Information-Theoretic Abstractions for Planning in Agents with Computational Constraints

Daniel T. Larsson Dipankar Maity Panagiotis Tsiotras

Abstract—**In this paper, we develop a framework for path-planning on abstractions that are not provided to the agent a priori but instead emerge as a function of the available computational resources. We show how a path-planning problem in an environment can be systematically approximated by solving a sequence of easier-to-solve problems on abstractions of the original space. The properties of the problem are analyzed, and a number of theoretical results are presented and discussed. A numerical example is presented to show the utility of the approach and to corroborate the theoretical findings. We conclude by providing a discussion detailing the connections of the proposed approach to anytime algorithms and bounded rationality.**

Index Terms—**hierarchical abstractions, planning, information theory, information bottleneck method.**

I. INTRODUCTION

PATH and motion planning for autonomous systems has long been an area of research within the robotics and artificial intelligence communities. This has led to the development of a number of frameworks which formulate planning tasks in terms of mathematical optimization problems, which can then be solved by utilizing techniques from optimization and optimal control theory [1], [2]. However, planning in complex domains can be a challenging problem, and requires the agents to spend time and computational resources in order to find solutions, giving rise to an intrinsic need for agents to balance computational complexity with optimality of the resulting plan [3]–[7].

As a result, a number of approaches within the path-planning community have been developed that aim to explicitly capture the interplay between complexity and optimality. For example, in [5], [8]–[13], the authors utilize wavelets to obtain multi-resolution representations of two-dimensional environments for planning. The use of abstractions for path-planning allows these works to leverage the computational benefits of executing graph-search algorithms, such as Dijkstra or A*, on reduced graphs of the environment that contain fewer vertices as compared to the original, full-resolution, representation.

In a similar spirit, other works [4], [14], [15] consider abstractions for planning, but instead employ hierarchical representations of the world in the form of multi-resolution quadtrees and octrees. The use of probabilistic tree structures enables these works to incorporate environment uncertainty [16]. With this added flexibility, these approaches can be used in an on-line manner, allowing autonomous agents to plan based on occupancy grid (OG) representations of the world that are dynamically updated as the agent interacts with the environment. To strike a balance between the complexity of the search and satisfactory performance, the aforementioned works recursively re-solve the planning problem as the agent traverses the world.

It should be noted that the interplay between complexity and optimality is not unique to the path-planning community. Recent work related to bounded-rational decision making has illustrated a growing need to develop decision-making frameworks for agents that are resource limited [17]–[22]. This area of research considers limitations in the traditional assumptions of artificial intelligence, and approaches problems by viewing agents as resource-limited entities that are constrained in terms of their information-processing capabilities. To model such agents, the authors in [19] utilize concepts from information theory, arguing that bounded-rational decision making can be modeled by considering Kullback-Leibler (KL) divergence constraints added to traditional maximum expected utility problems. Extensions of this work to sequential decision-making problems in stochastic domains is considered in [17], [21], whereby Markov Decision Processes (MDPs) are utilized with information-theoretic constraints to formulate information-limited MDPs (IL-MDPs). The frameworks include a trade-off parameter that balances the optimality of the decision policy and the effort required to obtain it, as measured by a KL-divergence measure between the resulting posterior policy and a default prior policy. These approaches offer one perspective of bounded-rational decision making and provide for interesting connections with information-theoretic frameworks for compression, such as rate-distortion theory [17], [19].

In this paper, we consider complexity reduction in path-planning problems by means of graph abstractions for resource-limited agents by combining aspects from both the planning and bounded-rational decision-making communities. Our contribution is two-fold. Firstly, we employ an

Manuscript received: February, 26, 2021; Revised May, 26, 2021; Accepted July, 3, 2021.

This paper was recommended for publication by Editor Editor A. Name upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by ONR awards N00014-18-1-2375 and N00014-18-1-2828 and by ARL under DCIST CRA W911NF-17-2-0181.

D. Larsson is a PhD student with the Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, 30332-0150, USA. Email: daniel.larsson@gatech.edu

D. Maity is an Assistant Professor in the Department of Electrical and Computer Engineering at the University of North Carolina, Charlotte, NC, 28223, USA. Email: dmaity@uncc.edu

P. Tsiotras is the Andrew and Lewis Chair Professor with the Guggenheim School of Aerospace Engineering and the Institute for Robotics and Intelligent Machines, Georgia Institute of Technology, Atlanta, GA, 30332-0150, USA. Email: tsiotras@gatech.edu

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

information-theoretic approach to generate multi-resolution abstractions that are not provided a priori for the purposes of path-planning and secondly, our framework couples the environment resolution to the resulting path quality. To the best of our knowledge, there are no existing approaches that utilize information-theoretic abstractions for complexity reduction in path-planning that also guarantee the monotonic improvement of the path-cost as a function of environment resolution. Coupling the path-cost with the environment resolution provides a link between the path quality, the complexity of executing graph-search algorithms and the information-processing capabilities of the agent determined by the information contained in the generated abstractions. In summary, our framework: (i) utilizes concepts from information theory to obtain reduced environment representations as a function of agent information-processing capabilities, and (ii) provides provable guarantees on the monotonic improvement of the path-cost as a function of environment resolution.

II. PRELIMINARIES

Denote the set of real numbers by $\mathbb{R}$ and, for any positive integer d, let $\mathbb{R}^d$ denote the d-dimensional Euclidean space. Assume that the environment $\mathcal{W} \subset \mathbb{R}^d$ is given by a d-dimensional OG and that there exists an integer $\ell > 0$ and real number $a \in (0, \infty)$ such that the environment is contained within a hypercube of side length $a \cdot 2^\ell$. The real number a is a scaling factor, and so we will assume, without loss of generality, that $a = 1$. The environment is represented as a tree $\mathcal{T} = (\mathcal{N}(\mathcal{T}), \mathcal{E}(\mathcal{T}))$, where the edge set $\mathcal{E}(\mathcal{T})$ describes the relationship between the nodes in $\mathcal{N}(\mathcal{T})$. In what follows, we restrict our attention to the case where the tree representation is that of a quadtree, however the contributions of this paper are valid for any tree structure. Let $\mathcal{T}^\mathcal{Q}$ be the space of all feasible quadtree representations of $\mathcal{W}$, where each $\mathcal{T} \in \mathcal{T}^\mathcal{Q}$ encodes a multi-resolution, hierarchical, representation of the world. Take $\mathcal{T}_\mathcal{W} \in \mathcal{T}^\mathcal{Q}$ be the quadtree corresponding to the original environment $\mathcal{W}$; that is, $\mathcal{T}_\mathcal{W}$ encodes the finest resolution depiction of $\mathcal{W}$.

Consider any node $n \in \mathcal{N}(\mathcal{T}_\mathcal{W})$ at depth $k \in \{0, \ldots, \ell\}$, then $n' \in \mathcal{N}(\mathcal{T}_\mathcal{W})$ is a child of n if the following hold:

1) Node n' is at depth $k+1$ in $\mathcal{T}_\mathcal{W}$,
2) Nodes n and n' are incident to a common edge, i.e., $(n, n') \in \mathcal{E}(\mathcal{T}_\mathcal{W})$.

In the sequel, we let the set of child nodes for any $n \in \mathcal{N}(\mathcal{T}_\mathcal{W})$ be denoted by $\mathcal{C}(n)$ and $\mathcal{N}_k(\mathcal{T}_\mathcal{W})$ to be the set of nodes at depth k. For any $\mathcal{T} \in \mathcal{T}^\mathcal{Q}$ we take $\mathcal{N}_{\text{leaf}}(\mathcal{T}) = \{n' \in \mathcal{N}(\mathcal{T}) : \mathcal{C}(n') \cap \mathcal{N}(\mathcal{T}) = \varnothing\}$ to denote the set of leaf nodes and $\mathcal{N}_{\text{int}}(\mathcal{T}) = \mathcal{N}(\mathcal{T}) \setminus \mathcal{N}_{\text{leaf}}(\mathcal{T})$ to be the set of interior nodes of the tree $\mathcal{T}$.

