

Active Collaborative Localization in Heterogeneous Robot Teams

Igor Spasojevic*, Xu Liu*, Alejandro Ribeiro, George J. Pappas, Vijay Kumar

Abstract—Accurate and robust state estimation is critical for autonomous navigation of robot teams. This task is especially challenging for large groups of size, weight, and power (SWAP) constrained aerial robots operating in perceptually-degraded GPS-denied environments. We can, however, actively increase the amount of perceptual information available to such robots by augmenting them with a small number of more expensive, but less resource-constrained, agents. Specifically, the latter can serve as sources of perceptual information themselves. In this paper, we study the problem of optimally positioning (and potentially navigating) a small number of more capable agents to enhance the perceptual environment for their lightweight, inexpensive, teammates that only need to rely on cameras and IMUs. We propose a numerically robust, computationally efficient approach to solve this problem via nonlinear optimization. Our method outperforms the standard approach based on the greedy algorithm, while matching the accuracy of a heuristic evolutionary scheme for global optimization at a fraction of its running time. Ultimately, we validate our solution in both photorealistic simulations and real-world experiments. In these experiments, we use lidar-based autonomous ground vehicles as the more capable agents, and vision-based aerial robots as their SWAP-constrained teammates. Our method is able to reduce drift in visual-inertial odometry by as much as 90%, and it outperforms random positioning of lidar-equipped agents by a significant margin. Furthermore, our method can be generalized to different types of robot teams with heterogeneous perception capabilities. It has a wide range of applications, such as surveying and mapping challenging dynamic environments, and enabling resilience to large-scale perturbations that can be caused by earthquakes or storms.

I. INTRODUCTION

The low size, weight, and power requirements of inertial measurement units (IMUs) and cameras have made them a standard combination of sensors for SWAP-constrained flying robots such as agile micro aerial vehicles. Nevertheless, harnessing these sensors for accurate state estimation requires overcoming several challenges. In particular, visual-inertial odometry (VIO) quickly accumulates significant drift in unstructured, dynamic, or featureless environments. Such drift can dramatically degrade the efficacy and safety of an autonomous system.

Effective use of cameras rests upon the presence of a sufficient quantity of visual information in the form of texture

*Equal contribution. This work was supported by The Institute for Learning-Enabled Optimization at Scale (TILOS) funded by the National Science Foundation (NSF) under NSF Grant CCR-2112665, IoT4Ag ERC funded through NSF Grant EEC-1941529, the ARL DCIST CRA W911NF-17-2-0181, and ONR Grant N00014-20-1-2822. All authors are with the GRASP Laboratory, University of Pennsylvania, United States {igorspas, liuxu, aribeiro, pappasg, kumar}@seas.upenn.edu.



Figure 1: Robot platforms used in our experiments. Both platforms are capable of long-range GPS-denied autonomous navigation. In the experiments of this paper, the UAV relies only on cameras and the IMU for state estimation, while the UGVs rely on lidars. We refer readers to [1] and [2] for details on our UAV and UGV platforms.

or landmarks in the environment. Information-rich scenarios display a large set of visual cues, typically in the form of landmarks spread throughout the surroundings of the agent. In the absence of such visual features, the quality of state estimates rapidly degrades. To overcome this challenge, we consider augmenting low-SWAP robots with a small number of agents without such stringent payload constraints to robustly navigate unstructured environments.

The predominant way of solving such problems for a single agent involves planning perception-aware trajectories. Intuitively, this means choosing paths that only traverse regions of the environment with sufficiently rich perceptual content. Naturally, this limits the operating space of the robot. However, in multi-agent settings, we can leverage the fact that we can design heterogeneous teams comprising of agents with different perceptual capabilities. In particular, a select subset of just a few perceptually-advantaged robots can act as landmarks for their vision-driven teammates. Our approach involves optimizing the positions of the former subset of agents using a second-order smooth optimization scheme.

We summarize our contributions as follows:

- 1) We propose a framework to tackle the multi-robot localization problem, by letting select members of the team serve as landmarks for their teammates with different sensing modalities.
- 2) We propose an algorithm for the underlying active robot positioning problem that outperforms an approximate method based on a greedy algorithm. Furthermore, numerical results suggest that our algorithm matches

the accuracy of a heuristic evolutionary scheme for global optimization while having a significantly lower computational demand.

- 3) We validate our method in photorealistic simulations and real-world experiments using a robot team composed of one UAV and multiple UGVs as shown in Fig. 1. In particular, we show that our method can reduce VIO drift by as much as 90%.

The rest of the paper is organized as follows. Section II summarizes the related work. Section III gives a detailed problem formulation, followed by Section IV in which we present our approach. The analysis of our algorithm is presented in Section V, and its numerical performance is showcased in Section VI. The simulation and real-world experiments in Section VII ultimately demonstrate the efficacy of our method.

II. RELATED WORK

A. Vision-based State Estimation

Vision-based state estimation has been maturing and gaining popularity during the past decade. Powerful monocular-based state estimation algorithms such as the classical structure from motion algorithm and recent state-of-the-art monocular odometry methods [3], [4] can estimate camera poses and 3D structures. However, such algorithms cannot be directly used for robot navigation since the absolute scale of the world is not observable with a single camera. VIO algorithms can estimate poses in metric scale and run up to the IMU rate. Therefore, they are commonly used in robotics applications [5]. State-of-the-art VIO algorithms [6]–[9] have robust performance in high-speed 3D navigation with aggressive motions [10]–[13].

However, pure geometric-based methods have some intrinsic limitations: (1) The storage demand is high when maintaining a geometric map over a long trajectory. (2) Geometric-based features are sensitive to changes in viewpoint or lighting conditions. (3) It is difficult to distinguish features extracted from dynamic and static objects, leading to failures in dynamic environments. As a result, they can accumulate significant drift over long trajectories, which leads to unsafe behaviors and errors in mapping. Learning-enabled approaches are used to improve state estimation and mapping. Among them, one popular choice is to use semantic features because they are distinctive, informative, memory-efficient, and robust to viewpoint changes [14]–[20]. The proposed work can be categorized into semantic-based state estimation, where the semantic landmarks are UGVs.

B. Active Perception

Our work falls within the class of active perception and state estimation [21] problems. Some of the earliest methods modeled this task using the framework of partially observable Markov decision processes (POMDP) [22], [23]. This paradigm addresses synthesizing optimal policies, that is, functions mapping the history of observations of an agent to the optimal control for executing a given task. However, first examples of solutions to such problems were either developed for linear systems with Gaussian noise [24], or problems with a

small number of states [23]. Scaling this approach to realistic, and more complex robotic systems proved challenging. On the theoretical side, researchers showed that this should come as no surprise [25]: solving *generic* POMDPs is a PSPACE-hard problem. Furthermore, they showed that even with infinite precompute, an optimal controller, in general, might be challenging to implement in an efficient algorithm.

As a result, the community turned to solving approximate versions of the problem. The hope was that by acting optimally according to a proxy objective, perception-aware behavior would emerge. One of the first classes of methods involved belief space planning [26]–[32]. Effectively, this meant solving the open-loop planning version of the more difficult POMDP. Somewhat unfortunately, [25] also showed that solving this problem in general is NP-hard. To date, the majority of approaches based on BSP have been computationally intensive, and often challenging to implement for real-time use.

Recently, there has been a surge in interest in perception-aware motion planning for agile robots. Roughly speaking, this paradigm optimizes trajectories of agents, while ensuring they maintain a sufficient amount of perceptual information within the sensing region of their onboard sensors [33]–[37]. However, these approaches focus on single-agent settings. Their environment is fixed. When generalizing the methods to perception-aware navigation for robot teams, naturally, the same algorithms can be used for every member of the team. However, none of these works adopt the collaborative approach of altering the perceptual content of the environment created by the agents themselves. This is precisely the theme of our paper.

C. Active Team Localization and Target Tracking

There has been a substantial amount of work on the localization of robot teams using relative observation measurements. Many different versions of the problem exist [38]–[42] depending on the nature of allowed communication between teammates, as well as the type of measurement models (e.g. relative position, range, or bearing, etc...). Naturally, nonlinear measurement models render the resulting inference problem more computationally challenging. Recently, a number of papers have proposed robust algorithms based on convex relaxations that appear to be lossless for a non-negligible regime of noise levels [43], [44].

Prior work has also addressed the problem of multi-robot active SLAM [31], [45], [46], however, with several differences. For example, [31] does not leverage mutual observation of robots to aid localization. [46] relies on features in the environment to derive relative measurements between robots, a step that can fail in challenging, featureless environments. [45] assumes constant communication and perfect data association, which are not always satisfied in real-world settings. Most importantly, none of the former consider actively positioning landmarks to help robots localize themselves. The work of [47] considers a problem directly related to ours, albeit with a small team of homogeneous robots forming a predefined shape in 2D space. Subsequent works such as [48], [49] extend the

former, but provide no empirical or theoretical sub-optimality results. In this work, we consider a fine-grained measure of localization accuracy of a heterogeneous robot team based on a minimax localization cost. Furthermore, unlike previous works, we empirically show that a suitable gradient-descent-based algorithm matches the output of an evolutionary method for positioning perceptually-advantaged agents with significant accuracy.

Last but not least, our problem is intimately related to that of target tracking [50]–[52]. The latter problem involves positioning the set of tracker agents to allow them to form the best estimate of the target states. The parallel to our problem amounts to identifying tracking agents with lidar-equipped ones, and the targets with vision-aided agents. Typically, tracker agents communicate among themselves in order to leverage their collective measurements to position themselves optimally. Our setup effectively considers the dual of this problem. Other notable differences include the fact that we consider a collaborative setting, in addition to allowing no communication between agents except at the start of the mission.

D. Sensor Selection

Optimally placing the small set of lidar-equipped agents can be viewed as a sensor selection problem. Intuitively, this involves choosing out of a set of possible sensor locations the subset that gives minimal uncertainty in the estimates of the unknown quantity being measured subject to suitable, typically budget, constraints. In our example, the set is in fact not discrete, but comprises a whole continuum of possible positions in the environment in which we can place the lidar-equipped agents. The budget comes simply from the number of such agents at our disposal. It has been shown that approximately solving the discrete version of the sensor selection problem to an arbitrary level of suboptimality is NP-hard [53], [54]. Notable approaches based on the greedy paradigm that have nevertheless worked well in practice include [55]–[59]. However, we adopt a different approach based on nonlinear optimization.

III. ACTIVE COLLABORATIVE TEAM LOCALIZATION

Our robot team is partitioned into two subsets of agents, \mathcal{L} and \mathcal{C} . Agents in set \mathcal{L} , also referred to as “ \mathcal{L} -agents”, are expensive lidar-equipped agents that can autonomously position themselves accurately without additional external localization infrastructure. Set \mathcal{C} comprises of inexpensive, resource-constrained agents (“ \mathcal{C} -agents”) that predominantly rely on their onboard cameras for state estimation. However, cameras can be used to obtain accurate state estimates only when there is a sufficient quantity of reliable visual cues in the environment. We propose a solution for accurate team localization robust to whether or not the latter condition holds. In particular, our approach involves leveraging \mathcal{L} -agents as artificial landmark features \mathcal{C} -agents can use for state estimation in desired regions of space.

