

Communication-Aware Iterative Map Compression for Online Path-Planning

Evangelos Psomias, Ali Reza Pedram, Dipankar Maity, Panagiotis Tsiotras

Abstract—This paper addresses the problem of optimizing communicated information among heterogeneous, resource-aware robot teams to facilitate their navigation. In such operations, a mobile robot compresses its local map to assist another robot in reaching a target within an uncharted environment. The primary challenge lies in ensuring that the map compression step balances network load while transmitting only the most essential information for effective navigation. We propose a communication framework that sequentially selects the optimal map compression in a task-driven, communication-aware manner. It introduces a decoder capable of iterative map estimation, handling noise through Kalman filter techniques. The computational speed of our decoder allows for a larger compression template set compared to previous methods, and enables applications in more challenging environments. Specifically, our simulations demonstrate a remarkable 98% reduction in communicated information, compared to a framework that transmits the raw data, on a large Mars inclination map and an Earth map, all while maintaining similar planning costs. Furthermore, our method significantly reduces computational time compared to the state-of-the-art approach.

I. INTRODUCTION

Situational awareness is a key element for the success of collaborative multi-robot missions ranging from search and rescue operations on Earth [1] to extraterrestrial space missions [2]. These missions involve robots collaboratively exploring an unknown region while exchanging *relevant* information with their teammates to improve team knowledge, thus developing global situational awareness.

Multi-robot collaboration begets several challenges, including map fusion [3] and intermittent communication [4]. Existing approaches, such as [5] and [6], address map fusion by merging local semantic and geometric maps, respectively, into a unified global representation. However, in real-world scenarios, communication is not always guaranteed. Recent works, such as [4], leverage communication loss as a signal to refine a robot’s belief about its teammate’s state and infer information about unvisited areas. Similar ideas are also explored in the controls community [7] proving that *no communication can serve as a form of (implicit) communication*.

In multi-robot navigation, path-planning solutions must account for both task requirements and communication constraints. Existing approaches often treat communication

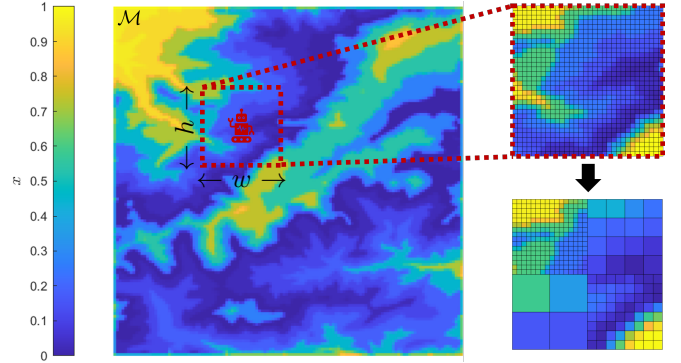


Fig. 1: Earth traversability map $\mathcal{M}$. The mobile robot perceives a local map of dimensions $w \times h$ as it traverses the map and compresses it based on an abstraction template.

as an afterthought [8], reducing their overall effectiveness. Thus, it is crucial to develop task-relevant, resource-aware compression schemes that enable the transmission of critical information in a timely manner and at an optimal resolution. Determining *which* information to send, *when* to send, and at *what* resolution are the key decision-making factors.

The coupled decision-making and communication problem has also been addressed in the controls community [9]–[12]. However, developing the optimal compression/quantization scheme still remains intractable, even for linear systems [13]. Recently, there have been studies on how to choose the *optimal* quantizer from a given set in a task- and communication-aware manner [14]–[16], alleviating the intractability of the quantizer design problem. In this work, we follow the same principle, where a set of quantizers is available to the robots to compress their perception data before transmitting it to another teammate.

Related Work: Several researchers have explored methods for determining what information to communicate based on a given task. In [17], the authors introduce a communication-aware framework that adjusts data transmission based on communication conditions in centralized multi-agent operations using 3D point cloud maps. They develop a control function for 5G-enabled multi-robot systems that regulates the transmission rate according to the network’s Key Performance Indicators (KPIs). In [18], the authors propose OCBC, an algorithm that solves a bandit-based combinatorial optimization problem to evaluate an agent’s observations. However, considering the combinatorial nature of the solution, this approach may become computationally intractable with increased robot’s field of view. In [19], the authors present an architecture consisting of a Con-

The work was supported by the ARL grant DCIST CRA W911NF-17-2-0181.

E. Psomias, A. R. Pedram, and P. Tsiotras are with the D. Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, 30332-0150, USA. Email: {epsomias3, apedram3, tsiotras}@gatech.edu

D. Maity is with the Department of Electrical and Computer Engineering, University of North Carolina at Charlotte, NC, 28223-0001, USA. Email: dmaity@charlotte.edu

$\mathcal{A}^{\theta_t} = \mathcal{A}^{\theta_t}(\mathbf{p}_t^S)$ and attempts to reconstruct x from o^{θ_t} .