While useful for describing the relationship between nodes in a given tree, the aforementioned sets do not describe how the nodes in the tree $\mathcal{T} \in \mathcal{T}^\mathcal{Q}$ are related to the spatial region described by the environment $\mathcal{W}$. This brings us to the following definition.

Definition 2.1 ([14]): Let $k \in \{0, \ldots, \ell\}$ and $n \in \mathcal{N}_k(\mathcal{T}_\mathcal{W})$. Then the node n:

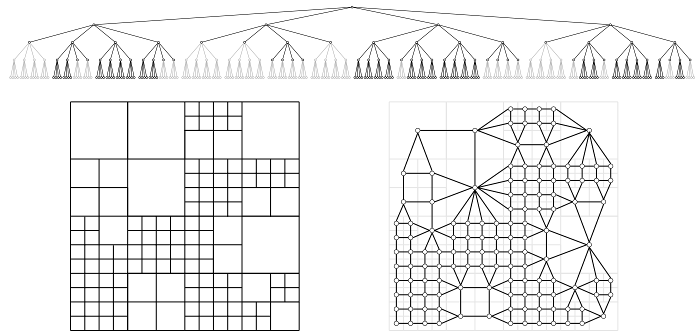


Fig. 1. Tree representation (top) of some $\mathcal{T} \in \mathcal{T}^\mathcal{Q}$, corresponding grid depiction (left) and associated graph (right) for a $2^\ell \times 2^\ell$ with $\ell = 4$ environment. The connectivity of the graph is consistent with the definition of nodal neighbor. The nodes in $\mathcal{T}_\mathcal{W}$ that are not in $\mathcal{T}$ are shown in grey.

1) Is at depth k and has an r-value given by the function $r : \mathcal{N}(\mathcal{T}_\mathcal{W}) \to \{0, \ldots, \ell\}$ defined by the rule $r(n) = \ell - k$. The inverse image of the function r is the set $r^{-1}(L) = \{n \in \mathcal{N}(\mathcal{T}_\mathcal{W}) : r(n) \in L\}$ for any $L \subseteq \{0, \ldots, \ell\}$.
2) Represents a hypercube $H(n) \subseteq \mathcal{W}$ with side length $2^{r(n)}$ and volume $2^{dr(n)}$ centered at the point $\mathbf{p}(n) \in \mathbb{R}^d$.
3) The hypercubes corresponding to the nodes that are the children of n form a partition of $H(n)$. That is,

$$H(n) = \bigcup_{n' \in \mathcal{C}(n)} H(n').$$

In order to utilize the tree $\mathcal{T} \in \mathcal{T}^\mathcal{Q}$ for planning, we must specify how the nodes in the tree $\mathcal{T}$ are connected. To this end, we consider the nodes $n, \hat{n} \in \mathcal{N}(\mathcal{T}_\mathcal{W})$ as *nodal neighbors* if the following statements hold:

1) $\|\mathbf{p}(n) - \mathbf{p}(\hat{n})\|_\infty = 2^{r(n)-1} + 2^{r(\hat{n})-1}$,
2) There exists a unique $i \in \{1, \ldots, d\}$ such that $|[\mathbf{p}(n) - \mathbf{p}(\hat{n})]_i| = 2^{r(n)-1} + 2^{r(\hat{n})-1}$,

where $[\mathbf{p}(n) - \mathbf{p}(\hat{n})]_i$ denotes the i^{th} entry of the vector $\mathbf{p}(n) - \mathbf{p}(\hat{n})$ and $|\cdot|$ is the absolute value. For each tree $\mathcal{T} \in \mathcal{T}^\mathcal{Q}$ there exists an associated graph $\mathcal{G}(\mathcal{T}) = (\mathcal{V}(\mathcal{T}), E(\mathcal{T}))$, constructed from the leaf nodes of $\mathcal{T}$, consisting of a set of vertices $\mathcal{V}(\mathcal{T})$ and edges $E(\mathcal{T})$, where the set $E(\mathcal{T})$ describes the connectivity of the vertices in $\mathcal{V}(\mathcal{T})$. To describe the relation between $\mathcal{V}(\mathcal{T})$ and $\mathcal{N}_{\text{leaf}}(\mathcal{T})$, we define the mapping $\texttt{Node}_{\mathcal{G}(\mathcal{T})} : \mathcal{V}(\mathcal{T}) \to \mathcal{N}(\mathcal{T}_\mathcal{W})$ such that if $n_v \triangleq \texttt{Node}_{\mathcal{G}(\mathcal{T})}(v)$, then the vertex $v \in \mathcal{V}(\mathcal{T})$ corresponds to the node $n_v \in \mathcal{N}(\mathcal{T}_\mathcal{W})$.[1] Thus, for any two vertices $v, \hat{v} \in \mathcal{V}(\mathcal{T})$, $(v, \hat{v}) \in E(\mathcal{T})$ if and only if the nodes $n_v, n_{\hat{v}} \in \mathcal{N}_{\text{leaf}}(\mathcal{T}) \subseteq \mathcal{N}(\mathcal{T}_\mathcal{W})$ are nodal neighbors. A visualization is provided in Fig. 1.

In order to develop and information-theoretic framework for abstraction, we require the formalism of a probability space. Thus, let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space with finite sample space Ω, σ-algebra $\mathcal{F}$, and probability measure $\mathbb{P} : \mathcal{F} \to [0, 1]$. Define random variables $X : \Omega \to \mathcal{N}_{\text{leaf}}(\mathcal{T}_\mathcal{W})$ and $Y : \Omega \to \{0, 1\}$. The distribution $p(x)$ is given by $p(x) = \mathbb{P}(\{\omega \in \Omega : X(\omega) = x\})$, where $p(y)$ is defined analogously. The random variables X and Y represent each of the unit

[1]The mapping $\texttt{Node}_{\mathcal{G}(\mathcal{T})}$ has co-domain $\mathcal{N}(\mathcal{T}_\mathcal{W})$ since the set $\mathcal{N}(\mathcal{T}_\mathcal{W})$ contains all nodes of any tree $\mathcal{T} \in \mathcal{T}^\mathcal{Q}$.

hypercubes of $\mathcal{W}$ and the total cell occupancy, respectively, where for $y \in \Omega_Y = \{0,1\}$, we let $y = 1$ represent the outcome of "occupied" and $y = 0$ correspond to the outcome of "empty". The OG representation of $\mathcal{W}$ provides the conditional distribution $p(y = 1|x)$ for all $x \in \Omega_X$.

III. PROBLEM FORMULATION

Our problem is defined as follows.

Problem 1: Given the tree $\mathcal{T}_{\mathcal{W}}$, a scalar $\varepsilon \in [0,1]$, constants $\lambda_1 \in (0,1], \lambda_2 \in [0,1]$ with $\lambda = (\lambda_1, \lambda_2)$, a start node $s_0 \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}})$ and a goal node $s_g \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}})$, we consider the problem of obtaining a *finest-resolution path* (FRP) $\pi = \{x_0, \dots, x_K\} \subseteq \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}})$ where $x_0 = s_0$, $x_K = s_g$, each $x \in \pi$ is distinct and $x_i, x_{i+1} \in \pi$ are nodal neighbors for all $i \in \{0, \dots, K-1\}$, so as to satisfy

$$\pi^* \in \underset{\pi \in \Pi}{\arg\min}\, J_\varepsilon^\lambda(\pi), \tag{1}$$

where

$$J_\varepsilon^\lambda(\pi) = \sum_{x \in \pi} c_\varepsilon^\lambda(x), \tag{2}$$

and

$$c_\varepsilon^\lambda(x) = \begin{cases} \lambda_1 + \lambda_2 p(y=1|x), & \text{if } x \in \mathcal{P}_\varepsilon, \\ M_\varepsilon^\lambda, & \text{if } x \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}}) \setminus \mathcal{P}_\varepsilon, \end{cases} \tag{3}$$

with $M_\varepsilon^\lambda = 2^{d\ell}(\lambda_1 + \varepsilon\lambda_2) + \gamma$ for any $\gamma > 0$,[2] $\mathcal{P}_\varepsilon = \{x \in \Omega_X : p(y=1|x) \leq \varepsilon\}$ and where Π denotes the set of FRPs leading from the start node s_0 to the goal s_g in the tree $\mathcal{T}_{\mathcal{W}}$. We aim to reduce the computational complexity of the planning problem (1) by leveraging environment abstractions that can be tailored to agent resource constraints. △

We call an FRP π for which $\pi \subseteq \mathcal{P}_\varepsilon$ an ε-*feasible* FRP. The role of ε is to define a feasible cell when the obstacle information is encoded probibalistically, and M_ε^λ is a constant that penalizes nodes considered to be obstacles. The value of M_ε^λ is chosen so as to ensure search algorithms do not include infeasible nodes as part of an FRP unless no feasible paths exist, as we do not exclude nodes $x \in \mathcal{P}_\varepsilon^c$ from the search. By not removing the nodes $x \in \mathcal{P}_\varepsilon^c$, we guarantee that Π is non-empty. The cost function (3) is inspired by previous works that have considered planning on multi-scale abstractions [14]. The approach in this paper is distinct from existing works in that we: (i) utilize an information-theoretic framework to generate abstractions not provided a priori, and (ii) provide theoretical results that couple environment resolution and path cost.