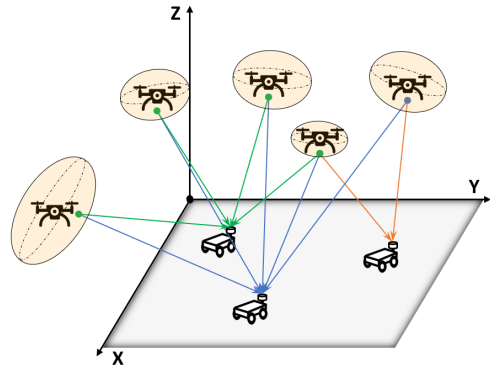


Figure 2: In the absence of reliable visual features or landmarks in the environment, \mathcal{C} -agents (UAVs) additionally triangulate their positions using bearing measurements to \mathcal{L} -agents (UGVs). We position \mathcal{L} -agents to minimize the localization uncertainty (ellipses) of \mathcal{C} -agents.

In light of the setup $|\mathcal{L}| \ll |\mathcal{C}|$ motivated by budget considerations, it is intuitively clear that certain configurations of \mathcal{L} -agents are better than others. For example, positioning them in degenerate configurations (such as a single point) would result in worse performance than spreading them further apart. On the other hand, placing \mathcal{L} -agents too far would supply \mathcal{C} -agents with bearing measurements with near-vanishing sensitivity to change in their positions. Furthermore, tailoring the positioning to enable accurate vision-aided localization in one region of space might render it useless for state estimation in another.

We therefore center our approach around a minimax optimization problem stated as follows. Given N representative points of operating regions of \mathcal{C} -agents (without loss of generality one point per each \mathcal{C} -agent so that $|\mathcal{C}| = N$), and a given budget on the number of \mathcal{L} -agents, say $|\mathcal{L}| \leq M$, position the \mathcal{L} -agents in a way that minimizes the maximum localization uncertainty across all \mathcal{C} -agents. Identifying \mathcal{C} and \mathcal{L} with $[N]$ and $[M]$, respectively, and denoting the locations of \mathcal{C} -agents and \mathcal{L} -agents by $(\mathbf{x}_i)_{i \in [N]} \in (\mathbb{R}^3)^N$ and $(\mathbf{z}_j)_{j \in [M]} \in (\mathbb{R}^3)^M$, respectively, and the covariance of the resulting state estimate at location \mathbf{x}_i by $\Sigma(\mathbf{x}_i; \mathbf{z}_{1:M})$, we have

Problem 1.

$$\begin{aligned} \min_{\mathbf{z}_{1:M} \in (\mathbb{R}^3)^M} \max_{i \in [N]} \text{tr}(\Sigma(\mathbf{x}_i; \mathbf{z}_{1:M})) \\ \text{s.t.} \\ \|\mathbf{z}_j\|_2 \leq R_{\max} \quad \forall j \in [M] \\ \|\mathbf{z}_j - \mathbf{x}_i\|_2 \geq R_{\min} \quad \forall i \in [N], \forall j \in [M] \end{aligned}$$

The latter two constraints encode the desideratum that we can only position \mathcal{L} -agents in a certain region of space, as well as the fact that no \mathcal{C} -agent can be closer than a minimum prescribed distance to any \mathcal{L} -agent. We measure the localization accuracy of each \mathcal{C} -agent using the trace of the covariance of its state estimate. This roughly captures its

total uncertainty across all three spatial dimensions. Overall, our problem is novel in two ways:

- 1) we propose the paradigm of enabling predominantly vision-driven robot teams to navigate perceptually-challenging environments by using select perceptually-advantaged teammates as visual landmarks;
- 2) furthermore, we model the task using a minimax active robot positioning problem in continuous space; previous work on the topic has typically focused on its discrete version, without a robust minimax objective.

Remark. Two special cases of our setup deserve particular mention. One involves positioning \mathcal{L} -agents in order to ensure high localization accuracy of \mathcal{C} -agents along pre-planned trajectories. This scenario is subsumed by Problem 1, as can be seen by placing fictitious \mathcal{C} -agents (Fig 3) at regular “discretization” points along paths to be taken by “real” \mathcal{C} -agents. The second involves using \mathcal{L} -agents to provide visual cues at a well-dispersed sample of positions for the purpose of coverage of a given environment. The latter instantiation of Problem 1 amounts to placing fictitious \mathcal{C} -agents at sampled points.

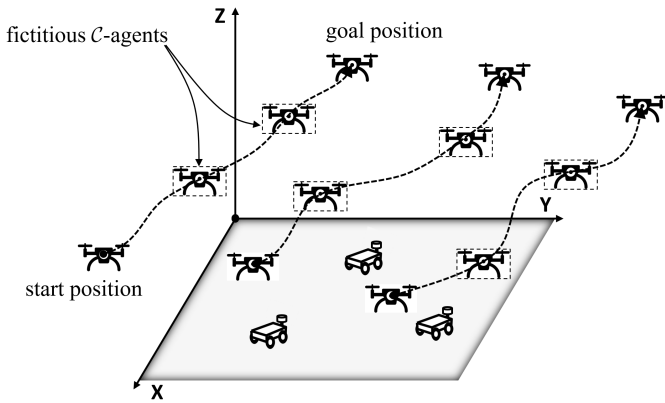


Figure 3: Placing sufficiently many *fictitious* \mathcal{C} -agents along planned trajectories of real \mathcal{C} -agents to ensure the latter have accurate state estimates.

Regarding communication, we assume the following protocol. The team rallies together at the start of the mission, has one agent solve the problem in a centralized way, communicate the positions of \mathcal{L} -agents to the rest of the team, and then the mission starts. From that moment onward, no explicit communication between teammates takes place.

IV. ACTIVE POSITIONING ALGORITHM

A. Modelling Sensing and Localization Uncertainty

The observation model of each \mathcal{C} -agent comprises a set of bearing measurements. In particular, \mathcal{C} -agent i at point $\mathbf{x} \in \mathbb{R}^3$ in the vicinity of setpoint \mathbf{x}_i registers the bearing to \mathcal{L} -agent j at position $\mathbf{z}_j \in \mathbb{R}^3$ in the form:

$$\mathbf{y}_{i,j} = \mathbf{h}(\mathbf{x}, \mathbf{z}_j) + \mathcal{N}(0, \sigma_m^2 \mathbf{I}_3). \quad (1)$$

Here $\mathbf{h}(\mathbf{x}, \mathbf{z}) := \frac{\mathbf{z}-\mathbf{x}}{\|\mathbf{z}-\mathbf{x}\|_2}$ denotes the noiseless bearing measurement function. We assume sensor noise is normally distributed with standard deviation $\sigma_m > 0$, and is independent across different \mathcal{C} - \mathcal{L} agent pairs.

A critical aspect of our problem involves the *nonlinear* sensing model in Equation 1. As a result, we now describe the way in which \mathcal{C} -agents translate such measurements into estimates of their pose. We take the route based on the paradigm of extended Kalman filtering (EKF). In particular, when the \mathcal{C} -agent at position \mathbf{x} registers bearings to \mathcal{L} -agents at positions $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_M \in \mathbb{R}^3$, its collection of measurements may be succinctly represented via

$$\mathbf{y}_{1:M} = \mathbf{H}(\mathbf{x}; \mathbf{z}_{1:M}) + \boldsymbol{\varepsilon}, \quad (2)$$

where $\mathbf{H}(\mathbf{x}; \mathbf{z}_{1:M}) = [\mathbf{h}^T(\mathbf{x}, \mathbf{z}_1), \mathbf{h}^T(\mathbf{x}, \mathbf{z}_2), \dots, \mathbf{h}^T(\mathbf{x}, \mathbf{z}_M)]^T$, and $\boldsymbol{\varepsilon} = [\varepsilon_1^T, \varepsilon_2^T, \dots, \varepsilon_M^T]^T$ are i.i.d. Gaussian random variables per Equation 1. Assuming a $\mathcal{N}(\mathbf{x}_i, \bar{\Sigma})$ prior on \mathbf{x} , the agent approximates the bearing measurement to agent j via the linearized measurement model

$$\bar{\mathbf{y}}_{i,j} + \Delta \mathbf{y}_{i,j} = \underbrace{\mathbf{h}(\mathbf{x}_i, \mathbf{z}_j)}_{=: \bar{\mathbf{y}}_{i,j}} + \frac{\partial \mathbf{h}}{\partial \mathbf{x}}(\mathbf{x}_i, \mathbf{z}_j) \underbrace{\Delta \mathbf{x}}_{=: \mathbf{x} - \mathbf{x}_i} + \boldsymbol{\varepsilon}_i, \quad (3)$$

so that its posterior distribution is approximately Gaussian with the following information (inverse covariance) matrix

$$\mathcal{J}(\mathbf{x}_i; \mathbf{z}_{1:M}) = \bar{\Sigma}^{-1} + \sigma_m^{-2} \sum_{j=1}^M \frac{\partial \mathbf{h}^T}{\partial \mathbf{x}}(\mathbf{x}_i, \mathbf{z}_j) \frac{\partial \mathbf{h}}{\partial \mathbf{x}}(\mathbf{x}_i, \mathbf{z}_j). \quad (4)$$

Recalling the objective function of Problem 1, we define

$$\Sigma(\mathbf{x}_i; \mathbf{z}_{1:M}) = \mathcal{J}^{-1}(\mathbf{x}_i; \mathbf{z}_{1:M}). \quad (5)$$

Regarding the sensing model and pose uncertainty of every \mathcal{C} -agent, we make two assumptions hereon:

- 1) \mathcal{C} -agents have accurate orientation estimates;
- 2) their onboard cameras are omnidirectional.

Remarks For the first assumption, we follow the approximation made in [57]. Intuitively, it posits that if bearing measurements arrive at a sufficiently high frequency (approximately every 5 seconds in our experiments), the gyroscope can be used to accurately estimate incremental changes in orientation between successive keyframes. At latter points, visual information is fused with IMU measurements to bring the orientation uncertainty down to a negligible value. The second assumption is valid when \mathcal{C} -agents are equipped with multiple cameras that together cover a 4π (in steradians) field of view (FOV). It can also be *emulated* by agents equipped with a camera with a limited FOV by periodically performing a 2π yawing motion at specified setpoints. In our physical experiments, we take the latter route.