We compute the total number of bits n^θ required to transmit an abstracted environment as the sum of bits needed to send the measurements $o_i^{\theta_t}$, denoted by n_m , and the bits required to transmit an abstraction index θ , denoted by n_a

$$n^\theta = k^\theta n_m + n_a. \quad (1)$$

III. PROBLEM FORMULATION

Consider a pair of mobile robots, an Actor and a Sensor, that move through an unfamiliar environment $\mathcal{M} \subset \mathbb{R}^2$ with varying traversability conditions. As they navigate the environment, each robot uses its sensing equipment to observe a portion of the environment (local map). The Actor is tasked with reaching a specified target while minimizing a certain cost (e.g., path length or time) via employing an online path-planning algorithm. Meanwhile, the Sensor aims to assist the Actor in achieving its objective by transmitting informative abstractions of its local map to the Actor at each timestep t . We assume that the Sensor's path is known a priori, and it is an aerial vehicle (i.e., the cell values do not affect its traversability). The design of the Sensor's path is not the subject of this work and interested readers may refer to [26].

The positions of the Actor and Sensor at timestep t are denoted by $\mathbf{p}_t^A, \mathbf{p}_t^S \in \mathcal{M}$, respectively. Let $u_t \in U$ represent the Actor's control action at timestep t , chosen from a finite set of control actions U .

A. Problem Statement

In this work, we design a framework where the Sensor facilitates the Actor to reach its target with minimum cost. The Sensor selects the optimal abstraction θ^* in an online manner, from a given set of abstractions Θ , to compress its local map in real-time. The Actor computes its control action u_t based on both its own measurements and the abstractions transmitted by the Sensor at timestep t . The inputs of the framework are the initial positions of the Actor and the Sensor, $\mathbf{p}_0^A$ and $\mathbf{p}_0^S$, along with the Sensor's predefined path, the Actor's (constant or moving) target $\mathbf{p}_G^A(t)$, the set of abstractions Θ , the communication channel noise v^S , and the Actor's perception noise v^A . The proposed framework is shown in Figure 3, and the roles of its components are explained in Section IV.

IV. FRAMEWORK ARCHITECTURE

In this section, we present the proposed framework, as illustrated in Figure 3. The framework incorporates a Path Planner, Path Converter, and Encoder similar to those described in [20]. While their functionality is briefly outlined here, any differences from [20] are explicitly highlighted.

A. Decoder

The robot's decoder performs the estimation of the map. Although both robots must use the same estimation strategy for compatibility, each one performs the estimation individually: the Sensor relies solely on its own abstractions, while the Actor uses both its own measurements and the communicated abstractions. As explained in Section II, both

the abstractions and the Actor's measurements can be represented by a set of linear equations $\mathcal{A}_t x + v = o_t$.

There are various techniques to estimate x . In [20], we introduced a decoder that provides an estimate by minimizing the variance of the estimation error, without considering noise in the measurements. Specifically, the map estimate was the solution of

$$\min_{x \in \mathbb{R}^N} \|x - \hat{x}_0\|^2, \quad (2a)$$

$$\mathcal{A}_{0:t} x = o_{0:t}, \quad (2b)$$

$$0 \leq x_j \leq 1, \quad j = 1, \dots, N, \quad (2c)$$

where $\hat{x}_0$ was the initial belief on the cell values, and $\mathcal{A}_{0:t} = \{\mathcal{A}_0, \mathcal{A}_1, \dots, \mathcal{A}_t\}$ and $o_{0:t} = \{o_0, o_1, \dots, o_t\}$ were the history of abstractions and measurements.

The above problem is a quadratic optimization problem subject to linear constraints. State-of-the-art solvers, such as MOSEK 10.1, utilize interior-point methods to address this type of problem. Generally, interior-point methods exhibit a computational complexity $O(\sqrt{k}N^3)$ [27], where k represents the number of constraints and N denotes the number of variables. Consequently, solving this optimization problem at each time step t can become a bottleneck over long horizons, particularly when managing a large history of abstractions.