The resulting search problem on the graph $\mathcal{G}(\mathcal{T}_{\mathcal{W}})$ may be computationally expensive. However, notice that by changing the leaf nodes of the tree $\mathcal{T} \in \mathcal{T}^{\mathcal{Q}}$, we alter the graph representation $\mathcal{G}(\mathcal{T})$ and, as a result, the complexity of the resulting graph-search. Thus, instead of solving (1) directly on $\mathcal{G}(\mathcal{T}_{\mathcal{W}})$, we propose to approximate (1) by a computationally easier-to-solve problem on a graph $\mathcal{G}(\mathcal{T})$ for some $\mathcal{T} \in \mathcal{T}^{\mathcal{Q}}$. The challenge is then to select the tree $\mathcal{T} \in \mathcal{T}^{\mathcal{Q}}$ as a function of agent resource constraints.

[2]Strictly speaking, $\gamma > 0$ may be any positive number. However, we let $\gamma = 2$ in this paper.

IV. SOLUTION APPROACH

Our approach to approximating Problem 1 proceeds in two phases. The first phase consists of selecting a tree $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ according to the agent's information-processing capabilities. The second phase is concerned with defining the planning problem on the abstract representations of the world.

A. Information-Theoretic Tree Selection

The mutual information between a compressed representation Z of X, given by

$$I(Z;X) \triangleq \sum_{z,x} p(z,x) \log \frac{p(z,x)}{p(z)p(x)}, \tag{4}$$

measures the amount of compression between the random variables X and Z [23]. However, maximizing compression via the minimization of $I(Z;X)$ is not a well-posed problem, as $I(Z;X) = 0$ is always attainable. Instead, the compression problem must be constrained by a measure that captures how good of a compressed representation Z is of X.

One particular method of interest is the *information bottleneck* (IB), which defines the quality of an abstraction by the amount of information retained in the compressed representation regarding a third, relevant, random variable [24]. The IB method considers the problem

$$p^*(z|x) = \underset{p(z|x)}{\arg\max}\, I(Z;Y) - \frac{1}{\beta} I(Z;X), \tag{5}$$

where X, Y, Z are random variables corresponding to the original signal, relevant variable and compressed signal, respectively, $I(Z;Y)$ is the amount of relevant information retained in the compressed representation, and $p(z|x)$ maps outcomes of X to outcomes of Z. The IB method assumes the joint distribution factors according to $p(x,y,z) = p(z|x)p(x,y)$, which implies $I(Z;Y) \leq I(X;Y)$ [24]. The trade-off parameter $\beta > 0$ balances the amount of relevant information retained in the compressed representation vs. the achieved compression of the original signal.

The problem (5) can be formulated over the space of multi-resolution trees by noting that each $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ corresponds to an encoder of the form $p_q(z|x)$, where $p_q(z|x)$ specifies how the leaf nodes $x \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}})$ are mapped to nodes $z \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_q)$ to create the tree $\mathcal{T}_q$ [25]. Thus, the IB problem over the space of multi-resolution trees is formulated as

$$\mathcal{T}_{q^*} \in \underset{\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}}{\arg\max}\, L_Y(\mathcal{T}_q; \beta), \tag{6}$$

where

$$L_Y(\mathcal{T}_q;\beta) = I(Z;Y) - \frac{1}{\beta} I(Z;X), \tag{7}$$

and the quantities $I(Z;Y)$ and $I(Z;X)$ are evaluated using the joint distribution $p_q(x,y,z) = p_q(z|x)p(x,y)$. In contrast to the original IB problem (5), the added constraint requiring $\mathcal{T}_q \in \mathcal{T}^{\mathcal{Q}}$ presents a significant challenge in obtaining a solution to (6). It was recently shown that (6) can be solved by employing an algorithm called Q-tree search [25]. We will employ the Q-tree search algorithm to obtain multi-resolution abstractions of the environment as a function of $\beta > 0$.

While we employ the abstraction framework from [25], we emphasize that [25] does not address the use of abstractions for the purposes of path-planning. The path-planning aspect of our problem is novel and has not been previously discussed. We present the path-planning details next.

B. Path-Planning on Abstractions

Given a sequence of strictly increasing $\beta > 0$, denoted by $\{\beta_i\}_{i=1}^N$, we generate a corresponding sequence of trees $\{\mathcal{T}_{\beta_i}\}_{i=1}^N$ by employing the Q-tree search algorithm to solve the information-theoretic problem in Section IV-A. A corresponding sequence of graphs $\{\mathcal{G}(\mathcal{T}_{\beta_i})\}_{i=1}^N$ can then be constructed, where each $\mathcal{G}(\mathcal{T}_{\beta_i})$ for $i \in \{1,\ldots,N\}$ represents a multi-resolution depiction of the environment $\mathcal{W}$ with fewer vertices than $\mathcal{G}(\mathcal{T}_\mathcal{W})$. We will now use these reduced graphs to form approximations to Problem 1, which brings us to the following definitions.

Definition 4.1 ([25]): Let $n \in \mathcal{N}(\mathcal{T})$ be a node in the tree $\mathcal{T} \in \mathcal{T}^\mathcal{Q}$. *The subtree of* $\mathcal{T} \in \mathcal{T}^\mathcal{Q}$ *rooted at node* n is denoted by $\mathcal{T}_{(n)}$ and has node set

$$\mathcal{N}\left(\mathcal{T}_{(n)}\right) = \Big\{n' \in \mathcal{N}(\mathcal{T}) : n' \in \bigcup_i \mathcal{D}_i\Big\},$$

where $\mathcal{D}_1 = \{n\}$, $\mathcal{D}_{i+1} = \mathcal{A}(\mathcal{D}_i)$, and

$$\mathcal{A}(\mathcal{D}_i) = \Big\{n' \in \mathcal{N}(\mathcal{T}_\mathcal{W}) : n' \in \bigcup_{\hat{n}\in\mathcal{D}_i} \mathcal{C}(\hat{n})\Big\}.$$

Definition 4.2: An *abstract path* (AP) is a sequence of nodes $\hat{\pi} = \{z_0,\ldots,z_R\} \subseteq \mathcal{N}_{\text{leaf}}(\mathcal{T})$ for some $\mathcal{T} \in \mathcal{T}^\mathcal{Q}$, $\mathcal{T} \neq \mathcal{T}_\mathcal{W}$, such that each $z \in \hat{\pi}$ is distinct, the nodes z_0 and z_R satisfy $s_0 \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}(z_0)})$ and $s_g \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}(z_R)})$, respectively, and if $R > 0$ then z_i, z_{i+1} are nodal neighbors for all $i \in \{0,\ldots,R-1\}$. An *ε-feasible abstract path* (ε-AP) is an AP $\hat{\pi}$ such that $\bigcup_{z\in\hat{\pi}} \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}(z)}) \subseteq \mathcal{P}_\varepsilon$.