B. Optimization Algorithm

Problem 1 is a challenging nonlinear optimization task. Nevertheless, several practical approaches come to mind. One method involves discretizing the continuous problem with a grid of points sampled from its feasible (“obstacle-free”) region. Starting from an empty set, this algorithm then greedily

picks a subsequent grid point that induces the maximum decrease in the objective value, until a given number of \mathcal{L} -agents have been positioned. A drawback of this approach is that it does not result in provably globally optimal solutions, while incurring a non-negligible computational cost - typically linear in the number of discretization points. The latter generally has to increase with the desired level of accuracy. Another method involves a heuristic evolutionary procedure for global optimization [60]. In spite of promising empirical performance, this algorithm can demand long compute times and it does not come with global optimality guarantees. Finally, the third class of approaches includes local gradient-descent based algorithms for smooth optimization. This class of methods has shown impressive results, scaling to high-dimensional optimization problems involving training neural networks. Nevertheless, their computational performance can vary significantly depending on the exact algorithm being used, as well as the structural properties of the problem being solved.

Our method belongs to the class of gradient-descent-based approaches. However, further specific design choices are still required. In particular, Problem 1 is a non-smooth non-convex optimization problem. Lack of global smoothness stems from the minimax form of the objective function that can dramatically slow down the rate of convergence of local algorithms for smooth optimization. The other challenging aspect of Problem 1 is its non-convexity. Indeed, we show there exist instances of the problem that exhibit multiple strict local optima with wildly different objective values. One such example is shown in Fig. 4. In what follows, we show how to tackle each of these hurdles in turn.

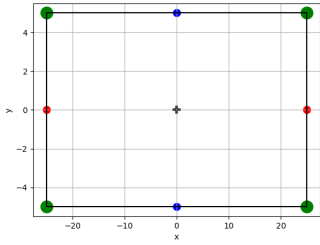


Figure 4: Two strict local minima of the objective function of the problem involving minimizing the maximum localization uncertainty of four \mathcal{C} -agents operating at setpoints depicted by green circles with the help of two \mathcal{L} -agents. The blue local optimum is more than 500% worse than the red local optimum. The cross at the centroid of the rectangle is a saddle point.

To ensure our objective function is globally smooth, we introduce two modifications to the original problem. The first involves choosing a positive real number $\delta \ll R_{min}$, and modifying the noiseless bearing measurement function via:

$$\mathbf{h}(\mathbf{z}, \mathbf{x}) = \frac{\mathbf{z} - \mathbf{x}}{\|\mathbf{z} - \mathbf{x}\|_2} \rightsquigarrow \mathbf{h}_\delta(\mathbf{z}, \mathbf{x}) = \frac{\mathbf{z} - \mathbf{x}}{\sqrt{\|\mathbf{z} - \mathbf{x}\|_2^2 + \delta^2}}. \quad (6)$$

Using the same methodology of extended Kalman filtering, we make the corresponding changes to the posterior covariance estimates of agents at corresponding setpoints. For example, the localization uncertainty of the \mathcal{C} -agent operating at setpoint i is modified through

$$\mathcal{J}_\delta(\mathbf{x}_i; \mathbf{z}_{1:M}) = \bar{\Sigma}^{-1} + \sigma_m^{-2} \sum_{j=1}^M \frac{\partial \mathbf{h}_\delta^T}{\partial \mathbf{x}}(\mathbf{x}_i, \mathbf{z}_j) \frac{\partial \mathbf{h}_\delta}{\partial \mathbf{x}}(\mathbf{x}_i, \mathbf{z}_j) \quad (7)$$

$$\Sigma_\delta(\mathbf{x}_i; \mathbf{z}_{1:M}) = \mathcal{J}_\delta^{-1}(\mathbf{x}_i; \mathbf{z}_{1:M}).$$

As a result, we solve

Problem 2.

$$\begin{aligned} \min_{\mathbf{z}_{1:M} \in (\mathbb{R}^3)^M} \max_{i \in [N]} \text{tr}(\Sigma_\delta(\mathbf{x}_i; \mathbf{z}_{1:M})) \\ \text{s.t.} \\ \|\mathbf{z}_j\|_2 \leq R_{max} \quad \forall j \in [M] \\ \|\mathbf{z}_j - \mathbf{x}_i\|_2 \geq R_{min} \quad \forall i \in [N], \forall j \in [M] \end{aligned}$$

The second modification involves reformulating the minimax Problem 2, in general one with non-smooth objective, into a smooth problem. We introduce a new decision variable $t \in \mathbb{R}$, and consider the following equivalent

Problem 3.

$$\begin{aligned} \min_{\mathbf{z}_{1:M} \in (\mathbb{R}^3)^M, t \geq 0} t \\ \text{s.t.} \\ \text{tr}(\Sigma(\mathbf{x}_i; \mathbf{z}_{1:M})) \leq t \quad \forall i \in [N] \\ \|\mathbf{z}_j\|_2 \leq R_{max} \quad \forall j \in [M] \\ \|\mathbf{z}_j - \mathbf{x}_i\|_2 \geq R_{min} \quad \forall i \in [N], \forall j \in [M] \end{aligned} \quad (8)$$

Intuitively, the fact that t is being minimized in conjunction with the first set of constraints will effectively make it take the value of the objective of Problem 2. Problem 3 is still a non-convex optimization problem. For this reason, we solve it using an interior point solver, IPOPT [61], to which we pass analytic derivatives of constraints of zeroth, first, and second orders. Here we leverage the fact that the function

$$J \rightarrow \text{tr}(J^{-1}) \quad (9)$$

is a smooth map on the space of positive *definite* matrices (in this case in $\mathbb{R}^{3 \times 3}$). Furthermore, assuming that J is a smooth, positive definite function of a set of parameters $\theta \in \mathbb{R}^k$, we have the following relations

$$\nabla_{\theta_i} \text{tr}(J^{-1}(\theta)) = (-1) \text{tr}(J^{-2}(\theta)) \frac{\partial J}{\partial \theta_i}(\theta), \quad (10)$$

and

$$\begin{aligned} \nabla_{\theta_i, \theta_j}^2 \text{tr}(J^{-1}(\theta)) = \\ -\text{tr}(J^{-2}(\theta)) \frac{\partial^2 J}{\partial \theta_i \partial \theta_j}(\theta) + 2 \text{tr}(J^{-2}(\theta)) \frac{\partial J}{\partial \theta_i}(\theta) J^{-1}(\theta) \frac{\partial J}{\partial \theta_j}(\theta) \end{aligned} \quad (11)$$

V. ALGORITHM ANALYSIS

A. Price of Smoothing

We now elaborate on our choice of $\delta > 0$ in Equation 6. Indeed, any $\delta > 0$ introduces fictitious information about the range from a \mathcal{C} -agent at position \mathbf{x} to a \mathcal{L} -agent at position \mathbf{z} . This follows from the fact that the directional derivative of \mathbf{h}_δ along the bearing vector $\frac{\mathbf{z}-\mathbf{x}}{\|\mathbf{z}-\mathbf{x}\|_2}$ is *non-vanishing* whenever $\delta > 0$.

However, at a fixed range (i.e. fixed $\|\mathbf{z}-\mathbf{x}\|_2$), the influence of δ decreases as $\delta \downarrow 0$. Intuitively, this implies that at any point in feasible regions of Problems 1 and 2 (which coincide), the values of the objective functions of the two problems match each other in the limit $\delta \downarrow 0$. At this stage two questions remain:

- 1) determine a suitable range (i.e. upper bound) of δ
- 2) quantify the suboptimality of solving Problem 2 in place of Problem 1 as a function of δ .

To answer these questions, we develop a result in the active robot positioning problem that is robust to measurement model mismatch. The key result that enables such analysis is the following

Theorem V.1. *Consider the setup of Problems 1 and 2. Let $\eta > 0$ and $\zeta > 1$ be an arbitrary pair of real numbers. Define*

$$s_0 = \frac{\eta}{\eta+1} \frac{R_{min}^2}{M} \sigma_m^2 (1 + \zeta^2). \quad (12)$$

Then, choosing any $\delta \in (0, R_{min}/\zeta)$, we have the implication

$$\begin{aligned} \forall \mathbf{x} \in \mathbb{R}^3 : \text{tr}(\mathcal{J}_\delta(\mathbf{x}; \mathbf{z}_{1:M})^{-1}) = s \leq s_0 \\ \Rightarrow \frac{s}{1+\eta} \leq \text{tr}(\mathcal{J}_0(\mathbf{x}; \mathbf{z}_{1:M})^{-1}) \leq s(1+\eta). \end{aligned} \quad (13)$$

Furthermore, the same implication holds with the roles of \mathcal{J}_δ and \mathcal{J}_0 reversed.

Proof. See Appendix in Section X. \square

Corollary 1. *Within the setup of Theorem V.1, choose any $\delta \in (0, R_{min}/\zeta)$. Denote the values of Problems 1 and 2 by V^0 and V^δ , respectively. Suppose the sublevel set*

$$\mathcal{S} := \{\mathbf{z}_{1:M} \mid V^\delta(\mathbf{z}_{1:M}) \leq s_0\} \quad (14)$$

is not empty. Then the optimal solution of Problem 2 is at most a $(1+\eta)^2$ -suboptimal solution of Problem 1.

Proof. See Appendix in Section X. \square

The significance of Corollary 1, is that if there exists a solution to the Problem 2 with a sufficiently low objective value, then that solution is also approximately optimal for the original problem. To choose an appropriate value of ζ , we set s_0 to be the square of the largest σ -uncertainty ball of each of the \mathcal{C} -agents that we are willing to tolerate. Naturally, this quantity can depend on the scale of the environment, and its density of obstacles. Here we call it R_{tol} . We then set

$$R_{tol}^2 \leq \frac{\eta}{\eta+1} \frac{R_{min}^2}{M} \sigma_m^2 (1 + \zeta^2), \quad (15)$$

and so it is enough to choose

$$\zeta^2 = \left(\frac{R_{tol}}{R_{min}} \right)^2 M \sigma_m^{-2} \frac{2}{\eta}. \quad (16)$$

Some ‘‘physically reasonable’’ values in the expression above involve setting

$$\frac{R_{tol}}{R_{min}} \rightarrow 10, M \rightarrow 10, \sigma_m \rightarrow 10^{-2}, \eta \rightarrow 10^{-1} \quad (17)$$

thus leading to

$$\zeta^2 = 2 \times 10^9. \quad (18)$$

In practice, we cap ζ to 10^5 out of concern for numerical precision.

B. Computational Complexity

The computational complexity of the algorithm is determined by two factors. Firstly, it depends on the complexity of computing derivatives of the desired order. In our case, all derivatives of order up to two are computed in $O(M^2N)$ time. Second, both the number of iterations, as well as the effort of computing gradient descent steps within the IPM influence the running time of the algorithm. The total number of variables and constraints is $O(MN)$, and so, roughly speaking, every gradient descent step is computed by inverting a square matrix of dimension $O(MN)$. For dense matrices, this is done in time at most $O((MN)^3)$. However, IPOPT can exploit the sparsity pattern of the constraints to solve such systems more efficiently. Nevertheless, we leave more refined (favorable) estimates of running time for future work. Assuming the total number of iterations is K , the running time of the algorithm is of order at most $O(K(MN)^3)$.