By relaxing the constraint (2c), we can provide an estimation with complexity $O(N^3)$. It is proven that the minimum variance estimate is equal to the conditional mean [28]. Additionally, we can analytically compute the conditional mean $\hat{x}_t$ and covariance Σ_t of the vector containing the values of the map x , given measurements or abstractions $o_t = \mathcal{A}_t x + v$, using Kalman Filter,

$$\hat{x}_t' = \hat{x}_{t-1}' + K_t(o_t - \mathcal{A}_t \hat{x}_{t-1}'), \quad (3a)$$

$$\Sigma_t = (I - K_t \mathcal{A}_t) \Sigma_{t-1}, \quad (3b)$$

$$K_t = \Sigma_{t-1} \mathcal{A}_t^\top (\mathcal{A}_t \Sigma_{t-1} \mathcal{A}_t^\top + V)^{-1}, \quad (3c)$$

where $v = v^A \sim \mathcal{N}(0, V^A)$ for the Actor's perception noise, and $v = v^S \sim \mathcal{N}(0, V^S)$ for the channel noise. To ensure that the estimate remains between 0 and 1, we use the estimate projection approach with inequality state constraints [29]. Hence, the projected estimate is given by:

$$\hat{x}_t = \max(\min(\hat{x}_t', \mathbf{1}), \mathbf{0}), \quad (4)$$

where $\mathbf{1}$ is a vector of all ones, $\mathbf{0}$ is a vector of all zeros, and the operations $\max$ and $\min$ are performed element-wise.

It should be noted that $\hat{x}_t$ does not yield the minimum variance estimate, since this would require the numerical solution of the optimization problem in (2), increasing the computational time of the algorithm. However, the simulations in Section V-D indicate that this relaxation does not impact the final result.

The decoder's algorithm is presented in Algorithm 1. With a slight abuse of notation, we define as $x_0 \sim \mathcal{N}(\hat{x}_0, \Sigma_0)$ the initial Gaussian belief of the map, where $\hat{x}_0$ is the mean and Σ_0 is the covariance matrix. The set $L_t^A = L_t^A(\mathbf{p}_t^A)$ contains the cells of the Actor's current local map at timestep t . Line 1 in the algorithm employs the function MEASURE to sense

Algorithm 1 The Actor's Decoder Algorithm

Input: $v^A, v^S, \hat{x}'_{t-1}, \Sigma_{t-1}, \theta_t, o^{\theta_t}, \mathbf{p}_t^A, \mathbf{p}_t^S$

Output: $\hat{x}_t, \hat{x}'_t, \Sigma_t$

- 1: $(\mathcal{A}_t^A, o_t^A) \leftarrow \text{MEASURE}(L_t^A)$
 - 2: $\hat{x}_t^-, \Sigma_t^- \leftarrow \text{DECODER}(\mathcal{A}_t^A, o_t^A, \hat{x}'_{t-1}, \Sigma_{t-1}, v^A)$
 - 3: $\hat{x}'_t, \Sigma_t \leftarrow \text{DECODER}(\mathcal{A}_t^{\theta_t}, o_t^{\theta_t}, \hat{x}_t^-, \Sigma_t^-, v^S)$
 - 4: $\hat{x}_t \leftarrow \text{PROJECTION}(\hat{x}'_t)$ using (4)
 - 5: **return** $\hat{x}_t, \hat{x}'_t, \Sigma_t$
-

the values of the cells within the Actor's local map. Lines 2-3 perform the estimation process using (3) for the Actor's measurements and the Sensor's abstractions, respectively. In case the matrix $\mathcal{A}_t^A$ or $\mathcal{A}_t^{\theta_t}$ exhibits singularities that prevent inversion, we use robust scheme by adding a small amount of artificial noise, $V = \varepsilon I$, to the measurements to mitigate these issues. Finally, Line 4 uses the function PROJECTION to project the measurements, so that their values are in $[0, 1]$.

The Sensor employs the same estimation algorithm, but omits Lines 1-2 if there is no overlap between the cells it has observed and those the Actor has explored up to timestep t . When an overlap occurs, the Sensor executes the DECODER using the overlapping cells, since it knows v^A . This approach ensures that the Sensor considers the Actor's knowledge of the overlapping cells for the abstraction selection process. Consequently, if the Actor has already observed a region, the Sensor's map estimate incorporates this information.

B. Path Planner

The Path Planner employs a graph search algorithm (e.g., Dijkstra's algorithm [30]), to compute the Actor's path based on the mean value of the estimated map $\hat{x}_t$. To achieve this, the Path Planner constructs a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ corresponding to the discretized map $\mathcal{M}$, where $\mathcal{N}$ denotes the set of vertices and $\mathcal{E}$ represents the set of edges. Each vertex in $\mathcal{G}$ corresponds to the center of a cell in $\mathcal{M}$. We will use $\mathbf{p}$ to denote both the vertex and the cell. Two vertices are connected by an edge, if the Actor can move between them using a control action $u \in U$. The cost of traversing a vertex is given by [20], [31]

$$c_t(\mathbf{p}) = \begin{cases} \hat{x}_{\mathbf{p},t} + a, & \text{if } \mathbf{p} \in P_\epsilon, \\ N(\epsilon + a), & \text{otherwise,} \end{cases} \quad (5)$$