To obtain an AP requires the specification of a cost-function for abstracted representations. This is challenging as the cost must: (i) be consistent with an FRP on the finest resolution; (ii) appropriately account for the cost of traversing aggregated nodes; and (iii) monotonically decrease with increased resolution, or equivalently, with increased β. The criterion (iii) is needed to ensure that the paths $\{\hat{\pi}_{\beta_i}\}_{i=1}^N$ represent approximations to an FRP π in that the cost of a path $\hat{\pi}_{\beta_i}$ should approach that of an FRP π as $\beta_i \to \infty$.

To plan on abstractions, we define $V_\varepsilon^\lambda : \mathcal{N}(\mathcal{T}_\mathcal{W}) \to (0,\infty)$ as

$$V_\varepsilon^\lambda(n) = \begin{cases} c_\varepsilon^\lambda(n), & n \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_\mathcal{W}), \\ \frac{1}{2^d}\sum_{n'\in\mathcal{C}(n)} V_\varepsilon^\lambda(n'), & \text{otherwise}, \end{cases} \tag{8}$$

and consider the objective

$$\hat{J}_\varepsilon^\lambda(\hat{\pi};\beta) = \sum_{z\in\hat{\pi}} 2^{dr(z)} V_\varepsilon^\lambda(z). \tag{9}$$

Note that $\hat{J}_\varepsilon^\lambda(\hat{\pi};\beta)$ depends on the trade-off parameter $\beta > 0$, as β determines the tree $\mathcal{T}_\beta \in \mathcal{T}^\mathcal{Q}$ on which the AP $\hat{\pi}$ is planned. Given $\beta > 0$, we consider the problem

$$\hat{\pi}_\beta^* \in \operatorname*{arg\,min}_{\hat{\pi}\in\hat{\Pi}_\beta} \hat{J}_\varepsilon^\lambda(\hat{\pi};\beta), \tag{10}$$

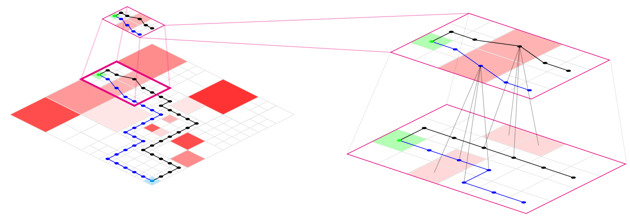


Fig. 2. Example OG of an environment $\mathcal{W}$ with two AP $\hat{\pi}_1$ (blue) and $\hat{\pi}_2$ (black) leading from a given start location (cyan) to goal location (green). For probabilistic obstacles (red), shading scales with the probability of occupancy. Notice that both $\hat{\pi}_1$ and $\hat{\pi}_2$ pass through identical (adjacent) abstracted cells with non-zero probability of occupancy. To determine the feasibility of these paths requires refinement, shown to the right. Observe that, upon refinement, the path $\hat{\pi}_1$ (blue) will be deemed infeasible, as it is not possible to traverse the left abstracted cell in the direction stipulated by $\hat{\pi}_1$. In contrast, the path $\hat{\pi}_2$ (black) is feasible, since the right abstracted cell can be traversed in the direction required by $\hat{\pi}_1$. Our definition of feasibility precludes an agent from discovering a path is infeasible upon refinement.

where $\hat{\Pi}_\beta$ is the set of APs in $\mathcal{T}_\beta \in \mathcal{T}^\mathcal{Q}$. What we must show is that the objective function value of (9) monotonically decreases with increased $\beta > 0$. The following theorem establishes this result.

Theorem 4.3: Let $\varepsilon \in [0,1]$ and assume that there exists $\beta_2 > \beta_1 > 0$ such that the corresponding trees $\mathcal{T}_{\beta_1}, \mathcal{T}_{\beta_2} \in \mathcal{T}^\mathcal{Q}$ satisfy $\mathcal{N}(\mathcal{T}_{\beta_2}) \setminus \mathcal{N}(\mathcal{T}_{\beta_1}) = \mathcal{C}(n)$ for some $n \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\beta_1})$. Furthermore, let $\hat{\pi}_{\beta_1}^* \subseteq \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\beta_1})$ denote an abstract path in the tree $\mathcal{T}_{\beta_1} \in \mathcal{T}^\mathcal{Q}$ satisfying $\hat{\pi}_{\beta_1}^* \in \arg\min_{\hat{\pi}_{\beta_1}\in\hat{\Pi}_{\beta_1}} \hat{J}_\varepsilon^\lambda(\hat{\pi}_{\beta_1};\beta_1)$. Then there exists an abstract path $\hat{\pi}_{\beta_2} \subseteq \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\beta_2})$ such that $\hat{J}_\varepsilon^\lambda(\hat{\pi}_{\beta_1}^*;\beta_1) \geq \hat{J}_\varepsilon^\lambda(\hat{\pi}_{\beta_2};\beta_2)$.

Proof: The proof is presented in Appendix A. ■

By definition, $\hat{J}_\varepsilon^\lambda(\hat{\pi}_{\beta_2};\beta_2) \geq \hat{J}_\varepsilon^\lambda(\hat{\pi}_{\beta_2}^*;\beta_2)$ for all $\hat{\pi}_{\beta_2} \in \hat{\Pi}_{\beta_2}$, and hence Theorem 4.3 establishes that $\hat{J}_\varepsilon^\lambda(\hat{\pi}_{\beta_1}^*;\beta_1) \geq \hat{J}_\varepsilon^\lambda(\hat{\pi}_{\beta_2}^*;\beta_2)$. Note that the result holds even if two consecutive trees in the sequence $\{\mathcal{T}_{\beta_i}\}_{i=1}^N$ do not satisfy $\mathcal{N}(\mathcal{T}_{\beta_{i+1}}) \setminus \mathcal{N}(\mathcal{T}_{\beta_i}) = \mathcal{C}(n)$ for some $n \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\beta_i})$. This is because moving from $\mathcal{T}_{\beta_i}$ to $\mathcal{T}_{\beta_{i+1}}$ can be done by considering another sequence $\{\mathcal{T}_u\}_{u=0}^m$ where $\mathcal{T}_0 = \mathcal{T}_{\beta_i}$, $\mathcal{T}_m = \mathcal{T}_{\beta_{i+1}}$, and $\mathcal{N}(\mathcal{T}_{u+1}) \setminus \mathcal{N}(\mathcal{T}_u) = \mathcal{C}(n)$ holds for some $n \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_u)$ and all $u \in \{0,\ldots,m-1\}$.

While Theorem 4.3 guarantees the monotonic improvement of the cost as a function of resolution, it does not guarantee that the cost of an AP converges to that of an FRP as $\beta \to \infty$. To address this, we require the following proposition.

Proposition 4.4: Let $\Delta I_Y : \mathcal{N}_{\text{int}}(\mathcal{T}_\mathcal{W}) \to [0,\infty)$ be the change in relevant information by expanding the node $n \in \mathcal{N}_{\text{int}}(\mathcal{T}_\mathcal{W})$.[3] Then the Q-tree search algorithm returns the tree $\mathcal{T}_\mathcal{W}$ as $\beta \to \infty$ if and only if $\Delta I_Y(n) > 0$ for all $n \in \mathcal{N}_{\ell-1}(\mathcal{T}_\mathcal{W})$.

Proof: See the unabridged version [26]. ■

Theorem 4.3 in conjunction with Proposition 4.4 guarantee that the path-cost sequence $\{\hat{J}_\varepsilon^\lambda(\hat{\pi}_{\beta_i}^*;\beta_i)\}_{i=1}^N$ monotonically decreases and converges to $J_\varepsilon^\lambda(\pi^*)$ as $\beta_N \to \infty$. We now present a number of other properties of our problem, for which the following fact is useful.

[3]See [25] for more information.

Fact 4.5: Let $u \in \{0, \ldots, \ell\}$, $\varepsilon \in [0,1]$ and $n \in r^{-1}(\{u\})$.[4] Then $V_\varepsilon^\lambda(n) = \frac{1}{2^{dr(n)}} \sum_{n' \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}(n)})} V_\varepsilon^\lambda(n')$.

Proof: See the unabridged version [26]. ∎

Proposition 4.6: Let $\varepsilon \in [0,1]$ and $\beta > 0$. Then $\hat{J}_\varepsilon^\lambda(\hat{\pi};\beta) < M_\varepsilon^\lambda$ if and only if $\hat{\pi}$ is a ε-feasible abstract path.