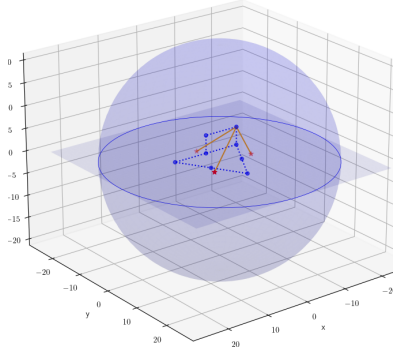
C. Limitations

Despite encouraging performance on random data (see Table I), our numerical algorithm is not perfect. Firstly, as a local-optimization-based method, it can get stuck in suboptimal local minima, such as in scenarios shown in Fig. 4. Second, our algorithm is a centralized solution. Although it empirically scales favourably with the size of robot teams we consider, we note that the current method cannot handle scenarios with an a priori unknown number of spatially distributed \mathcal{C} -agents. We leave addressing these limitations for future work.

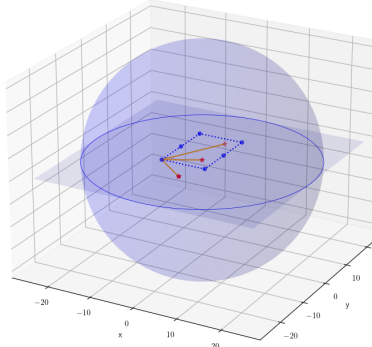
VI. NUMERICAL ANALYSIS

In this section, we test the numerical performance of our algorithm. We compare it against two baseline approaches, one a greedy approximate algorithm, and the other a heuristic evolutionary procedure for global optimization [60] as implemented in the package NLOPT [62]. First, we look at the morphology of solutions generated by our algorithm, and then turn to comparing its accuracy and time efficiency compared to the other two methods.

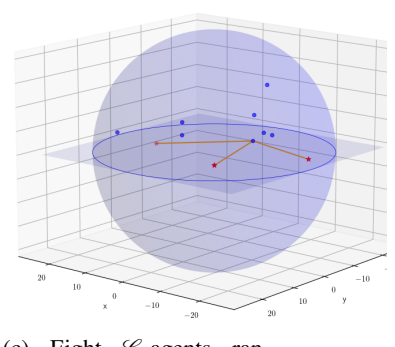
We begin by looking at the form of solutions of several robot team problems with the following parameters. The prior covariance is set to $\Sigma \rightarrow 30 \times I_3$, and the measurement standard



(a) Eight \mathcal{C} -agents flying in formation.



(b) Six \mathcal{C} -agents on two straight line paths.



(c) Eight \mathcal{C} -agents randomly positioned inside the upper hemisphere.

Figure 5: Examples of \mathcal{L} -agents positioned by our method. Blue dots are centered on each \mathcal{C} -agent/setpoint, while the red stars denote \mathcal{L} -agents. Orange lines show bearings from one of the \mathcal{C} -agents to all \mathcal{L} -agents.

N	5			10			20		
M	2	5	10	2	5	10	2	5	10
ours	-0.002	-0.01	-0.02	-0.001	-0.01	-0.03	-0.002	-0.01	-0.03
greedy	0.25	0.48	0.54	0.23	0.48	0.52	0.25	0.37	0.41

Table I: The suboptimality of the approaches for various team sizes. We measure the suboptimality (loss of accuracy) of a solution in terms of its relative value compared to the objective achieved by the evolutionary optimization scheme.

deviation $\sigma_m \rightarrow 0.1$. We consider three problems, each involving three \mathcal{L} -agents, but varying numbers and configurations of \mathcal{C} -agents. Some characteristic solutions appear in Fig. 5.

A. Accuracy

Here we show the comparison of the objective values achieved by our algorithm and the two alternate approaches on a range of different problems, differing in the number of agents, and the amount of noise in the sensor measurements.

First, we set the noise to a set value, namely $\Sigma \rightarrow 30 \times I_3$ and $\sigma_m \rightarrow 0.1$, and vary the size of the team. In particular, we look at the performance of the algorithms as the number of \mathcal{C} -agents (denoted by N) and \mathcal{L} -agents (denoted by M) varies according to $N \in \{5, 10, 20\}$ and $M \in \{2, 5, 10\}$. For every (N, M) pair, we evaluate each algorithm on ten different random configurations of \mathcal{C} -agents, sampled from a sphere of radius 40 m, and perform ten independent algorithm runs for each instance. Table I shows that the second-order method attains objectives matching those of the evolutionary algorithm on random data, and consistently outperforms the greedy approach - in some cases by as much as 50%.

Second, we test the accuracy of the algorithm on a problem with $N = 10$ \mathcal{C} -agents and $M = 5$ \mathcal{L} -agents, and varying amounts of sensing noise. For each level of noise, we run each algorithm on ten different random instances, generated as before. Table II shows that our algorithm produces solutions that match the quality of those output by the evolutionary optimization scheme. Similarly as before, our algorithm con-

σ_m	1	2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}
ours	-0.0002	-0.003	-0.003	-0.013	-0.025	-0.043
greedy	0.039	0.116	0.272	0.417	0.581	0.624

Table II: Suboptimality of our algorithm and the greedy algorithm with respect to the solution produced by the evolutionary algorithm.

sistently outperforms the greedy baseline, sometimes by as much as 50%.

B. Compute Speed

Here we show the timings of the three procedures on a problem involving ten \mathcal{C} -agents and five \mathcal{L} -agents. Table II shows that our approach produces solutions that match the solution of the evolutionary algorithm to two significant digits at a fraction of the running time, as illustrated in Table III.

M	2	5	10
ours	0.079	0.108	0.218
greedy	1.072	2.605	4.881
global	0.663	4.970	30.029

Table III: Computation time (in seconds) of our algorithm and the two alternate approaches.

VII. EXPERIMENTS

The setup for our simulation and real-world experiments involves a team of multiple UGVs and UAVs. To analyze the

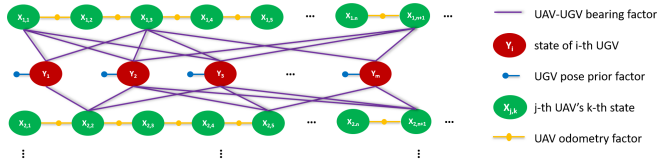


Figure 6: **Active collaborative team localization factor graph representation.** In our experiments, since the UGVs are static, each UGV is assigned exactly one state in the factor graph.

influence of the degree of structure of the environment and test the robustness of the system, we performed experiments across three types of environments, i.e., structured, semi-structured, and unstructured. We keep the same mission specification for real-world and simulated experiments in order to analyze the influence of real-world noise in controlled settings. Concretely, the UAV’s waypoints are identical in real-world and simulation experiments. We identify the concept of waypoints and setpoints from Section III. Instead of executing these waypoints once, the UAV is commanded to execute multiple rounds so that we can evaluate the robustness of our method for long-range missions.

The UGVs serve the role of \mathcal{L} -agents, and the UAV that of a \mathcal{C} -agent. The UGVs move along the ground plane at all times. To ensure the robustness of the state estimation module for our team of agents, we add a further constraint stipulating that at each setpoint, there must exist one yaw angle from which the UAV can view *all* UGVs. Introducing additional variables, the unit heading vectors $\mathbf{n}_1, \mathbf{n}_2, \dots, \mathbf{n}_N \in S^1$ along such yaw angles, the constraint-augmented problem becomes:

$$\begin{aligned} \min_{\mathbf{z}_{1:M} \in (\mathbb{R}^2)^M, \mathbf{n}_{1:N} \in (S^1)^N} \max_{i \in [N]} \text{tr}(\Sigma(\mathbf{x}_i; \mathbf{z}_{1:M})) \\ \text{s.t.} \\ \|\mathbf{z}_j\|_2 \leq R_{\max} \quad \forall j \in [M] \\ \|\mathbf{z}_j - \mathbf{x}_i\|_2 \geq R_{\min} \quad \forall i \in [N], \forall j \in [M] \\ \angle(\mathbf{n}_i, \mathbf{z}_j - \mathbf{x}_i) \leq \frac{\alpha}{2} \quad \forall i \in [N], \forall j \in [M]. \end{aligned} \quad (19)$$

The solution proceeds similarly to before. The addition of the aforementioned set of constraints translates into

$$\mathbf{n}_i \cdot (\mathbf{z}_j - \mathbf{x}_i) \geq \cos\left(\frac{\alpha}{2}\right) \sqrt{\|\mathbf{z}_j - \mathbf{x}_i\|^2 + \delta^2}, \quad (20)$$

which is a smooth approximation that becomes ideal in the limit as $\delta \downarrow 0$. Larger δ provides more conservative approximations of the field of view.

The rest of this section is organized as follows. The first subsection provides an overview of the software-hardware system used for our experiments. The second subsection presents simulation experiments, followed by real-world experiments.

A. System overview

System overview (hardware): In real-world experiments, our system consists of multiple UGVs and a UAV as shown

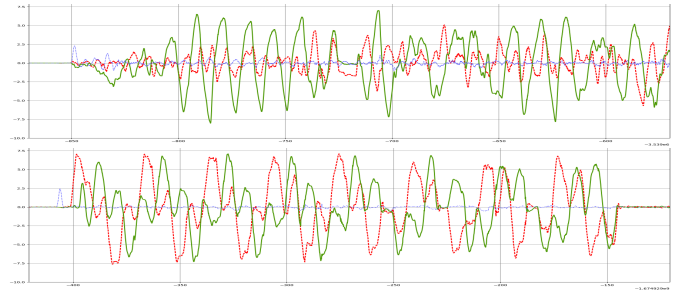


Figure 7: **Velocity profiles** for real-world experiments (top) and simulated experiments (bottom) illustrating aggressive maneuvers in experimentation and simulation. The curves show the velocities along X (dotted red), Y (solid green), and Z (dash-dotted blue) axes for flights in the unstructured environment. The vertical axis is the velocity, which ranges from -10 m/s to +10 m/s. The horizontal axis is the timestamp in seconds. The UAV accelerates drastically and frequently. Such aggressive motions pose significant challenges to state estimation.

in Fig. 1. The UGVs are equipped with ouster OS1-64 lidars. The UAV platform is equipped with a hardware-synchronized collection of greyscale stereo cameras, an IMU, and an RGB monocular camera. The greyscale stereo cameras and the IMU are used for stereo VIO [7]. In addition, the UAV platform also has an ouster OS1-64 lidar, but it is only used for obstacle avoidance; the UAV’s localization relies only on cameras and the IMU. Our UGV and UAV platforms are capable of autonomous navigation in cluttered and GPS-denied environments using only onboard sensing and computation. Note that only the monocular RGB camera is used for detecting UGVs and generating bearing measurements.