where $\hat{x}_{\mathbf{p},t} \in [0, 1]$ is the estimated cell value at position $\mathbf{p}$, a is a constant penalty for movement, and $P_\epsilon = \{\mathbf{p} \in \mathcal{N} : \hat{x}_{\mathbf{p},t} \leq \epsilon\}$ is the set of cells meeting a feasibility condition, with the threshold ϵ satisfying $\hat{x}_{\mathbf{p},0} < \epsilon \leq 1, \forall \mathbf{p} \in \mathcal{N}$. We require ϵ to be greater than the initial belief, ensuring that when the Actor must choose between exploring an unknown region or traversing known obstacles, it will select the former. The intuition behind this cost is based on the assumption that the traversal time of a cell is proportional to its value (difficulty of traversal) and a constant penalty for movement.

Therefore, the Actor's computed path is given by

$$\pi_t^* = \arg \min_{\pi \in \Pi_t} \sum_{\mathbf{p} \in \pi} c_t(\mathbf{p}), \quad (6)$$

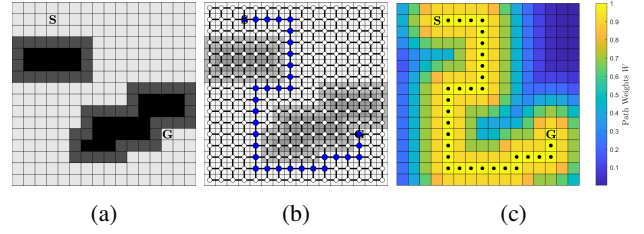


Fig. 4: (a) Discretized environment with static obstacles. **S** denotes the starting cell while **G** denotes the target; (b) Associated graph of the environment employing the set of actions $U = \{\text{UP, DOWN, LEFT, RIGHT}\}$, along with the optimal path (blue nodes) with cost function (5); (c) Path weights computed using (7).

where Π_t is the set of all paths with the first element being the Actor's current position $\mathbf{p}_t^A$, and the last element being its goal location $\mathbf{p}_G^A(t)$. An example of a grid with its associated graph and computed path is presented in Figure 4a and 4b.

C. Path Converter

The Sensor's Path Converter identifies a region of interest around the Actor's current path to guide the abstraction selection process. After receiving the Actor's current path π_t^* , the Path Converter assigns weights to each cell in $\mathcal{M}$ based on its proximity to the path. The weights are given by

$$w_t(\mathbf{p}) = \max_{\mathbf{p}^* \in \pi_t^*} e^{-\frac{\|\mathbf{p} - \mathbf{p}^*\|^2}{2\sigma^2}}, \quad \mathbf{p} \in \mathcal{M}, \quad (7)$$

where σ is a scalar characterizing the width around the path. Figure 4(c) presents an example of path weights.

D. Encoder

The Sensor's Encoder selects the optimal abstraction to communicate to the Actor. Let $\theta_t \in \Theta$ represent the Sensor's abstraction at timestep t , where Θ denotes the set of available abstractions. Unlike the Encoder in [20], this Encoder does not rely on the history of abstractions to compute the next abstraction, as all the necessary past information is contained in $\hat{x}_t$. Hence, the optimal abstraction at time t is determined through an exhaustive search using the following function

$$J_t(\theta_t) = \|W_t \circ (\tilde{x}_t - \hat{x}_t(\theta_t))\|^2 + \lambda(\theta_t), \quad (8a)$$

$$\theta_t^* = \arg \min_{\theta_t \in \Theta} J_t(\theta_t), \quad (8b)$$

where $W_t \in \mathbb{R}^N$ is the vector containing the weights $w_t(\mathbf{p})$ of every cell, $\tilde{x}_t \in [0, 1]^N$ is the Sensor's sensed cell values until time t , $\hat{x}_t(\theta_t) \in [0, 1]^N$ is the estimation vector computed from (3), and $\lambda(\theta_t)$ is the communication cost (related to n^θ in (1)). Note that in our case, the Sensor's perception noise is assumed to be negligible, so $\tilde{x}_t$ contains the true values of the cells the Sensor has already observed. To compute $\hat{x}_t$ for each abstraction θ_t , we assume that the selection is based on the most likely noise realization, $v = 0$, which leads to the observation $o^{\theta_t} = \mathcal{A}^{\theta_t}x$. Note that the Encoder's computational cost is proportional to the cost of the Decoder and the number of the abstractions in the set. The Encoder's algorithm is similar to the one presented in [20], with the preceding differences taken into account.