Proof: The proof is presented in Appendix B. ∎

Corollary 4.7: Let $\varepsilon \in [0,1]$. Then $J_\varepsilon^\lambda(\pi) < M_\varepsilon^\lambda$ if and only if π is a ε-feasible finest resolution path.

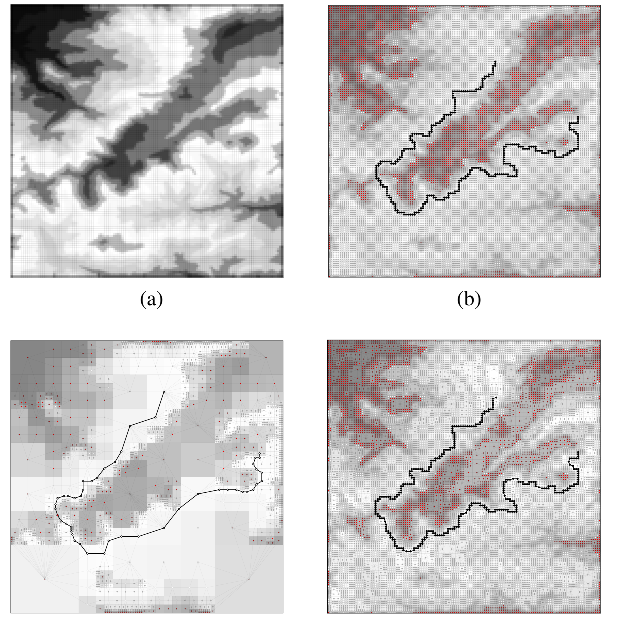
Proof: Identical to the proof of Proposition 4.6. ∎

The utility of Proposition 4.6 and Corollary 4.7 is that they provide conditions for quickly determining the feasibility of a path from knowledge of only the objective function value $\hat{J}_\varepsilon^\lambda(\hat{\pi};\beta)$ (or $J_\varepsilon^\lambda(\pi)$). If the search terminates before an ε-feasible path has been found, the agent is provided with the most recent solution, which is guaranteed to be the least infeasible path available in the current tree. Furthermore, as a result of Theorem 4.3 and Proposition 4.6, if an AP $\hat{\pi}_{\beta_j}^*$ is ε-feasible for some $j \in \{1, \ldots, N\}$, then all AP in the sequence $\{\hat{\pi}_{\beta_i}^*\}_{i=j}^N$ are also ε-feasible. This ensures that the autonomous agent can never discover that a feasible path becomes infeasible with further refinement of the environment. An illustration is provided in Fig. 2. To conclude this section, we present the following proposition.

Proposition 4.8: Let $\varepsilon \in [0,1]$ and $n \in \mathcal{N}_{\text{int}}(\mathcal{T}_\mathcal{W})$. Then $V_\varepsilon^\lambda(n) > \lambda_1 + \varepsilon\lambda_2$ if and only if $\mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}(n)}) \cap \mathcal{P}_\varepsilon^c \neq \varnothing$.

Proof: The proof is presented in Appendix C. ∎

Proposition 4.8 allows an autonomous agent to quickly identify which leafs of the tree $\mathcal{T} \in \mathcal{T}^\mathcal{Q}$ are considered to be ε-obstacles and, consequently, which vertices in $\mathcal{G}(\mathcal{T})$ to avoid, if possible. Next, we present a numerical example to demonstrate the utility of our approach.



(a) (b) (c) (d)

Fig. 3. 128×128 environment ($|\Omega_X| = 16384$) with graph abstraction and path examples for $\varepsilon = 0.5$. Cost parameters are $\lambda_1 = 0.001$ and $\lambda_2 = 1$. Shading of grey scales with probability of occupancy. Red vertices are considered ε-obstacles as determined by Proposition 4.8. (a) original environment, (b) example FRP, (c) example AP and graph for $\beta = 55$ $(\%|\Omega_X| = 8.3\%)$, (d) example AP and graph for $\beta = 1 \times 10^6$ $(\%|\Omega_X| = 83.4\%)$.

V. NUMERICAL EXAMPLE

We consider the world $\mathcal{W}$ to be given by the 128×128 OG shown in Fig. 3a. The OG representation provides information regarding the conditional distribution $p(y|x)$, whereby we then define the joint distribution $p(x,y) = p(y|x)p(x)$ with $p(x) = 1/|\mathcal{N}_{\text{leaf}}(\mathcal{T}_\mathcal{W})|$ for all $x \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_\mathcal{W})$.[5] By utilizing the uniform distribution $p(x)$, we encode that the autonomous agent is equally likely to occupy any cell $x \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_\mathcal{W})$ and will result in the IB method refining the environment in a region-agnostic manner [25]. The joint distribution $p(x,y)$, along with a sequence of strictly increasing positive values of $\{\beta_i\}_{i=1}^N$, are provided to the IB abstraction framework of Section IV-A to obtain the sequence of trees $\{\mathcal{T}_{\beta_i}\}_{i=1}^N$ along with the corresponding $\{\mathcal{G}(\mathcal{T}_{\beta_i})\}_{i=1}^N$. Given a start and goal location, the path planning problem (10) is solved on each of the trees $\{\mathcal{T}_{\beta_i}\}_{i=1}^N$ to obtain $\{\hat{\pi}_{\beta_i}^*\}_{i=1}^N$. Examples of obtained abstract paths are shown in Figs. 3c – 3d with an FRP shown in Fig. 3b.

In Fig. 4, we show the average path-cost ratio when the conditions of Proposition 4.4 are satisfied. To generate the average results, we consider a sequence $\{\beta_i\}_{i=1}^N$ and sample 200 pairs of start-goal points on the finest resolution. A sequence of reduced graphs $\{\mathcal{G}(\mathcal{T}_{\beta_i})\}_{i=1}^N$ is then created, which are employed to obtain N abstract paths for each sampled start-goal pair. This processes furnishes a dataset of compression and path-cost values for each start and goal pair sampled, which forms the basis of the averaging results.

From Fig. 4, we see that, on average, roughly 15% to 18% of the vertices of $\mathcal{G}(\mathcal{T}_\mathcal{W})$ are required in order to obtain a feasible path in the environment. A reduced graph containing only 15% to 18% of the vertices of $\mathcal{G}(\mathcal{T}_\mathcal{W})$ substantially reduces the computational effort required to find feasible solutions. Furthermore, Fig. 4 shows that the path-cost monotonically decreases with increased resolution. The decreasing nature of the cost in Fig. 4 is expected, since Theorem 4.3 guarantees that the path-cost between *any* two points monotonically decreases as a function of resolution (or, equivalently, increased $\beta > 0$). We also observe that the average path-cost ratio converges to 1, corroborating the conditions for convergence set forth by Proposition 4.4. Lastly, note that by utilizing a representation with approximately 70% of the nodes in $\mathcal{T}_\mathcal{W}$ results, on average, in an abstract path $\hat{\pi}_{\beta_i}^*$ for which $\hat{J}_\varepsilon^\lambda(\hat{\pi}_{\beta_i}^*;\beta_i)$ is within 30% of $J_\varepsilon^\lambda(\pi^*)$. Next, we discuss how elements of our framework relate to bounded-rational decision making and anytime algorithms.

VI. DISCUSSION

The role of $\beta > 0$ in our framework can be interpreted similarly to its role in other approaches for resource-constrained and bounded-rational decision making. Previous works [17],

[4] Notice that $r^{-1}(\{u\}) = \mathcal{N}_{\ell-u}(\mathcal{T}_\mathcal{W})$ for any $u \in \{1, \ldots, \ell\}$.

[5] While we assume for the numerical example that $p(x)$ is uniform, any valid $p(x)$ is allowable.

[19], [20] have considered information-constrained MDPs by adding KL-divergence constraints to traditional MDP problems. The KL-constraints limit to what extent an optimal policy is permitted to differ from a default policy provided to the agent beforehand, where $\beta > 0$ serves as a parameter to weight the relative importance of maximizing expected reward and minimizing deviation from the default policy. These studies argue that $\beta > 0$ parameterizes a spectrum of agents, where optimal policies for rational agents are recovered as $\beta \to \infty$ and for resource-limited agents as $\beta \to 0$. Similarly, in our work, β trades the complexity of the search with the value of the resulting path, where larger values of β lead to lower path costs at the penalty of increased search complexity. In actual agents, the value of $\beta > 0$ should be chosen so that the worst-case computational cost of obtaining a path as well as the memory requirements for storing the environment representation do not exceed the on-board resources available to the system.