We carry out two sets of simulation experiments. In the first set, we have 1 UAV with 2 UGVs and 1 UAV with 3 UGVs. This allows us to study how much performance gain we can obtain by adding more UGVs into the environment. In the second set, we have 10 UAVs with 10 UGVs, which allows us to analyze how well our method can generalize to larger robot teams. To simulate real-world perception, we add noise to the UAV’s camera measurements and odometry, as well as introducing dynamic objects in the environment.

System overview (estimation pipeline): Our estimation pipeline builds upon a factor-graph-based method [63], [64], which takes in VIO estimates and bearing measurements and outputs the UAVs’ SE3 pose estimates. The choice of a factor graph approach over an EKF for the experiments is due to its better accuracy and consistency. The structure of our factor graph is shown in Fig. 6, which consists of two kinds of nodes (UAV and UGV poses) and three kinds of factors (UAV odometry factors, UAV-UGV bearing factors, and UGV pose prior factors). For the UAV odometry factors, we use the stereo-MSCKF algorithm [7] to estimate the relative transformation between two consecutive key poses, i.e., $\mathbf{p}_i^{vio} \ominus \mathbf{p}_{i-1}^{vio}$.

The generation of bearing factors is different between

Environment / Traj. Length	Drift Red.	Ours Drift (X/Y/Z) (m)	Rand. UGVs Drift (X/Y/Z) (m)	VIO Drift (X/Y/Z) (m)
Structured (2 UGVs) / 1.4 km	86.29%	2.92 (-1.21 / -1.61 / +2.11)	4.16 (-3.70 / -0.33 / +1.87)	21.27 (-12.74 / -8.85 / +14.56)
Structured (3 UGVs) / 1.4 km	96.19%	0.81 (-0.77 / +0.20 / +0.16)	2.84 (-1.36 / -2.47 / +0.30)	21.27 (-12.74 / -8.85 / +14.56)
Semi-Structured (2 UGVs) / 1.0 km	86.40%	3.21 (-0.01 / -2.73 / -1.68)	15.00 (-9.38 / +11.69 / +0.70)	23.57 (-21.85 / -6.29 / +6.22)
Semi-Structured (3 UGVs) / 1.0 km	88.38%	2.74 (+0.15 / -1.69 / -2.15)	8.01 (-6.20 / +5.04 / +0.59)	23.57 (-21.85 / -6.29 / +6.22)
Unstructured (2 UGVs) / 1.4 km	60.64%	7.50 (-2.77 / +6.95 / -0.51)	16.04 (-2.45 / +15.85 / -0.37)	19.05 (-1.15 / +19.01 / +0.56)
Unstructured (3 UGVs) / 1.4 km	74.40%	4.88 (-1.77 / +4.10 / -1.96)	11.53 (-2.91 / +11.12 / -0.94)	19.05 (-1.15 / +19.01 / +0.56)

Figure 8: Quantitative results for Unity-based photo-realistic simulation experiments. The performance improvement achieved by detecting and using UGVs as bearing constraints (3rd column) over using VIO alone (5th column) is shown in the 2nd column. All flight missions are autonomous with aggressive motions and a maximum design velocity of 13 m/s, as shown in the example in Fig. 7. The average drift reduction is 77.74% with 2 UGVs and 86.32% with 3 UGVs across all environments. Random positioning of UGVs produces much worse results (4th column) than our method. Therefore, the proposed method is superior to using VIO alone as well as randomly positioning UGVs.

UAV Index / Traj. Length	Drift Red.	Ours Drift (X/Y/Z) (m)	VIO Drift (X/Y/Z) (m)
1st / 0.3087 km	75.35%	1.31 (+1.00 / -0.83 / +0.12)	5.29 (+0.46 / -5.25 / -0.51)
2nd / 0.3835 km	80.88%	1.80 (-1.57 / +0.74 / -0.46)	9.39 (-8.44 / +3.99 / -1.00)
3rd / 0.9042 km	62.68%	4.91 (+0.50 / -4.88 / -0.19)	13.15 (-2.40 / -12.33 / -3.90)
4th / 0.1232 km	57.73%	0.82 (-0.30 / -0.01 / +0.76)	7.30 (+1.47 / -7.13 / +0.51)
5th / 0.3353 km	60.04%	2.77 (-2.67 / -0.47 / -0.58)	6.94 (-5.67 / -3.83 / -1.15)
6th / 0.1947 km	68.22%	2.98 (+1.05 / -2.78 / -0.18)	9.37 (-5.17 / -7.73 / -1.12)
7th / 0.5200 km	81.69%	1.60 (-1.12 / +1.11 / -0.26)	8.73 (-6.21 / +6.08 / +0.80)
8th / 0.3310 km	62.98%	2.73 (-1.24 / +2.39 / +0.43)	7.37 (-6.04 / +4.15 / +0.74)
9th / 0.5872 km	79.10%	2.59 (-1.69 / -1.82 / -0.73)	12.39 (+1.17 / -12.07 / +2.52)
10th / 0.4848 km	86.61%	1.28 (-0.60 / -1.13 / +0.05)	9.56 (+3.54 / -8.81 / -1.13)

UAVs	Drift Red. Mean	Drift Red. Std. Dev.	Traj. Length Median	Traj. Length Sum
1st-10th	71.50%	9.85%	0.3594 km	4.1726 km

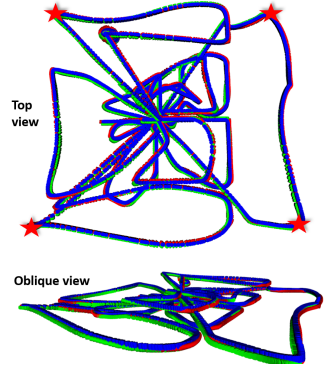


Figure 9: **Left: Quantitative results for simulation experiments with 10 UAVs and 10 UGVs in the unstructured environment.** The average drift correction across all UAVs is 71.50%. The median trajectory length for each UAV is 359.4 m, which is significantly shorter than trajectories in Fig. 8. This leads to a smaller VIO drift and thus less room for drift reduction. The sum of all UAVs' trajectories is 4.2 km. **Right: Trajectories for all 10 UAVs.** Each UAV operates at a slightly different altitude and in a different part of the environment. Red stars show the corners of the environment.

real-world and simulation experiments. For simulation experiments, the bearing measurements are generated by first calculating the relative bearing based on the ground-truth UGV and UAV poses and then adding Gaussian noise. For real-world experiments, the bearing factors are based on UGV detections. Since object detection is not the focus of our work, we put colored flags on the UGVs and detect them by image processing techniques including color filtering, opening, connected component analysis, and bounding box fitting. Data association is then carried out by projecting the 3D positions of the UGV centers back onto the image plane. The camera pose used during this projection process is $\hat{\mathbf{p}}_t = \mathbf{p}_{t-1}^g \oplus (\mathbf{p}_t^{vio} \ominus \mathbf{p}_{t-1}^{vio})$, i.e., the composition of the latest key pose from the factor graph (\mathbf{p}_{t-1}^g) and the relative motion estimated by the VIO.

The UGV pose prior factor is generated by integrating poses estimated by a state-of-the-art lidar-inertial odometry algorithm [65]. Since the UGVs' trajectories are much shorter in distance and less aggressive in motion, the lidar-inertial odometry can estimate their poses with only centimeter-level drift, which is several orders of magnitude smaller than the UAV's VIO drift. We use the GTSAM library [63], [64] as

the backend.

B. Simulation Experiments



Figure 10: **Simulation experiment environments:** Structured (left), semi-structured (middle), unstructured (right).

Experiment environments: We use a custom Unity-based simulator. The simulator is integrated with ROS and can simulate photorealistic sensor data such as RGB and depth images, and lidar point clouds. We choose the simulation environments that approximate the real-world experiment environments, as shown in Fig. 10. The structured environment is a city-like environment. The semi-structured environment is a rural area with flooded grounds and some structures like farmhouses, bridges, and cars. The unstructured environment is a sparse forest. These simulation environments have dynamic objects such as grass, water, and leaves.

Environment / Traj. Length	Drift Reduction	Ours Drift (X/Y/Z) (m)	VIO Drift (X/Y/Z) (m)
Structured / 2 km	42.30%	4.49 (+4.22 / +0.62 / -1.41)	7.79 (+7.02 / +2.29 / -2.47)
Semi-Structured / 1 km	68.28%	1.85 (+0.46 / +0.27 / -1.77)	5.83 (+3.10 / +4.26 / -2.49)
Semi-Structured Aggressive / 2.5 km	70.89%	3.66 (-1.58 / +3.17 / -0.92)	12.57 (+2.49 / -7.98 / -9.39)
Unstructured / 2.5 km	79.08%	0.70 (+0.69 / +0.09 / +0.06)	3.34 (-0.78 / -2.90 / -1.46)
Unstructured Aggressive / 3 km	83.37%	2.80 (-0.80 / +2.68 / -0.05)	16.82 (-13.22 / -10.08 / -2.58)

Figure 11: **Quantitative results for real-world experiments.** The performance improvement by our method is shown in the 2nd column. The average drift reduction is 68.78% with 2 UGVs. Going from structured to unstructured environments, UGV detection becomes increasingly reliable. This is due to numerous false positives in the structured environment caused by objects with similar appearance to the UGV, such as cars, buses, and fire hydrants. In contrast, there are much fewer false positives in the unstructured environment. Furthermore, in unstructured environments, the VIO becomes less reliable due to the lack of static and reliable geometric features. As a result, the performance gain from the UGV bearing measurements is much more significant in unstructured environments. The performance gain in the real world with 2 UGVs is $\sim 10\%$ lower than in the simulation. One major difference we observe is that, in simulation, although we add noise to the bearing measurements, the UGV detection and data association are noise-free. This also shows the importance of accurate and robust UGV detection and data association. Improving these will be the future work of this paper.

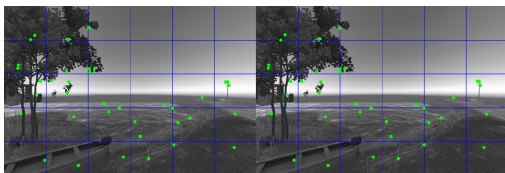


Figure 12: **Simulation stereo images and feature extraction example.** The simulated images are photo-realistic. The plot is generated by the stereo-MSCKF algorithm [7], where features extracted and tracked by the VIO are illustrated by green dots. Some features are extracted from dynamic objects (e.g., tree leaves, grass, and water), leading to additional drifts in VIO.