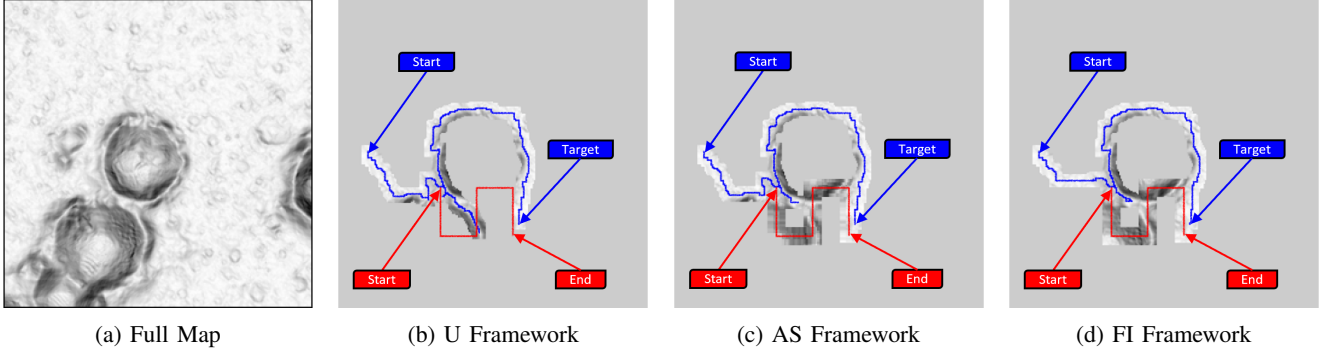


Fig. 5: (a) Mars inclination map computed from the depth map in [32]; (b-d) Final estimated map with $v^A \sim \mathcal{N}(0, 10^{-5}I)$ for (b) Uninformed Framework, (c) Abstraction-Selection Framework, (d) Fully-Informed Framework. Blue and red indicate the cells that the Actor and the Sensor have already traversed, respectively.

V. SIMULATIONS

For the simulations, we utilized two distinct satellite maps, shown in Figure 1 and Figure 5(a). Figure 1 depicts an 128×128 Earth traversability map. Figure 5(a) depicts a 256×256 inclination map of Mars. This map is derived from the depth data provided in [32]. To apply our framework, we first need to normalize the map. Let y_j represent the value of the j -th cell in the depth map. The inclination value for the j -th cell, x_j , is computed by $x_j = \frac{z_j - \min z}{\max z - \min z}$, where $z_j = \sum_{j' \in J} |y_j - y_{j'}|$, and J denotes the index set of neighboring cells to the j -th cell.

Three different scenarios were studied using these maps. In the first scenario, the Mars map was used to evaluate our algorithm in a challenging environment, characterized by a large map size and varying levels of Actor perception noise. The second scenario is a target tracking operation, where the objective is for the Actor to reach a moving target with a known location on the Earth's map. In the third scenario, we compared the performance of our proposed framework with that of the framework presented in [20] in terms of computational time. In all scenarios, we utilized a single Actor-Sensor pair, both starting simultaneously and moving one cell per timestep. The Sensor, designed to operate as a surveillance drone, has the capability to move over obstacles and follows a predefined lawnmower path, as shown in Figure 5. Both robots are positioned at the center of their respective local maps and the Actor's control set is $U = \{\text{UP, DOWN, LEFT, RIGHT}\}$.

The Actor's Path Planner parameter in (5) is $a = 0.025$, which was tuned based on the range of values of $\hat{x}$. ϵ varies depending on the specific scenario, as detailed in Section IV-B. For the parameter in (7), we use $\sigma = 20$. The communication cost, as defined in (8), is given by $\lambda(\theta) = 0.02k^\theta$, where k^θ depends on the resolution of the abstraction template, indicating that the cost of abstraction is proportional to the cardinality of the abstracted map.

A. Performance Metrics

In the first two scenarios, we compare our Abstraction Selection (AS) framework to a Fully-Informed (FI) framework, where in the latter the Sensor transmits all observed cells at each timestep. The comparison focuses on their accumulated

path cost and communication cost. In the third scenario, we compare the proposed framework to the one presented in [20] based on the average computational time of the Actor's Decoder using a computer equipped with a 2.3 GHz, 8-Core, 11th Gen Intel Core i7-11800H CPU and 16GB of RAM. The simulations were conducted in MATLAB.