A number of other approaches aim to decrease planning complexity by leveraging environment abstractions [4], [5], [8]–[15]. While these works simplify the planning problem, they provide no guarantees that the solution improve with increased planning time or resolution. A connection between time $t > 0$ and plan quality can be established in our method by considering a strictly increasing mapping $\Gamma(t) = \beta$. In this way, the IB abstractions as well as the planning problem (10) become time-dependent, whereby the improvement of the objective value with time is established by Theorem 4.3. This is akin to the interplay of plan quality and deliberation time as suggested by anytime algorithms [3].

To quantify the computational cost of our approach, we note that the complexity of the Q-tree search algorithm is $\mathcal{O}(|\mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}})|)$ [25]. Thus, the worst-case complexity of executing Q-tree search and planning is $\mathcal{O}(|\mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}})|) + \mathcal{O}(|E(\mathcal{T})| + |\mathcal{V}(\mathcal{T})| \log|\mathcal{V}(\mathcal{T})|)$ for any $\mathcal{T} \in \mathcal{T}^{\mathcal{Q}}$, as compared to $\mathcal{O}(|E(\mathcal{T}_{\mathcal{W}})| + |\mathcal{V}(\mathcal{T}_{\mathcal{W}})| \log|\mathcal{V}(\mathcal{T}_{\mathcal{W}})|)$ for an FRP. We provide experimental timing results for various two-dimensional grid sizes in Fig. 5. Comparing Fig. 4 and Fig. 5, we see that, for the example considered in Section V, a feasible path can be found faster than executing a search on the finest resolution. Furthermore, the computational benefits of our approach increase as the size of the finest resolution space grows. The computational savings come at the cost of a diminished performance due to the intrinsic need to trade computational complexity and path optimality, as discussed throughout this paper. Lastly, it is important to view Fig. 5 keeping in mind: (i) the results assume the worst-case scenario that the abstractions must be generated from scratch and used to produce a single abstract path and (ii) the environment is two-dimensional.

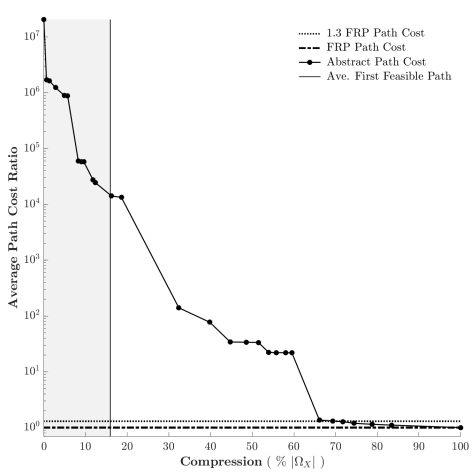


Fig. 4. Logarithmic (base 10) value of average $\hat{J}_{\varepsilon}^{\lambda}(\hat{\pi}^*_{\beta_i};\beta_i)/J_{\varepsilon}^{\lambda}(\pi^*)$ versus compression for $\varepsilon = 0.5$, $\lambda_1 = 0.001$ and $\lambda_2 = 1$. Note that $|\Omega_X| = 16384$. Average values computed over 200 randomly sampled start and goal locations. Moving along the curve to the right is done by increasing β. Average first feasible path line represents average compression at which first guaranteed feasible path in the abstracted environment is found.

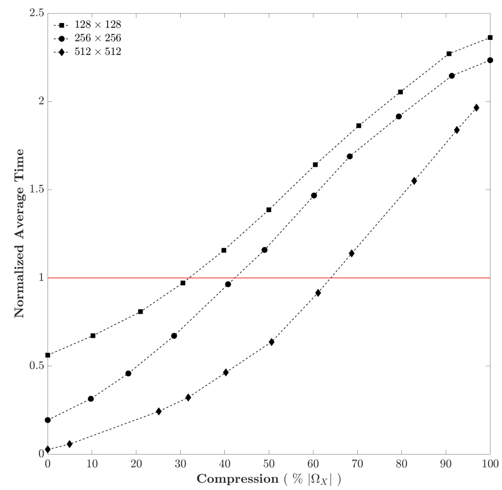


Fig. 5. Average computation time vs. environment compression for two-dimensional grid sizes of 128×128, 256×256 and 512×512. By increasing β one moves along the curves to the right. Timing results are obtained by averaging over 100 randomly selected start and goal locations. The y-axis represents average (normalized) time obtained by averaging the time for (i) information computation, (ii) computing Q-values, (iii) running Q-tree search and (iv) employing Dijkstra graph-search to obtain an abstract path and normalizing by the average time to run Dijkstra on the finest-resolution map. Timing results assume the worst-case scenario that abstractions are generated from scratch for each value of β (or compression level) on a computer with a 2.9 GHz Intel i5 CPU with 8 GB of RAM running MATLAB.

To conclude, recall that for a given value of $\beta > 0$, the abstraction returned by Q-tree search will be a tree that retains the maximum amount of information regarding the relevant variable Y. This process is subject to the choice of the relevant variable, which we assumed to be the cell occupancy. Importantly, our framework holds for other choices of the relevant variable. An investigation into the selection of the relevant random variable and its implications is left for future work.

VII. CONCLUSIONS

In this paper, we have shown how a path-planning problem can be systematically simplified by employing multi-resolution tree abstractions generated by an information-theoretic framework. The abstractions are not provided a priori and can be tailored to agent resource constraints. A number of theoretical results were presented that establish formal connections between the path quality, graph-search complexity, and the information contained in the reduced graphs. To corroborate our theoretical findings, a non-trivial numerical example was presented together with a discussion analyzing the interpretation of our framework in the context of bounded-rational decision making and anytime algorithms.

REFERENCES

[1] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, 4th ed. Athena Scientific, 2012, vol. 2.

[2] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. MIT Press, 2016.

[3] T. Dean and M. Boddy, "An analysis of time-dependent planning," in *AAAI*, St. Paul, MN, USA, August 21-26 1988, pp. 49–54.

[4] S. Kambhampati and L. S. Davis, "Multiresolution path planning for mobile robots," *IEEE Journal of Robotics and Automation*, vol. RA-2, no. 3, pp. 135–145, September 1986.

[5] P. Tsiotras, D. Jung, and E. Bakolas, "Multiresolution hierarchical path-planning for small UAVs using wavelet decompositions," *Journal of Intelligent and Robotic Systems*, vol. 66, no. 4, pp. 505–522, June 2012.

[6] S. Behnke, "Local multiresolution path planning," in *RoboCup 2003: Robot Soccer World Cup VII*. Springer, 2003, pp. 332–343.

[7] Y. Lu, X. Huo, and P. Tsiotras, "A beamlet-based graph structure for path planning using multiscale information," *IEEE Transactions on Automatic Control*, vol. 57, no. 5, pp. 1166–1178, 2012.

[8] P. Tsiotras and E. Bakolas, "A hierarchical on-line path planning scheme using wavelets," in *European Control Conference*, Kos, GR, July 2-5 2007, pp. 2806–2812.

[9] R. V. Cowlagi and P. Tsiotras, "Multiresolution path planning with wavelets: A local replanning approach," in *American Control Conference*, Seattle, WA, USA, June 11-13 2008, pp. 1220–1225.

[10] ——, "Multi-resolution path planning: Theoretical analysis, efficient implementation, and extensions to dynamic environments," in *IEEE Conference on Decision and Control*, Atlanta, GA, USA, December 15-17 2010, pp. 1384–1390.

[11] R. V. Cowlagi, "Hierarchical motion planning for autonomous aerial and terrestrial vehicles," Ph.D. dissertation, Georgia Institute of Technology, 2011.

[12] 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, October 2012.

[13] ——, "Hierarchical motion planning with kinodynamic feasibility guarantees," *IEEE Transactions on Robotics*, vol. 28, no. 2, pp. 379–395, April 2012.