Quantitative results and analysis: The results of simulation experiments are shown in Fig. 8. We use our autonomous flight software stack [1] in the simulator to accomplish all these flight missions. The trajectory planner is set to have a maximum acceleration of 9.90 m/s^2 ($\sim 1g$) and a maximum velocity of 13 m/s . Such aggressive motions and the simulated noisy visual perception lead to significant VIO drifts, as shown in the last column of Fig. 8. An illustration showing the aggressive motions of one of the simulated experiments is in Fig. 7.

The second column of Fig. 8 shows that by using our method, the VIO drifts are significantly reduced. The drift reduction is consistent across all environments. With 2 UGVs, the drift is reduced by 77.74%. With 3 UGVs, the position drift is reduced by 86.32%.

Such drastic drift reduction results from optimizing the positions of UGVs using our method, and cannot be obtained by merely introducing them into the environment at arbitrary locations. To show this, we randomly position the UGVs within the region enclosed by the UAV’s waypoints. The UAV is commanded to execute the exact same missions, and uses the same factor graph estimation pipeline. The results are shown in the third column of Fig. 8. Our method consistently outperforms this random positioning method by a significant

margin, resulting in 59.43% less drift on average across all experiments.

With larger UAV and UGV teams, the results are shown in Fig. 9. Our algorithm can position the UGVs in a configuration so that the drift of each of the 10 UAVs is drastically reduced. The average drift reduction is 71.50%. Compared to results in Fig. 8, the drift reduction is slightly smaller. This is a result of two factors: First, the average trajectory length of experiments in Fig. 8 is 304% the average trajectory length of experiments in Fig. 9. Since the VIO estimates the relative motion w.r.t. the previous time step, the errors in pose estimates accumulate over time. On the contrary, our method utilizes measurements from UGVs which can provide global correction to the odometry drift. Therefore, the longer the flight trajectory, the bigger drift reduction is expected. Second, even though we have more UGVs in Fig. 9, the UAV to UGV ratio is smaller. This means that on average fewer UGV resources are allocated for each UAV.

This set of simulation experiments demonstrates that our proposed method can consistently help the UAVs minimize their odometry drift, across all environments and flight missions. The corrected drift is an order of magnitude smaller than the raw VIO. Such drastic drift correction is critical when UAVs execute autonomous flight missions at scale.

C. Real World Experiments

Experiment environments: To quantitatively verify the performance of our system, we perform experiments across multiple real-world scenarios, including an unstructured grass-covered area, an open structured area, and a moderately cluttered parking lot. To ensure that the UAV has the desired maneuvers (e.g., aggressive motions, landing at exactly the same position as takeoff), we manually piloted the UAV to fly through the waypoints. An illustration of these environments is shown in Fig. 13.

Quantitative results and analysis: The real world experiment results are shown in Fig. 11. The drift reduction in the unstructured environment is most significant, with an average



Figure 13: **Real world experiment environments:** Structured (left), semi-structured (middle), unstructured (right).

reduction of 81.23%. A counter-intuitive result is that the resulting drift using our method decreases with the increase of the unstructuredness of the environment. This is mainly due to false data association since the structured environment has objects with similar colors as our UGV marker flag colors (buses, fire hydrants, construction barrels, etc.). On the contrary, the unstructured environment provides good contrast in color since the majority of the environment is covered by green grass, as shown in Fig. 13. This can be addressed by using a better object detector and a more robust data association strategy.

To sum up, from multiple long-range UAV flight experiments in environments with a large variance in appearance, clutteredness, and scale, we show by using UGV detections as bearing constraints, the UAV's state estimation is significantly improved. Such improvement is consistent across all scenarios, but most significant when the UGVs are observed more frequently, and data association is more reliable. This indicates that the theoretical and algorithmic aspects of our system are correct and effective, and the system is robust to real-world noise.

VIII. DISCUSSION

Throughout the experiments, we have demonstrated the proposed method's robustness and performance across various environments. Here, we will present its potential applications.

Coverage problem: The missions demonstrated in our experiments all had prescribed setpoints for UAVs. However, since our algorithm's computational complexity is not dominated by the number of UAVs, it can be used in the coverage problem where the robot team must cover and map a given region. The user can specify the environment, and a sampling-based method can be used to determine the positions of UAVs' setpoints Fig. 14. The UGVs will be positioned by the proposed method based on these setpoints to improve the UAVs' localization accuracy.

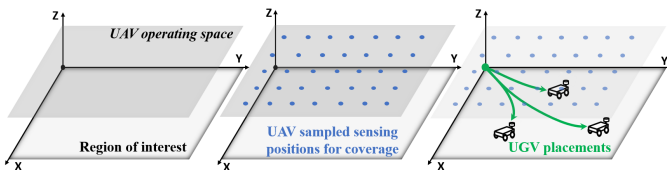


Figure 14: Coverage application example.

Online active collaborative localization: In our current experiments, we assumed that there was no communication during the mission execution. However, when the robots are

allowed to communicate, our method can enable the UGVs to adapt their positions dynamically based on the performance of the collaborative localization algorithm using feedback of the errors in the trajectories of UAVs. Specifically, when this happens, the UAVs can inform the UGVs to actively relocate themselves using the proposed method to minimize the UAVs' localization uncertainties.

Robust perception in perturbed environments: Another important application of our method is to the problems where the environments are susceptible to changes. For example, some landmarks may be destroyed or disappear as time passes (e.g. rocks disappear with the tide, and buildings might get destroyed by earthquakes). It is challenging for UAVs to predict which part of the environment is perturbed. Our method offers UAVs a resort when they are uncertain about their localization.

IX. CONCLUSION

This paper proposes an algorithm for active collaborative localization of a large number of SWAP-constrained robots by optimally positioning (and potentially navigating) a small number of more capable teammates. Specifically, we consider a team of aerial robots with cameras and IMUs that can leverage measurements of ground vehicles equipped with heavier sensors and processors. Numerical results show that our smooth optimization approach for the underlying active robot positioning problem outperforms the greedy algorithm in terms of accuracy, and matches the objective value of a computationally intensive heuristic evolutionary algorithm for global optimization while running in real time. Through experiments in photorealistic simulation environments, the proposed method reduces the UAV odometry drift by 90% with 3 UGVs. It also outperforms randomly positioning UGVs, and shows a robust and significant reduction in drift in real-world experiments. The performance of our system is consistent throughout a range of scenarios that feature large variations in robot motions as well as the amount of structure in the environment. This robust and drastic improvement in state estimation enables long-range autonomous navigation. The proposed active collaborative localization method can be used in various real-world applications in which we encounter environments without texture or features for aerial flight. Future work will address the use of errors in localization as a feedback mechanism to modify trajectories for the \mathcal{C} -agents and allow bidirectional flow of information between \mathcal{C} -agents and \mathcal{L} -agents. We also acknowledge the need to improve the robustness of our method to potential errors in data association and scale up to a bigger team of robots.

REFERENCES

- [1] X. Liu, G. V. Nardari, F. C. Ojeda, Y. Tao, A. Zhou, T. Donnelly, C. Qu, S. W. Chen, R. A. Romero, C. J. Taylor *et al.*, "Large-scale autonomous flight with real-time semantic slam under dense forest canopy," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5512–5519, 2022.
- [2] I. D. Miller, F. Cladera, T. Smith, C. J. Taylor, and V. Kumar, "Stronger together: Air-ground robotic collaboration using semantics," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9643–9650, 2022.