We compute the Actor's accumulated cost using $\mathcal{C} = \sum_{\mathbf{p} \in \pi_f} c(\mathbf{p})$, where $c(\mathbf{p})$ is computed using the first scale of (5), and π_f represents the set of all cells the Actor traversed to reach its goal, including any duplicate cells. The average cost ratio is

$$r_{\text{cost}} = \frac{1}{n_{\text{sim}}} \sum_{i=0}^{n_{\text{sim}}} \frac{\mathcal{C}(i)}{\mathcal{C}_{\text{max}}(i)}, \quad (9)$$

where n_{sim} is the total simulation number, $\mathcal{C}(i)$ denotes the accumulated cost for simulation i , and $\mathcal{C}_{\text{max}}$ is the accumulated cost employing an Uninformed (U) framework, where the Actor reaches its target without any assistance from the Sensor. For the communication cost, we compute the average ratio of the transmitted bits

$$r_{\text{bits}} = \frac{1}{n_{\text{sim}}} \sum_{i=0}^{n_{\text{sim}}} \frac{\mathcal{B}(i)}{\mathcal{B}_{\text{max}}(i)}, \quad (10)$$

where $\mathcal{B} = \sum_{t=0}^{T^S} n_t$ with n_t given in (1) and referring to the number of bits sent by the Sensor at t , and T^S is the Sensor's time horizon (equal to the number of timesteps that it operates). We set $n_m = 12$, and $n_a = 4$. $\mathcal{B}_{\text{max}}$ refers to the communication cost $\mathcal{B}$ of the FI framework.

B. Scenario 1: Mars

The objective of this scenario is for the Actor to reach a stationary location on the Mars map, shown in Figure 5(a). The Actor's field of view is an 11×11 grid centered on its current position, its initial position is $\mathbf{p}_0^A = (25, 130)$ and aims to reach its goal at $\mathbf{p}_G^A = (150, 70)$. The Sensor covers a larger area with a 15×15 grid, and uses a finite set of 16 abstractions, depicted in Figure 2. It starts at $\mathbf{p}_0^S = (85, 100)$, and its time horizon is $T^S = 180$. The communication channel noise is modelled as $v^S \sim \mathcal{N}(0, 10^{-4}I)$. The threshold parameter of the Path Planner is set to $\epsilon = 0.201$, chosen to ensure the robot avoids traversing areas with

TABLE I: Results for Scenario 1 (Mars).

	$v^A \sim \mathcal{N}(0, 10^{-3}I)$		$v^A \sim \mathcal{N}(0, 10^{-5}I)$	
Metrics	AS Fram.	FI Fram.	AS Fram.	FI Fram.
$\mathcal{C}(i)$	55.9	47.3	52.2	47.5
$B(i)$	7692	486000	7680	486000
$r_{\text{cost}} (\%)$	55.3	46.8	77.6	70.7
$r_{\text{bits}} (\%)$	1.6	100	1.6	100

TABLE II: Results for Scenario 2 (Earth).

Metrics	AS Fram.	FI Fram.
$\mu_C \pm \sigma_C$	19.4 ± 0.9	16.7 ± 6.5
$\mu_B \pm \sigma_B$	6602 ± 115	283500 ± 0
$r_{\text{cost}} (\%)$	22.2	19.0
$r_{\text{bits}} (\%)$	2.3	100

inclinations beyond a certain limit for safety and power-saving reasons. In our map, this value corresponds to 2.2m per cell of average inclination. The initial belief is set to $x_0 \sim \mathcal{N}(0.2, I)$ since its mean needs to be less than ϵ .

The results for two different values of Actor's perception noise v^A are presented in Table I. Considering that r_{cost} is computed using the U framework, we can conclude that both the FI and AS frameworks significantly reduced the accumulated cost compared to the U Framework. However, our proposed AS framework stands out by achieving a remarkable 98% reduction in communicated information compared to the FI framework. It is important to note that, in the FI framework, the Sensor transmits all observed data at each timestep due to the presence of noise. If the noise was negligible, the Sensor would only need to send the newly observed cells. Figure 5(b - d) presents the final timestep of the simulation by using the three frameworks.

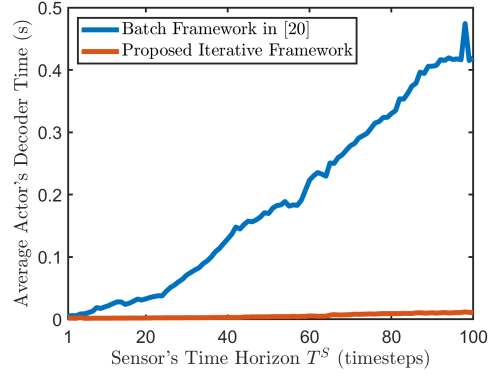
C. Scenario 2: Earth (Moving Target)

In this scenario, we conducted 100 simulations using the Earth map shown in Figure 1. In each simulation, the target starts at position $\mathbf{p}_G^A(t=0) = (90, 49)$ and follows a random path. To ensure that the Actor can always reach the target, we assume that the target moves every even timestep. The Sensor's field of view and set of abstractions is the same as in scenario 1, its initial position is $\mathbf{p}_0^S = (46, 62)$, and its time horizon is $T^S = 105$. The Actor's field of view is 5×5 and its initial location is $\mathbf{p}_0^A = (12, 57)$. The characteristics of the noise are $v^S \sim \mathcal{N}(0, 10^{-5}I)$, and $v^A \sim \mathcal{N}(0, 10^{-6}I)$. The Path Planner parameter is $\epsilon = 0.501$, while the map's initial belief is $x_0 \sim \mathcal{N}(0.5, I)$.

