed optimization over time-varying directed graphs,” *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 601–615, 2014.

[13] S. S. Ram, A. Nedić, and V. V. Veeravalli, “A new class of distributed optimization algorithms: application to regression of distributed data,” *Optimization Methods and Software*, vol. 27, no. 1, pp. 71–88, 2012.

[14] T. Halsted, O. Shorinwa, J. Yu, and M. Schwager, “A survey of distributed optimization methods for multi-robot systems,” *arXiv preprint arXiv:2103.12840*, 2021.

[15] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifold*. Princeton University Press, 2008.

[16] S. Chen, A. Garcia, M. Hong, and S. Shahrampour, “Decentralized Riemannian gradient descent on the Stiefel manifold,” in *International Conference on Machine Learning*, 2021, pp. 1594–1605.

[17] L. Wang and X. Liu, “Decentralized optimization over the Stiefel manifold by an approximate augmented Lagrangian function,” *IEEE Transactions on Signal Processing*, vol. 70, pp. 3029–3041, 2022.

[18] H. Zhang, S. J. Reddi, and S. Sra, “Riemannian SVRG: Fast stochastic optimization on Riemannian manifolds,” *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[19] J. Li and S. Ma, “Federated learning on Riemannian manifolds,” *arXiv preprint arXiv:2206.05668*, 2022.

[20] Y. Tian, A. Koppel, A. S. Bedi, and J. P. How, “Asynchronous and parallel distributed pose graph optimization,” *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5819–5826, 2020.

[21] S. M. Shah, “Distributed optimization on Riemannian manifolds for multi-agent networks,” *arXiv preprint arXiv:1711.11196*, 2017.

[22] N. Atanasov, R. Tron, V. M. Preciado, and G. J. Pappas, “Joint estimation and localization in sensor networks,” in *IEEE Conference on Decision and Control*, 2014, pp. 6875–6882.

[23] E. Zobeidi, A. Koppel, and N. Atanasov, “Dense incremental metric-semantic mapping for multiagent systems via sparse Gaussian process regression,” *IEEE Transactions on Robotics*, vol. 38, no. 5, pp. 3133–3153, 2022.

[24] Y. Tian, Y. Chang, F. H. Arias, C. Nieto-Granda, J. P. How, and L. Carloni, “Kimera-multi: Robust, distributed, dense metric-semantic slam for multi-robot systems,” *IEEE Transactions on Robotics*, vol. 38, no. 4, 2022.

[25] P. Paritosh, N. Atanasov, and S. Martinez, “Distributed Bayesian estimation of continuous variables over time-varying directed networks,” *IEEE Control Systems Letters*, vol. 6, pp. 2545–2550, 2022.

[26] M. Corah, C. O’Meadhra, K. Goel, and N. Michael, “Communication-efficient planning and mapping for multi-robot exploration in large environments,” *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1715–1721, 2019.

[27] H. Dong, J. Yu, Y. Xu, Z. Xu, Z. Shen, J. Tang, Y. Shen, and Y. Wang, “MR-GMMMapping: Communication efficient multi-robot mapping system via Gaussian mixture model,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3294–3301, 2022.

[28] Y. Wu, Q. Gu, J. Yu, G. Ge, J. Wang, Q. Liao, C. Zhang, and Y. Wang, “MR-GMMExplore: Multi-robot exploration system in unknown environments based on Gaussian mixture model,” in *IEEE International Conference on Robotics and Biomimetics*, 2022, pp. 1198–1203.

[29] J. Yu, J. Tong, Y. Xu, Z. Xu, H. Dong, T. Yang, and Y. Wang, “SMMR-Explore: Submap-based multi-robot exploration system with multi-robot multi-target potential field exploration method,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 8779–8785.

[30] T. Duhautbou, J. Moras, and J. Marzat, “Distributed 3D TSDF manifold mapping for multi-robot systems,” in *European Conference on Mobile Robots (ECMR)*, 2019, pp. 1–8.

[31] Y. Deng, Y. Tang, Y. Yang, D. Wang, and Y. Yue, “MACIM: Multi-agent collaborative implicit mapping,” *IEEE Robotics and Automation Letters*, vol. 9, no. 5, pp. 4369–4376, 2024.

[32] J. Jessup, S. N. Givigi, and A. Beaulieu, “Merging of octree based 3D occupancy grid maps,” in *IEEE International Systems Conference*, 2014, pp. 371–377.

[33] A. Madridano, A. Al-Kaff, D. Martín, and A. De La Escalera, “Trajectory planning for multi-robot systems: Methods and applications,” *Expert Systems with Applications*, vol. 173, p. 114660, 2021.