[14] F. Hauer, A. Kundu, J. M. Rehg, and P. Tsiotras, "Multi-scale perception and path planning on probabilistic obstacle maps," in *IEEE International Conference on Robotics and Automation*, Seattle, WA, USA, May 26-30 2015, pp. 4210–4215.

[15] F. Hauer and P. Tsiotras, "Reduced complexity multi-scale path-planning on probabilitic maps," in *IEEE Conference on Robotics and Automation*, Stockholm, SE, May 16-21 2016, pp. 83–88.

[16] G. K. Kraetzschmar, G. P. Gassull, and K. Uhl, "Probabilistic quadtrees for variable-resolution mapping of large environments," *IFAC Proceedings Volumes*, vol. 37, no. 8, pp. 675–680, July 2004.

[17] N. Tishby and D. Polani, "Information theory of decisions and actions," in *Preception-Action Cycle*. Springer, 2010, pp. 601–636.

[18] P. A. Ortega and D. A. Braun, "Information, utility and bounded rationality," in *International Conference on Artifical General Intelligence*, Mountain View, CA, USA, August 3-6 2011, pp. 269–274.

[19] T. Genewein, F. Leibfried, J. Grau-Moya, and D. A. Braun, "Bounded rationality, abstraction, and hierarchical decision-making: An information-theoretic optimality principle," *Frontiers in Robotics and AI*, vol. 2, no. 27, November 2015.

[20] D. T. Larsson, D. Braun, and P. Tsiotras, "Hierarchical state abstractions for decision-making problems with computational constraints," in *IEEE Conference on Decision and Control*, Melbourne, AU, December 12-15 2017, pp. 1138–1143.

[21] J. Rubin, O. Shamir, and N. Tishby, "Trading value and information in MDPs," in *Decision Making with Imperfect Decision Makers*. Springer, 2012, pp. 57–74.

[22] P. Tsiotras, "Bounded rationality in learning, perception, decision-making, and stochastic games," in *Handbook of Reinforcement Learning and Control*. Springer, 2021, pp. 491–523.

[23] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. John Wiley & Sons, 2006.

[24] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," in *Allerton Conference on Communication, Control and Computing*, Monticello, IL, USA, September 22-24 1999, pp. 368–377.

[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, December 2020.

[26] ——, "Information-theoretic abstractions for planning in agents with computational constraints," *arXiv:2005.09611v2*, 2021.

APPENDIX A
PROOF OF THEOREM 4.3

The proof is given in two parts. We first present and provide a proof of the following lemma before providing the proof of Theorem 4.3.

Lemma A.1: Let $n \in \mathcal{N}_{\text{int}}(\mathcal{T}_{\mathcal{W}})$. Then $2^{dr(n)} V_\varepsilon^\lambda(n) \geq \sum_{n' \in \mathcal{C}(n) \setminus \mathcal{S}} 2^{dr(n')} V_\varepsilon^\lambda(n')$ for all $\mathcal{S} \subseteq \mathcal{C}(n)$.

Proof: The proof is given by contradiction. Assume that $2^{dr(n)} V_\varepsilon^\lambda(n) < \sum_{n' \in \mathcal{C}(n) \setminus \mathcal{S}} 2^{dr(n')} V_\varepsilon^\lambda(n')$ for some $\mathcal{S} \subseteq \mathcal{C}(n)$. This means,

$$2^{dr(n')} \left(2^d V_\varepsilon^\lambda(n) - \sum_{n' \in \mathcal{C}(n) \setminus \mathcal{S}} V_\varepsilon^\lambda(n') \right) < 0,$$

as $n' \in \mathcal{C}(n)$ and since $2^{dr(n')} > 0$, $V_\varepsilon^\lambda(n) < 2^{-d} \sum_{n' \in \mathcal{C}(n) \setminus \mathcal{S}} V_\varepsilon^\lambda(n')$. However, $0 \leq V_\varepsilon^\lambda(n')$ for all $n' \in \mathcal{C}(n)$, and so

$$V_\varepsilon^\lambda(n) < 2^{-d} \sum_{n' \in \mathcal{C}(n) \setminus \mathcal{S}} V_\varepsilon^\lambda(n') \leq 2^{-d} \sum_{n' \in \mathcal{C}(n)} V_\varepsilon^\lambda(n').$$

Since $2^{-d} \sum_{n' \in \mathcal{C}(n)} V_\varepsilon^\lambda(n') = V_\varepsilon^\lambda(n)$, the above implies $V_\varepsilon^\lambda(n) < V_\varepsilon^\lambda(n)$, a contradiction. ■

We now prove Theorem 4.3.

Proof: The proof is given by construction. There are two cases to consider: $\hat{\pi}_{\beta_1}^* \cap \{n\} = \varnothing$, and $\hat{\pi}_{\beta_1}^* \cap \{n\} \neq \varnothing$. First consider the case $\hat{\pi}_{\beta_1}^* \cap \{n\} = \varnothing$. It follows,

$$\hat{\pi}_{\beta_1}^* \subseteq \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\beta_1}) \cap \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\beta_2}) \subset \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\beta_2}).$$

Take $\hat{\pi}_{\beta_2} = \hat{\pi}_{\beta_1}^*$ and thus $\hat{J}_\varepsilon^\lambda(\hat{\pi}_{\beta_1}^*; \beta_1) \geq \hat{J}_\varepsilon^\lambda(\hat{\pi}_{\beta_2}; \beta_2)$. Now consider $\hat{\pi}_{\beta_1}^* \cap \{n\} \neq \varnothing$. Without loss of generality, $\hat{\pi}_{\beta_1}^* = \{z_0, \ldots, z_{i-1}, z_i, z_{i+1}, \ldots, z_R\} \subseteq \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\beta_1})$ is an abstract path where $z_i = n$. As the node n is expanded, we re-route the path through the children of n. Consider $\hat{\pi}_{\beta_2} = \{z_0, \ldots, z_{i-1}, z'_{i_1}, \ldots, z'_{i_u}, z_{i+1}, \ldots, z_R\}$, where $\{z'_{i_1}, \ldots, z'_{i_u}\} \subseteq \mathcal{C}(n)$ is a sequence of nodes so as to render $\hat{\pi}_{\beta_2}$ an abstract path. Notice that $\hat{\pi}_{\beta_2} \subseteq \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\beta_2})$, and

$$\hat{J}_\varepsilon^\lambda(\hat{\pi}_{\beta_1}^*; \beta_1) - \hat{J}_\varepsilon^\lambda(\hat{\pi}_{\beta_2}; \beta_2) = 2^{dr(n)} V_\varepsilon^\lambda(n) - \sum_{n' \in \{z'_{i_1}, \ldots, z'_{i_u}\}} 2^{dr(n')} V_\varepsilon^\lambda(n').$$

Since $\{z'_{i_1}, \ldots, z'_{i_u}\} \subseteq \mathcal{C}(n)$, it follows from Lemma A.1 that

$$2^{dr(n)} V_\varepsilon^\lambda(n) - \sum_{n' \in \{z'_{i_1}, \ldots, z'_{i_u}\}} 2^{dr(n')} V_\varepsilon^\lambda(n') \geq 0,$$

and so $\hat{J}_\varepsilon^\lambda(\hat{\pi}_{\beta_1}^*; \beta_1) \geq \hat{J}_\varepsilon^\lambda(\hat{\pi}_{\beta_2}; \beta_2)$. ■

APPENDIX B
PROOF OF PROPOSITION 4.6

Proof: (Neccessity) The proof is given by contradiction. Assume there exists $\beta > 0$ such that $\hat{J}_\varepsilon^\lambda(\hat{\pi};\beta) \geq M_\varepsilon^\lambda$ for some ε-AP $\hat{\pi}$. From the definition of $\hat{J}_\varepsilon^\lambda$ and Fact 4.5, we have

$$M_\varepsilon^\lambda \leq \sum_{z\in\hat{\pi}} 2^{dr(z)} V_\varepsilon^\lambda(z) = \sum_{z\in\hat{\pi}} \sum_{n'\in\mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}(z)})} V_\varepsilon^\lambda(n'),$$
$$= \sum_{n\in \bigcup_{z\in\hat{\pi}} \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}(z)})} V_\varepsilon^\lambda(n).$$

Now, from the definition of V_ε^λ, the non-negativity of the cost c_ε^λ, and that $\hat{\pi}$ is ε-feasible, we obtain

$$\sum_{n\in \bigcup_{z\in\hat{\pi}} \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}(z)})} V_\varepsilon^\lambda(n) \leq \sum_{x\in\mathcal{P}_\varepsilon} \lambda_1 + \lambda_2 p(y=1|x),$$
$$\leq \sum_{x\in\mathcal{N}_{\text{leaf}}(\mathcal{T}_\mathcal{W})} \lambda_1 + \lambda_2\varepsilon,$$
$$< 2^{d\ell}\left(\lambda_1 + \lambda_2\varepsilon\right) + \gamma = M_\varepsilon^\lambda.$$

The above result implies

$$M_\varepsilon^\lambda \leq \sum_{z\in\hat{\pi}} 2^{dr(z)} V_\varepsilon^\lambda(z) < M_\varepsilon^\lambda,$$

which is a contradiction.