Table II presents the results of the simulations. The first two metrics are the mean and the standard deviation of the results over all the 100 simulations. We observe that the AS framework performs similarly to the FI framework, based on their accumulated cost. Additionally, similar to scenario 1, the AS Framework achieves a 98% reduction in communicated information, compared to the FI framework.

D. Computational Time Analysis

For the third scenario, we compared our proposed framework with the one introduced in [20], based on the computational time of the Actor's Decoder Algorithm (see Algorithm

Fig. 6: Average computational time of Actor's Decoder vs. Sensor's time horizon T^S .

1). To assess the algorithm's complexity, we varied the Sensor's time horizon T^S , as it is proportional to the number of sent abstractions. We used the map shown in Figure 1.

The Actor has the same field of view and initial position as in scenario 2, while its target location is fixed at $\mathbf{p}_G^A = (86, 49)$. The Sensor's field of view is 7×7 , its start location is $\mathbf{p}_0^S = (46, 62)$, and utilizes a finite set of 10 abstractions, as illustrated in Figure 6 of [20]. No noise is introduced in the simulations, as [20] does not provide a straightforward method for handling noise. The initial belief and the Path Planner parameter are the same as in scenario 2.

Figure 6 shows the average Actor's Decoder runtime (over all timesteps of each simulation) for 100 simulations, where each simulation is performed with a different Sensor Time Horizon T^S . We observe that the rate of increase in time using the framework from [20] is significantly larger than with the proposed framework. This is because the former relies on the history of abstractions, considering that the Sensor's time horizon T^S is proportional to the number of transmitted abstractions by the end of the simulation. It should be noted that although the Decoder in [20] provides slightly better estimation compared to the proposed Decoder, it ultimately does not impact the Actor's accumulated cost $\mathcal{C}$. Therefore, its effect on the problem's objective is negligible.

VI. CONCLUSIONS

In this paper, we addressed the problem of facilitating a robot's path-planning operation using a mobile sensor. The sensor selects the optimal compressed representation of its local map, transmitting it to the robot while automatically considering both communication costs and the predicted robot trajectory. We proposed a novel framework with a decoder that estimates the unknown environment with reduced computational complexity compared to the state-of-the-art approach, while also handling noise. Our simulations demonstrated the framework's effectiveness, achieving a 98% reduction in transmitted data compared to sending raw data, while maintaining comparable path lengths and low computational time. Future work will focus on implementing this framework in real-world UAV-UGV collaborative operations.