- [3] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "Dtam: Dense tracking and mapping in real-time," in *2011 international conference on computer vision*. IEEE, 2011, pp. 2320–2327.
- [4] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *European conference on computer vision*. Springer, 2014, pp. 834–849.
- [5] J. Delmerico and D. Scaramuzza, "A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 2502–2509.
- [6] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.
- [7] K. Sun, K. Mohta, B. Pfrommer, M. Watterson, S. Liu, Y. Mulgaonkar, C. J. Taylor, and V. Kumar, "Robust stereo visual inertial odometry for fast autonomous flight," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 965–972, 2018.
- [8] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.
- [9] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "Orb-slam3: An accurate open-source library for visual, visual-inertial, and multimap slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.
- [10] K. Mohta, M. Watterson, Y. Mulgaonkar, S. Liu, C. Qu, A. Makineni, K. Saulnier, K. Sun, A. Zhu, J. Delmerico *et al.*, "Fast, autonomous flight in gps-denied and cluttered environments," *Journal of Field Robotics*, vol. 35, no. 1, pp. 101–120, 2018.
- [11] K. Mohta, K. Sun, S. Liu, M. Watterson, B. Pfrommer, J. Svacha, Y. Mulgaonkar, C. J. Taylor, and V. Kumar, "Experiments in fast, autonomous, gps-denied quadrotor flight," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 7832–7839.
- [12] Y. Lin, F. Gao, T. Qin, W. Gao, T. Liu, W. Wu, Z. Yang, and S. Shen, "Autonomous aerial navigation using monocular visual-inertial fusion," *Journal of Field Robotics*, vol. 35, no. 1, pp. 23–51, 2018.
- [13] X. Zhou, X. Wen, Z. Wang, Y. Gao, H. Li, Q. Wang, T. Yang, H. Lu, Y. Cao, C. Xu *et al.*, "Swarm of micro flying robots in the wild," *Science Robotics*, vol. 7, no. 66, p. eabm5954, 2022.
- [14] S. Y. Bao and S. Savarese, "Semantic structure from motion," in *CVPR 2011*. IEEE, 2011, pp. 2025–2032.
- [15] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. Kelly, and A. J. Davison, "Slam++: Simultaneous localisation and mapping at the level of objects," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 1352–1359.
- [16] S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas, "Probabilistic data association for semantic slam," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 1722–1729.
- [17] L. Nicholson, M. Milford, and N. Sünderhauf, "Quadricslam: Dual quadrics from object detections as landmarks in object-oriented slam," *IEEE Robotics and Automation Letters*, vol. 4, no. 1, pp. 1–8, 2018.
- [18] S. Yang and S. Scherer, "Cubeslam: Monocular 3-d object slam," *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 925–938, 2019.
- [19] M. Shan, Q. Feng, and N. Atanasov, "Orcv: object residual constrained visual-inertial odometry," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 5104–5111.
- [20] S. W. Chen, G. V. Nardari, E. S. Lee, C. Qu, X. Liu, R. A. F. Romero, and V. Kumar, "Sloam: Semantic lidar odometry and mapping for forest inventory," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 612–619, 2020.
- [21] R. Bajcsy, "Active perception," *Proceedings of the IEEE*, vol. 76, no. 8, pp. 966–1005, 1988.
- [22] E. J. Sondik, "The optimal control of partially observable markov processes over the infinite horizon: Discounted costs," *Operations Research*, vol. 26, no. 2, pp. 282–304, 1978. [Online]. Available: <http://www.jstor.org/stable/169635>
- [23] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial intelligence*, vol. 101, no. 1-2, pp. 99–134, 1998.
- [24] M. Athans, "The role and use of the stochastic linear-quadratic-gaussian problem in control system design," *IEEE Transactions on Automatic Control*, vol. 16, no. 6, pp. 529–552, 1971.
- [25] C. H. Papadimitriou and J. N. Tsitsiklis, "The complexity of markov decision processes," *Mathematics of Operations Research*, vol. 12, no. 3, pp. 441–450, 1987. [Online]. Available: <http://www.jstor.org/stable/3689975>
- [26] S. Prentice and N. Roy, "The belief roadmap: Efficient planning in belief space by factoring the covariance," *The International Journal of Robotics Research*, vol. 28, no. 11-12, pp. 1448–1465, 2009.
- [27] A. Bry and N. Roy, "Rapidly-exploring random belief trees for motion planning under uncertainty," in *2011 IEEE international conference on robotics and automation*. IEEE, 2011, pp. 723–730.
- [28] J. Van Den Berg, P. Abbeel, and K. Goldberg, "Lqg-mp: Optimized path planning for robots with motion uncertainty and imperfect state information," *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 895–913, 2011.
- [29] J. Van Den Berg, S. Patil, and R. Alterovitz, "Motion planning under uncertainty using iterative local optimization in belief space," *The International Journal of Robotics Research*, vol. 31, no. 11, pp. 1263–1278, 2012.
- [30] B. Chaffrow, G. Kahn, S. Patil, S. Liu, K. Goldberg, P. Abbeel, N. Michael, and V. Kumar, "Information-theoretic planning with trajectory optimization for dense 3d mapping," in *Robotics: Science and Systems*, vol. 11. Rome, 2015, pp. 3–12.
- [31] N. Atanasov, J. Le Ny, K. Daniilidis, and G. J. Pappas, "Decentralized active information acquisition: Theory and application to multi-robot slam," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 4775–4782.
- [32] K. Sun, B. Schlotfeldt, G. J. Pappas, and V. Kumar, "Stochastic motion planning under partial observability for mobile robots with continuous range measurements," *IEEE Transactions on Robotics*, vol. 37, no. 3, pp. 979–995, 2020.
- [33] D. Falanga, P. Foehn, P. Lu, and D. Scaramuzza, "Pampc: Perception-aware model predictive control for quadrotors," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–8.
- [34] Z. Zhang and D. Scaramuzza, "Fisher information field: an efficient and differentiable map for perception-aware planning," *arXiv preprint arXiv:2008.03324*, 2020.
- [35] J. Tordesillas and J. P. How, "Panther: Perception-aware trajectory planner in dynamic environments," *IEEE Access*, vol. 10, pp. 22662–22677, 2022.
- [36] I. Spasojevic, V. Murali, and S. Karaman, "Perception-aware time optimal path parameterization for quadrotors," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 3213–3219.
- [37] V. Murali, I. Spasojevic, W. Guerra, and S. Karaman, "Perception-aware trajectory generation for aggressive quadrotor flight using differential flatness," in *2019 American Control Conference (ACC)*, 2019, pp. 3936–3943.
- [38] A. Martinelli, F. Pont, and R. Siegwart, "Multi-robot localization using relative observations," in *Proceedings of the 2005 IEEE international conference on robotics and automation*. IEEE, 2005, pp. 2797–2802.
- [39] A. Franchi, G. Oriolo, and P. Stegagno, "Mutual localization in a multi-robot system with anonymous relative position measures," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 3974–3980.
- [40] A. Prorok and A. Martinoli, "Accurate indoor localization with ultra-wideband using spatial models and collaboration," *The International Journal of Robotics Research*, vol. 33, no. 4, pp. 547–568, 2014.
- [41] T. Nguyen, K. Mohta, C. J. Taylor, and V. Kumar, "Vision-based multi-mav localization with anonymous relative measurements using coupled probabilistic data association filter," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 3349–3355.
- [42] Y. Wang, X. Wen, L. Yin, C. Xu, Y. Cao, and F. Gao, "Certifiably optimal mutual localization with anonymous bearing measurements," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9374–9381, 2022.
- [43] Y. Tian, K. Khosoussi, D. M. Rosen, and J. P. How, "Distributed certifiably correct pose-graph optimization," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 2137–2156, 2021.
- [44] Y. Tian, A. S. Bedi, A. Koppel, M. Calvo-Fullana, D. M. Rosen, and J. P. How, "Distributed riemannian optimization with lazy communication for collaborative geometric estimation," in *2022 IEEE/RSJ International*

Conference on Intelligent Robots and Systems (IROS), 2022, pp. 4391–4398.

- [45] M. Kontitsis, E. A. Theodorou, and E. Todorov, “Multi-robot active slam with relative entropy optimization,” in *2013 American Control Conference*. IEEE, 2013, pp. 2757–2764.
- [46] Y. Chen, L. Zhao, K. M. B. Lee, C. Yoo, S. Huang, and R. Fitch, “Broadcast your weaknesses: cooperative active pose-graph slam for multiple robots,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2200–2207, 2020.
- [47] R. Kurazume, S. Nagata, and S. Hirose, “Cooperative positioning with multiple robots,” in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*. IEEE, 1994, pp. 1250–1257.
- [48] N. Trawny and T. Barfoot, “Optimized motion strategies for cooperative localization of mobile robots,” in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA’04. 2004*, vol. 1. IEEE, 2004, pp. 1027–1032.
- [49] S. Tully, G. Kantor, and H. Choset, “Leap-frog path design for multi-robot cooperative localization,” in *Field and Service Robotics: Results of the 7th International Conference*. Springer, 2010, pp. 307–317.
- [50] J. R. Spletzer and C. J. Taylor, “Dynamic sensor planning and control for optimally tracking targets,” *The International Journal of Robotics Research*, vol. 22, no. 1, pp. 7–20, 2003.
- [51] L. Zhou and V. Kumar, “Robust multi-robot active target tracking against sensing and communication attacks,” *IEEE Transactions on Robotics*, pp. 1–13, 2023.
- [52] L. Zhou, V. Tzoumas, G. J. Pappas, and P. Tokekar, “Resilient active target tracking with multiple robots,” *IEEE Robotics and Automation Letters*, vol. 4, no. 1, pp. 129–136, 2019.
- [53] U. Feige, “A threshold of $\ln n$ for approximating set cover,” *Journal of the ACM (JACM)*, vol. 45, no. 4, pp. 634–652, 1998.
- [54] A. Olshevsky, “Minimal controllability problems,” *IEEE Transactions on Control of Network Systems*, vol. 1, no. 3, pp. 249–258, 2014.
- [55] A. Krause, A. Singh, and C. Guestrin, “Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies,” *Journal of Machine Learning Research*, vol. 9, no. 2, 2008.
- [56] S. T. Jawaid and S. L. Smith, “Submodularity and greedy algorithms in sensor scheduling for linear dynamical systems,” *Automatica*, vol. 61, pp. 282–288, 2015.
- [57] L. Carlone and S. Karaman, “Attention and anticipation in fast visual-inertial navigation,” *IEEE Transactions on Robotics*, vol. 35, no. 1, pp. 1–20, 2018.
- [58] V. Tzoumas, L. Carlone, G. J. Pappas, and A. Jadbabaie, “LQG control and sensing co-design,” *IEEE Transactions on Automatic Control*, vol. 66, no. 4, pp. 1468–1483, 2020.
- [59] L. F. Chamon, G. J. Pappas, and A. Ribeiro, “Approximate supermodularity of Kalman filter sensor selection,” *IEEE Transactions on Automatic Control*, vol. 66, no. 1, pp. 49–63, 2020.
- [60] T. Runarsson and X. Yao, “Search biases in constrained evolutionary optimization,” *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 35, pp. 233 – 243, 06 2005.
- [61] A. Wächter and L. T. Biegler, “On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming,” *Mathematical programming*, vol. 106, pp. 25–57, 2006.
- [62] S. G. Johnson. The NLOpt nonlinear-optimization package. [Online]. Available: <http://github.com/stevengj/nlopt>
- [63] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, “isam2: Incremental smoothing and mapping using the bayes tree,” *The International Journal of Robotics Research*, vol. 31, no. 2, pp. 216–235, 2012.
- [64] F. Dellaert, “Factor graphs and gtsam: A hands-on introduction,” Georgia Institute of Technology, Tech. Rep., 2012.
- [65] C. Bai, T. Xiao, Y. Chen, H. Wang, F. Zhang, and X. Gao, “Faster-lio: Lightweight tightly coupled lidar-inertial odometry using parallel sparse incremental voxels,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4861–4868, 2022.
- [66] H. Weyl, “Das asymptotische verteilungsgesetz der eigenwerte linearer partieller differentialgleichungen (mit einer anwendung auf die theorie der hohlraumstrahlung),” *Mathematische Annalen*, vol. 71, no. 4, pp. 441–479, 1912.

X. APPENDIX

We split the proof of Theorem V.1 into two claims. Roughly speaking, our reasoning hinges on the fact that the trace

of the covariance matrix is just the sum of inverses of the eigenvalues of the information matrix. The rest of the proof involves analyzing how the sum of inverses of a sequence of positive real numbers behaves under suitable perturbations. This is precisely the content of the following

Claim 1. *Let $\delta, \eta > 0$ be a fixed pair of real numbers, and define*

$$s_0 = \frac{\eta}{\eta + 1} \delta^{-1}. \quad (21)$$

For any pair of sequences $(\lambda_k)_{k=1}^d, (\tilde{\lambda}_k)_{k=1}^d \in (0, \infty)^d$ such that $|\tilde{\lambda}_k - \lambda_k| \leq \delta \forall 1 \leq k \leq d$, we have the implication

$$\sum_{k=1}^d \frac{1}{\lambda_k} = s \leq s_0 \quad \Rightarrow \quad \frac{s}{1 + \eta} \leq \sum_{k=1}^d \frac{1}{\tilde{\lambda}_k} \leq s(1 + \eta). \quad (22)$$

Proof. To prove the inequality on the right, it suffices to show

$$\sum_{k=1}^d \frac{1}{\lambda_k} = s \leq s_0 \quad \Rightarrow \quad \sum_{k=1}^d \frac{1}{\lambda_k - \delta} \leq s(1 + \eta). \quad (23)$$

Indeed, since $\lambda_k > 0 \forall k$, we have

$$\frac{1}{\lambda_k} \leq s \quad \Rightarrow \quad \lambda_k \geq s^{-1} \geq s_0^{-1} = \frac{1 + \eta}{\eta} \delta > \delta \quad \forall k, \quad (24)$$

showing that the expression on the right of Equation 23 is well-defined. Now, consider the following optimization problem

$$\begin{aligned} & \sup_{(\lambda_k)_{k=0}^d \in (0, \infty)^d} \sum_{k=1}^d \frac{1}{\lambda_k - \delta} \\ & \text{s.t.} \quad \sum_{k=1}^d \frac{1}{\lambda_k} = s. \end{aligned} \quad (25)$$

To show that it is bounded above by $(1 + \eta)s$, upon introducing variables $a_k := \lambda_k^{-1}$, we reformulate it as follows

$$\begin{aligned} & \max_{(a_k)_{k=0}^d \in [0, s]^d} \sum_{k=1}^d \frac{1}{a_k^{-1} - \delta} \\ & \text{s.t.} \quad \sum_{k=1}^d a_k = s. \end{aligned} \quad (26)$$

We note that the function $f(x) = \frac{1}{x^{-1} - \delta}$ is convex on $x \in [0, s] \subseteq [0, \delta^{-1}]$ due to $f''(x) = \frac{2\delta}{(1 - x\delta)^3} > 0$ on the the mentioned interval. As a result, Problem 26 involves maximizing a convex function subject to a simplex constraint. Hence, an optimum is attained at one of the extreme vertices of the feasible region, which all take the form $(s, 0, \dots, 0)$ up to a permutation of indices. In any case, all such extreme points have the objective value

$$\frac{1}{s^{-1} - \delta} = \frac{s}{1 - s\delta} \leq \frac{s}{1 - s_0\delta} = (1 + \eta)s, \quad (27)$$

thus proving the desired inequality.