(Sufficiency) Now assume $J_\varepsilon^\lambda(\hat{\pi};\beta) < M_\varepsilon^\lambda$ for some AP $\hat{\pi}$ and define the sets

$$\mathcal{A}_{\hat{\pi}} = \bigcup_{z\in\hat{\pi}} \mathcal{N}_{\text{leaf}}\left(\mathcal{T}_{\mathcal{W}(z)}\right) \cap \mathcal{P}_\varepsilon,$$
$$\mathcal{B}_{\hat{\pi}} = \bigcup_{z\in\hat{\pi}} \mathcal{N}_{\text{leaf}}\left(\mathcal{T}_{\mathcal{W}(z)}\right) \cap \mathcal{P}_\varepsilon^c.$$

Then, by Fact 4.5 and the definition of V_ε^λ, we have

$$\hat{J}_\varepsilon^\lambda\left(\hat{\pi};\beta\right) = \sum_{z\in\mathcal{A}_{\hat{\pi}}} V_\varepsilon^\lambda(z) + |\mathcal{B}_{\hat{\pi}}| M_\varepsilon^\lambda.$$

Note that $\mathcal{A}_{\hat{\pi}} \subseteq \mathcal{P}_\varepsilon$ and hence

$$0 \leq \sum_{z\in\mathcal{A}_{\hat{\pi}}} V_\varepsilon^\lambda(z) \leq 2^{d\ell}(\lambda_1 + \lambda_2\varepsilon) < M_\varepsilon^\lambda.$$

Thus, if $\hat{J}_\varepsilon^\lambda\left(\hat{\pi};\beta\right) < M_\varepsilon^\lambda$ then

$$\hat{J}_\varepsilon^\lambda\left(\hat{\pi};\beta\right) = \sum_{z\in\mathcal{A}_{\hat{\pi}}} V_\varepsilon^\lambda(z) + |\mathcal{B}_{\hat{\pi}}| M_\varepsilon^\lambda < M_\varepsilon^\lambda,$$

which requires $|\mathcal{B}_{\hat{\pi}}| = 0$. Therefore, if $\hat{J}_\varepsilon^\lambda\left(\hat{\pi};\beta\right) < M_\varepsilon^\lambda$ then $\mathcal{B}_{\hat{\pi}} = \varnothing$. Hence $\bigcup_{z\in\hat{\pi}} \mathcal{N}_{\text{leaf}}\left(\mathcal{T}_{\mathcal{W}(z)}\right) \subseteq \mathcal{P}_\varepsilon$, which implies $\hat{\pi}$ is an ε-AP. ∎

APPENDIX C
PROOF OF PROPOSITION 4.8

Proof: (Neccessity) Let $n \in \mathcal{N}_{\text{int}}\left(\mathcal{T}_\mathcal{W}\right)$ and assume $\mathcal{N}_{\text{leaf}}\left(\mathcal{T}_{\mathcal{W}(n)}\right) \cap \mathcal{P}_\varepsilon^c \neq \varnothing$. Define the sets

$$\mathcal{A}_n \triangleq \mathcal{N}_{\text{leaf}}\left(\mathcal{T}_{\mathcal{W}(n)}\right) \cap \mathcal{P}_\varepsilon,$$
$$\mathcal{B}_n \triangleq \mathcal{N}_{\text{leaf}}\left(\mathcal{T}_{\mathcal{W}(n)}\right) \cap \mathcal{P}_\varepsilon^c,$$

and note that, by assumption, $|\mathcal{B}_n| \neq 0$. From Fact 4.5, we have that

$$V_\varepsilon^\lambda(n) = \frac{1}{2^{dr(n)}}\left[\sum_{n'\in\mathcal{A}_n} V_\varepsilon^\lambda(n') + \sum_{n'\in\mathcal{B}_n} V_\varepsilon^\lambda(n')\right],$$

and since the function V_ε^λ is non-negative,

$$V_\varepsilon^\lambda(n) = \frac{1}{2^{dr(n)}}\left[\sum_{n'\in\mathcal{A}_n} V_\varepsilon^\lambda(n') + |\mathcal{B}_n| M_\varepsilon^\lambda\right],$$
$$\geq \frac{1}{2^{dr(n)}} |\mathcal{B}_n| M_\varepsilon^\lambda.$$

Now, as $|\mathcal{B}_n| \neq 0$,

$$V_\varepsilon^\lambda(n) \geq \frac{1}{2^{dr(n)}} |\mathcal{B}_n| M_\varepsilon^\lambda \geq \frac{1}{2^{dr(n)}} M_\varepsilon^\lambda,$$

and, hence, from the definition of M_ε^λ, we obtain

$$V_\varepsilon^\lambda(n) \geq \frac{1}{2^{dr(n)}} M_\varepsilon^\lambda > 2^{d(\ell - r(n))}(\lambda_1 + \lambda_2\varepsilon). \tag{11}$$

Since $2^{d(\ell - r(n))} \geq 1$, relation (11) implies $V_\varepsilon^\lambda(n) > \lambda_1 + \lambda_2\varepsilon$.

(Sufficiency) Now, let $n \in \mathcal{N}_{\text{int}}\left(\mathcal{T}_\mathcal{W}\right)$ and assume $V_\varepsilon^\lambda(n) > \lambda_1 + \lambda_2\varepsilon$. Then, from the definition of V_ε^λ, we have

$$V_\varepsilon^\lambda(n) = \frac{1}{2^{dr(n)}} \sum_{n'\in\mathcal{A}_n} V_\varepsilon^\lambda(n') + \frac{2^{d\ell}(\lambda_1 + \lambda_2\varepsilon) + \gamma}{2^{dr(n)}} |\mathcal{B}_n|,$$

and,

$$0 \leq \frac{1}{2^{dr(n)}} \sum_{n'\in\mathcal{A}_n} V_\varepsilon^\lambda(n') \leq \frac{1}{2^{dr(n)}}(\lambda_1 + \varepsilon\lambda_2)|\mathcal{A}_n| \leq \lambda_1 + \varepsilon\lambda_2,$$

which follows since $|\mathcal{A}_n| \leq 2^{dr(n)}$. Consequently,

$$0 \leq \frac{1}{2^{dr(n)}} \sum_{n'\in\mathcal{A}_n} V_\varepsilon^\lambda(n') \leq \lambda_1 + \varepsilon\lambda_2.$$

Therefore, if $V_\varepsilon^\lambda(n) > \lambda_1 + \lambda_2\varepsilon$ then

$$\frac{1}{2^{dr(n)}} \sum_{n'\in\mathcal{A}_n} V_\varepsilon^\lambda(n') + \frac{2^{d\ell}(\lambda_1 + \lambda_2\varepsilon) + \gamma}{2^{dr(n)}} |\mathcal{B}_n| > \lambda_1 + \lambda_2\varepsilon,$$

which requires $|\mathcal{B}_n| > 0$. This means $\mathcal{N}_{\text{leaf}}\left(\mathcal{T}_{\mathcal{W}(n)}\right) \cap \mathcal{P}_\varepsilon^c \neq \varnothing$. ∎