REFERENCES

- [1] J. P. Queralta, J. Taipalmaa, B. Can Pullinen, V. K. Sarker, T. Nguyen Gia, H. Tenhunen, M. Gabbouj, J. Raitoharju, and T. West-erlund, "Collaborative multi-robot search and rescue: Planning, co-ordination, perception, and active vision," *IEEE Access*, vol. 8, pp. 191 617–191 643, Oct 12 2020.
- [2] J. Thangavelautham, A. Chandra, and E. Jensen, "Autonomous robot teams for lunar mining base construction and operation," in *IEEE Aerospace Conference*, March 07-14 2020, pp. 1–16.
- [3] 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, Oct. 2022.
- [4] M. A. Schack, J. G. Rogers, and N. T. Dantam, "The sound of silence: Exploiting information from the lack of communication," *IEEE Robotics and Automation Letters*, vol. 9, no. 7, pp. 6736–6743, June 2024.
- [5] Y. Yue, C. Zhao, Z. Wu, C. Yang, Y. Wang, and D. Wang, "Collaborative semantic understanding and mapping framework for autonomous systems," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 2, pp. 978–989, April 2020.
- [6] A. Cunningham, V. Indelman, and F. Dellaert, "DDF-SAM 2.0: Consistent distributed smoothing and mapping," in *IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany, May 06-10, 2013, pp. 5220–5227.
- [7] D. Maity and J. S. Baras, "Minimal feedback optimal control of linear-quadratic-gaussian systems: No communication is also a communication," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 2201–2207, 2020.
- [8] Y. Chang, L. Ballotta, and L. Carlone, "D-lite: Navigation-oriented compression of 3d scene graphs under communication constraints," 2023, arXiv:2209.06111.
- [9] D. F. Delchamps, "Stabilizing a linear system with quantized state feedback," *IEEE Transactions on Automatic Control*, vol. 35, no. 8, pp. 916–924, Aug. 1990.
- [10] R. W. Brockett and D. Liberzon, "Quantized feedback stabilization of linear systems," *IEEE Transactions on Automatic Control*, vol. 45, no. 7, pp. 1279–1289, July 2000.
- [11] G. N. Nair and R. J. Evans, "Stabilizability of stochastic linear systems with finite feedback data rates," *SIAM Journal on Control and Optimization*, vol. 43, no. 2, pp. 413–436, July 2004.
- [12] V. Kostina and B. Hassibi, "Rate-cost tradeoffs in control," *IEEE Transactions on Automatic Control*, vol. 64, no. 11, pp. 4525–4540, Nov. 2019.
- [13] S. Yüksel, "A note on the separation of optimal quantization and control policies in networked control," *SIAM Journal on Control and Optimization*, vol. 57, no. 1, pp. 773–782, 2019.
- [14] D. Maity and P. Tsiotras, "Optimal controller synthesis and dynamic quantizer switching for linear-quadratic-Gaussian systems," *IEEE Transactions on Automatic Control*, vol. 67, no. 1, pp. 382–389, Jan. 2022.
- [15] —, "Optimal quantizer scheduling and controller synthesis for partially observable linear systems," *SIAM Journal on Control and Optimization*, vol. 61, no. 4, pp. 2682–2707, Aug. 2023.
- [16] M. Afshari, D. Maity, and P. Tsiotras, "Communication-and control-aware optimal quantizer selection for multi-agent control," *IEEE Control Systems Letters*, 2024.
- [17] G. Damigos, N. Stathouloupoulos, A. Koval, T. Lindgren, and G. Nikolakopoulos, "Communication-aware control of large data transmissions via centralized cognition and 5G networks for multi-robot map merging," *Journal of Intelligent & Robotic Systems*, vol. 110, no. 22, 2024.
- [18] R. Marcotte, X. Wang, D. Mehta, and E. Olson, "Optimizing multi-robot communication under bandwidth constraints," *Autonomous Robots*, vol. 44, no. 1, pp. 43–55, Jan. 2020.
- [19] Q. Li, F. Gama, A. Ribeiro, and A. Prorok, "Graph neural networks for decentralized multi-robot path planning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Macau, China, Nov. 4-8, 2019, pp. 11 785–11 792.
- [20] E. Psomiadis, D. Maity, and P. Tsiotras, "Communication-aware map compression for online path-planning," in *IEEE International Conference on Robotics and Automation (ICRA)*, Yokohama, Japan, May 13-17, 2024, pp. 12 368–12 374.
- [21] A. Elfes, "Sonar-based real-world mapping and navigation," *IEEE Transactions on Robotics and Automation*, vol. 3, no. 3, pp. 249–265, 1987.
- [22] H. P. Moravec, "Sensor fusion in certainty grids for mobile robots," *AI Magazine*, vol. 9, no. 2, pp. 61–74, 1988.
- [23] R. V. Cowlagi and P. Tsiotras, "Multiresolution motion planning for autonomous agents via wavelet-based cell decompositions," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 5, pp. 1455–1469, Oct. 2012.
- [24] G. K. Kraetzschmar, G. P. Gassull, and K. Uhl, "Probabilistic quadrees for variable-resolution mapping of large environments," *IFAC Proceedings Volumes*, vol. 37, no. 8, pp. 675–680, July 2004.
- [25] D. T. Larsson, D. Maity, and P. Tsiotras, "Q-tree search: An information-theoretic approach toward hierarchical abstractions for agents with computational limitations," *IEEE Transactions on Robotics*, vol. 36, no. 6, pp. 1669–1685, Dec. 2020.
- [26] E. Psomiadis, D. Maity, and P. Tsiotras, "Multi-agent task-driven exploration via intelligent map compression and sharing," in *International Symposium on Distributed Autonomous Robotic Systems (DARS)*, New York, NY, Oct 28-30, 2024.
- [27] S. Boyd and L. Vandenberghe, *Convex Optimization*, 1st ed. Cambridge, UK: Cambridge University Press, 2004.
- [28] J. L. Speyer and W. H. Chung, *Stochastic Processes, Estimation, and Control*. Los Angeles, CA: Society for Industrial and Applied Mathematics, 2008.
- [29] D. Simon, "Kalman filtering with state constraints: A survey of linear and nonlinear algorithms," *Control Theory and Applications, IET*, vol. 4, pp. 1303 – 1318, Sep. 2010.
- [30] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, Dec. 1959.
- [31] D. T. Larsson, D. Maity, and P. Tsiotras, "Information-theoretic abstractions for planning in agents with computational constraints," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7651–7658, Oct. 2021.
- [32] HiRISE, "Digital Terrain Models (DTMs), Map: DTEEC_041277_2115_040776_2115_A01," <https://www.uahirise.org> (accessed Sep. 1, 2024).