[34] W. Gosrich, S. Mayya, R. Li, J. Paulos, M. Yim, A. Ribeiro, and V. Kumar, “Coverage control in multi-robot systems via graph neural networks,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2022, pp. 8787–8793.

[35] L. Zhou, V. D. Sharma, Q. Li, A. Prorok, A. Ribeiro, P. Tokekar, and V. Kumar, “Graph neural networks for decentralized multi-robot target tracking,” in *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2022, pp. 195–202.

[36] W. Wu, S. Bhattacharya, and A. Prorok, “Multi-robot path deconfliction through prioritization by path prospects,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 9809–9815.

[37] L. Freda, T. Novo, D. Portugal, and R. P. Rocha, “3d multi-robot exploration with a two-level coordination strategy and prioritization,” *arXiv preprint arXiv:2307.02417*, 2023.

[38] N. Atanasov, J. Le Ny, K. Daniilidis, and G. J. Pappas, “Decentralized active information acquisition: Theory and application to multi-robot SLAM,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 4775–4782.

[39] B. Schlotfeldt, D. Thakur, N. Atanasov, V. Kumar, and G. J. Pappas, “Anytime planning for decentralized multirobot active information gathering,” *IEEE Robotics and Automation Letters (RA-L)*, vol. 3, no. 2, pp. 1025–1032, 2018.

[40] Y. Kantaros, B. Schlotfeldt, N. Atanasov, and G. J. Pappas, “Asymptotically optimal planning for non-myopic multi-robot information gathering,” in *Robotics: Science and Systems*, 2019, pp. 22–26.

[41] M. Tzes, Y. Kantaros, and G. J. Pappas, “Distributed sampling-based planning for non-myopic active information gathering,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 5872–5877.

[42] X. Cai, B. Schlotfeldt, K. Khosoussi, N. Atanasov, G. J. Pappas, and J. P. How, “Energy-aware, collision-free information gathering for heterogeneous robot teams,” *IEEE Transactions on Robotics*, vol. 39, no. 4, pp. 2585–2602, 2023.

[43] L. Zhou and V. Kumar, “Robust multi-robot active target tracking against sensing and communication attacks,” *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 1768–1780, 2023.

[44] M. Tzes, N. Bousias, E. Chatzipantazis, and G. J. Pappas, “Graph neural networks for multi-robot active information acquisition,” in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 3497–3503.

[45] J. Liu, L. Zhou, R. Ramachandran, G. S. Sukhatme, and V. Kumar, “Decentralized risk-aware tracking of multiple targets,” in *Distributed Autonomous Robotic Systems (DARS)*, 2024, pp. 408–423.

[46] R. Zahroo, J. Liu, L. Zhou, and V. Kumar, “Multi-robot localization and target tracking with connectivity maintenance and collision avoidance,” in *2023 American Control Conference (ACC)*, 2023, pp. 1331–1338.

[47] M. Corah and N. Michael, "Distributed matroid-constrained submodular maximization for multi-robot exploration: Theory and practice," *Autonomous Robots*, vol. 43, pp. 485–501, 2019.

[48] G. Best, O. M. Cliff, T. Patten, R. R. Mettu, and R. Fitch, "Dec-MCTS: Decentralized planning for multi-robot active perception," *The International Journal of Robotics Research*, vol. 38, no. 2-3, pp. 316–337, 2019.

[49] S. Koga, A. Asgharivaskasi, and N. Atanasov, "Active exploration and mapping via iterative covariance regulation over continuous SE(3) trajectories," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 2735–2741.

[50] S. Koga, A. Asgharivaskasi, and N. Atanasov, "Active SLAM over continuous trajectory and control: A covariance-feedback approach," in *American Control Conference (ACC)*, 2022, pp. 5062–5068.

[51] P. Yang, S. Koga, A. Asgharivaskasi, and N. Atanasov, "Policy learning for active target tracking over continuous SE(3) trajectories," in *Learning for Dynamics and Control Conference*, 2023, pp. 64–75.

[52] P. Yang, Y. Liu, S. Koga, A. Asgharivaskasi, and N. Atanasov, "Learning continuous control policies for information-theoretic active perception," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 2098–2104.

[53] J. Hu, H. Niu, J. Carrasco, B. Lennox, and F. Arvin, "Voronoi-based multi-robot autonomous exploration in unknown environments via deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 12, pp. 14413–14423, 2020.

[54] N. Boumal, *An introduction to optimization on smooth manifolds*. Cambridge University Press, 2023.

[55] R. Sepulchre, "Consensus on nonlinear spaces," *Annual Reviews in Control*, vol. 35, no. 1, pp. 56–64, 2011.

[56] A. Nedić, *Convergence Rate of Distributed Averaging Dynamics and Optimization in Networks*. Now Publishers Inc., 2015.