Similarly, to prove the inequality on the left, it suffices to show

$$\sum_{k=1}^d \frac{1}{\lambda_k} = s \leq s_0 \quad \Rightarrow \quad \sum_{k=1}^d \frac{1}{\lambda_k + \delta} \geq \frac{s}{1 + \eta}. \quad (28)$$

Now, consider the following optimization problem

$$\begin{aligned} \inf_{(\lambda_k)_{k=0}^d \in (0, \infty)^d} \sum_{k=1}^d \frac{1}{\lambda_k + \delta} \\ \text{s.t. } \sum_{k=1}^d \frac{1}{\lambda_k} = s. \end{aligned} \quad (29)$$

To show that it is bounded below by $\frac{s}{1+\eta}$, upon introducing variables $a_k := \lambda_k^{-1}$, we reformulate it as follows

$$\begin{aligned} \min_{(a_k)_{k=0}^d \in [0, s]^d} \sum_{k=1}^d \frac{1}{a_k^{-1} + \delta} \\ \text{s.t. } \sum_{k=1}^d a_k = s. \end{aligned} \quad (30)$$

We note that the function $f(x) = \frac{1}{x^{-1} + \delta}$ is concave on $x \in [0, s] \subseteq [0, \delta^{-1}]$ due to $f''(x) = \frac{-2\delta}{(1+x\delta)^3} < 0$. As a result, Problem 30 involves minimizing a concave function subject to a simplex constraint. Hence, an optimum is attained at one of the extreme vertices of the feasible region, which all take the form $(s, 0, \dots, 0)$ up to a permutation of indices. In any case, all such extreme points have the objective value

$$\frac{1}{s^{-1} + \delta} = \frac{s}{1+s\delta} \geq \frac{s}{1+s_0\delta} = \frac{1+\eta}{1+2\eta} s \geq \frac{s}{1+\eta}, \quad (31)$$

thus proving the desired inequality. \square

Claim 2. Consider an arbitrary $\delta > 0$, $\mathbf{x} \in \mathbb{R}^3$, and suppose

$$\frac{\delta}{\|\mathbf{x} - \mathbf{z}_j\|_2} \leq \zeta^{-1} \quad \forall 1 \leq j \leq M, \quad (32)$$

for some $\zeta > 0$. Then,

$$\|\mathcal{J}_\delta(\mathbf{x}; \mathbf{z}_{1:M}) - \mathcal{J}_0(\mathbf{x}; \mathbf{z}_{1:M})\|_2 \leq \frac{M \sigma_m^{-2}}{\min_{j \leq M} \|\mathbf{x} - \mathbf{z}_j\|_2^2} \frac{1}{(1+\zeta^2)}. \quad (33)$$

Proof. Letting $\Delta_j = \|\mathbf{z}_j - \mathbf{x}\|_2 \forall 1 \leq j \leq M$, we have

$$\begin{aligned} S^\delta(\mathbf{x}; \mathbf{z}_j) &:= \frac{\partial \mathbf{h}_\delta^T}{\partial \mathbf{x}}(\mathbf{x}, \mathbf{z}_j) \frac{\partial \mathbf{h}_\delta}{\partial \mathbf{x}}(\mathbf{x}, \mathbf{z}_j) \\ &= \frac{1}{\Delta_j^2 + \delta^2} \left(I_3 - \frac{\Delta_j^2 + 2\delta^2}{(\Delta_j^2 + \delta^2)^2} (\mathbf{z}_j - \mathbf{x})(\mathbf{z}_j - \mathbf{x})^T \right). \end{aligned} \quad (34)$$

By noting that the family of symmetric matrices $(S^\delta(\mathbf{x}; \mathbf{z}_j))_{\delta \in \mathbb{R}}$ may be jointly diagonalized with respect to an orthonormal basis containing $\frac{\mathbf{z}_j - \mathbf{x}}{\Delta_j}$, and noting that the spectral norm of a symmetric matrix is just the maximum absolute value of its eigenvalues, we get

$$\|S^\delta(\mathbf{x}; \mathbf{z}_j) - S^0(\mathbf{x}; \mathbf{z}_j)\|_2 \leq \Delta_j^{-2} \frac{1}{1+\zeta^2}. \quad (35)$$

By the triangle inequality, we have

$$\begin{aligned} \frac{\|\mathcal{J}_\delta(\mathbf{x}; \mathbf{z}_{1:M}) - \mathcal{J}_0(\mathbf{x}; \mathbf{z}_{1:M})\|_2}{\sigma_m^{-2}} &= \left\| \sum_{j=1}^M (S^\delta(\mathbf{x}; \mathbf{z}_j) - S^0(\mathbf{x}; \mathbf{z}_j)) \right\|_2 \\ &\leq \sum_{j=1}^M \|(S^\delta(\mathbf{x}; \mathbf{z}_j) - S^0(\mathbf{x}; \mathbf{z}_j))\|_2 \\ &\leq \sum_{j=1}^M \Delta_j^{-2} \frac{1}{1+\zeta^2} \\ &\leq \frac{M}{\min_{j \leq M} \Delta_j^2} \frac{1}{1+\zeta^2}, \end{aligned} \quad (36)$$

as desired. \square

Now we leverage Claims 1 and 2 to provide a proof of Theorem V.1.

Proof. (of Theorem V.1) Since the prior covariance is positive definite, $\mathcal{J}_\delta > 0$ for all δ . Next, denote the eigenvalues of \mathcal{J}_δ in decreasing order by $\lambda_1^\delta \geq \lambda_2^\delta \geq \lambda_3^\delta > 0$. It then follows that

$$\text{tr}((\mathcal{J}_\delta)^{-1}) = \sum_{j=1}^3 \frac{1}{\lambda_j^\delta}. \quad (37)$$

By Weyl's inequality [66], we have

$$|\lambda_j^\delta - \lambda_j^0| \leq \|\mathcal{J}_\delta - \mathcal{J}_0\|_2 \leq \frac{M \sigma_m^{-2}}{R_{\min}^2} \frac{1}{(1+\zeta^2)} \quad \forall j \leq 3, \quad (38)$$

where the latter inequality follows from Claim 2. As a result, making substitutions

$$\lambda_j \rightarrow \lambda_j^\delta, \quad \tilde{\lambda}_j \rightarrow \tilde{\lambda}_j^\delta, \quad \delta \rightarrow \frac{M \sigma_m^{-2}}{R_{\min}^2} \frac{1}{(1+\zeta^2)} \quad (39)$$

in Claim 1, we get that for s_0 given by Equation 12, the theorem holds. \square

Finally we turn to proving Corollary 1 with the help of an additional claim.

Claim 3. Denote the value of Problem 2 at a given $\mathbf{z}_{1:M}$ for a given δ by V^δ . Within the same setup as for Theorem V.1, we have

$$V^\delta = s \leq s_0 \Rightarrow \frac{V^\delta}{1+\eta} \leq V^0 \leq V^\delta(1+\eta). \quad (40)$$

By symmetry the same implication holds with the roles of V^δ and V^0 reversed.

Proof. To prove the right-hand inequality, note that by the definition of V^δ , we have $\forall i \leq N$

$$\text{tr}(\mathcal{J}_\delta(\mathbf{x}_i; \mathbf{z}_{1:M})^{-1}) \leq V_\delta = s \leq s_0. \quad (41)$$

Hence, by Claim 1, we have $\forall i \leq N$

$$\text{tr}(\mathcal{J}_0(\mathbf{x}_i; \mathbf{z}_{1:M})^{-1}) \leq \text{tr}(\mathcal{J}_\delta(\mathbf{x}_i; \mathbf{z}_{1:M})^{-1})(1+\eta) \leq (1+\eta)V^\delta, \quad (42)$$

and by taking the maximum of the latter inequality over $i \leq N$, the inequality follows. To prove the left-hand inequality, again by definition, there exists an $i \leq N$ such that

$$\text{tr}(\mathcal{J}_\delta(\mathbf{x}_i; \mathbf{z}_{1:M})^{-1}) = V^\delta = s \leq s_0. \quad (43)$$

Hence, by Claim 1, we have

$$V^0 \geq \text{tr}(\mathcal{J}_0(\mathbf{x}_i; \mathbf{z}_{1:M})^{-1}) \geq \frac{\text{tr}(\mathcal{J}_\delta(\mathbf{x}_i; \mathbf{z}_{1:M})^{-1})}{1 + \eta} = \frac{V^\delta}{1 + \eta}, \quad (44)$$

as desired. \square

Proof. (of Corollary 1) Let \mathbf{z}_*^δ and \mathbf{z}_*^0 be optimal solutions for Problem 2 and Problem 1, respectively. By assumption, we have

$$\mathbf{z}_*^\delta \in \mathcal{S} \Rightarrow V^\delta(\mathbf{z}_*^\delta) \leq s_0. \quad (45)$$

By Claim 1, we then have

$$V^0(\mathbf{z}_*^\delta) \leq (1 + \eta)V^\delta(\mathbf{z}_*^\delta) \leq (1 + \eta)s_0. \quad (46)$$

We now distinguish two cases. First, if $V^0(\mathbf{z}_*^0) > s_0$, the inequality trivially holds. Otherwise, $V^0(\mathbf{z}_*^0) \leq s_0$, and by the previous claim, we have

$$V^0(\mathbf{z}_*^0) \geq \frac{V^\delta(\mathbf{z}_*^0)}{1 + \eta} \geq \frac{V^\delta(\mathbf{z}_*^\delta)}{1 + \eta} \geq \frac{V^0(\mathbf{z}_*^\delta)}{(1 + \eta)^2}, \quad (47)$$

and so the result follows. \square