[57] A. Milioto and C. Stachniss, "Bonnet: An open-source training and deployment framework for semantic segmentation in robotics using CNNs," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019, pp. 7094–7100.

[58] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, "RangeNet++: Fast and accurate LiDAR semantic segmentation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 4213–4220.

[59] T. D. Barfoot, *State Estimation for Robotics*. Cambridge University Press, 2017.

[60] A. Asgharivaskasi, S. Koga, and N. Atanasov, "Active mapping via gradient ascent optimization of Shannon mutual information over continuous SE(3) trajectories," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 12994–13001.

[61] T. B. Cohn, M. Petersen, M. Simchowitz, and R. Tedrake, "Non-Euclidean motion planning with graphs of geodesically-convex sets," in *Robotics: Science and Systems (RSS)*, 2023.

[62] B. J. Julian, S. Karaman, and D. Rus, "On mutual information-based control of range sensing robots for mapping applications," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013, pp. 5156–5163.

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

[64] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous robots*, vol. 34, pp. 189–206, 2013.

[65] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[66] S. Song, S. P. Lichtenberg, and J. Xiao, "Sun RGB-D: A RGB-D scene understanding benchmark suite," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 567–576.

[67] NVIDIA, "Deep Learning Nodes for ROS/ROS2," https://github.com/dusty-nv/ros_deep_learning, accessed: 2023-07-10.

[68] K. Chen, B. T. Lopez, A.-a. Agha-mohammadi, and A. Mehta, "Direct LiDAR odometry: Fast localization with dense point clouds," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2000–2007, 2022.

[69] A. Censi, "An ICP variant using a point-to-line metric," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2008, pp. 19–25.

[70] J. Wang and E. Olson, "Apriltag 2: Efficient and robust fiducial detection," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 4193–4198.

[71] T. Duong, A. Altawaitan, J. Stanley, and N. Atanasov, "Port-Hamiltonian neural ODE networks on Lie groups for robot dynamics learning and control," *arXiv preprint arXiv:2401.09520*, 2024.

[72] M. Assran and M. Rabbat, "On the convergence of Nesterov's accelerated gradient method in stochastic settings," in *International Conference on Machine Learning*, 2020, pp. 410–420.

[73] T. Pierrot, N. Perrin-Gilbert, and O. Sigaud, "First-order and second-order variants of the gradient descent in a unified framework," in *International Conference on Artificial Neural Networks*, 2021, pp. 197–208.

[74] H. Zhang and S. Sra, "First-order methods for geodesically convex optimization," in *Conference on Learning Theory*, vol. 49, 2016, pp. 1617–1638.



Arash Asgharivaskasi (S'22) is a Ph.D. student of Electrical and Computer Engineering at the University of California San Diego, La Jolla, CA. He obtained a B.S. degree in Electrical Engineering from Sharif University of Technology, Tehran, Iran, and an M.S. degree in Electrical and Computer Engineering from the University of California San Diego, La Jolla, CA, USA. His research focuses on active information acquisition using mobile robots with applications to mapping, security, and environmental monitoring.



Fritz Girke is pursuing an M.S. degree in Electrical Engineering and Information Technology at the Technical University of Munich (TUM), Munich, Germany. They commenced their B.S. degree in Electrical Engineering and Information Technology at TUM in 2017 and completed it in 2021. Their research interests lie in the realm of robotics, specifically focusing on point cloud registration, semantic segmentation, odometry, and simultaneous localization and mapping (SLAM) in mobile robots.



Nikolay Atanasov (S'07-M'16-SM'23) is an Associate Professor of Electrical and Computer Engineering at the University of California San Diego, La Jolla, CA, USA. He obtained a B.S. degree in Electrical Engineering from Trinity College, Hartford, CT, USA in 2008, and M.S. and Ph.D. degrees in Electrical and Systems Engineering from University of Pennsylvania, Philadelphia, PA, USA in 2012 and 2015, respectively. Dr. Atanasov's research focuses on robotics, control theory, and machine learning with emphasis on active perception problems for autonomous mobile robots. He works on probabilistic models that unify geometric and semantic information in simultaneous localization and mapping (SLAM) and on optimal control and reinforcement learning algorithms for minimizing probabilistic model uncertainty. Dr. Atanasov's work has been recognized by the Joseph and Rosaline Wolf award for the best Ph.D. dissertation in Electrical and Systems Engineering at the University of Pennsylvania in 2015, the Best Conference Paper Award at the IEEE International Conference on Robotics and Automation (ICRA) in 2017, the NSF CAREER Award in 2021, and the IEEE RAS Early Academic Career Award in Robotics and Automation in 2023.