

A Linear Programming Approach for Resource-Aware Information-Theoretic Tree Abstractions

Daniel T. Larsson, Dipankar Maity, and Panagiotis Tsiotras

Abstract In this chapter, an integer linear programming formulation for the problem of obtaining task-relevant, multi-resolution, environment abstractions for resource-constrained autonomous agents is presented. The formulation leverages concepts from information-theoretic signal compression, specifically, the information bottleneck (IB) method, to pose an abstraction problem as an optimal encoder search over the space of multi-resolution trees. The abstractions emerge in a task-relevant manner as a function of agent information-processing constraints. We detail our formulation, and show how hierarchical tree structures, signal encoders, and information-theoretic methods for signal compression can be unified under a common theme. A discussion delineating the benefits and drawbacks of our formulation is presented, as well as a detailed explanation how our approach can be interpreted within the context of generating abstractions for resource-constrained autonomous systems. It is shown that the resulting information-theoretic abstraction problem over the space of multi-resolution trees can be formulated as a integer linear programming (ILP) problem. We demonstrate the approach on a number of examples, and provide a discussion detailing the differences of the proposed framework compared to existing methods. Lastly, we consider a linear program relaxation of the ILP problem, thereby demonstrating that multi-resolution information-theoretic tree abstractions can be obtained by solving a convex program.

Daniel T. Larsson
Georgia Institute of Technology, Atlanta, GA, USA, e-mail: daniel.larsson@gatech.edu

Dipankar Maity
University of North Carolina at Charlotte, Charlotte, NC, USA, e-mail: dmaity@uncc.edu

Panagiotis Tsiotras
Georgia Institute of Technology, Atlanta, GA, USA, e-mail: tsiotras@gatech.edu

1 Introduction

The identification and preservation of task-relevant information has long been considered central to the development of intelligent systems [28, 38]. However, the design of task-relevant abstractions in autonomous systems has been traditionally handled by system engineers, who provide specific expert knowledge of the problem domain by specifying features that are important for the specific task at hand [38, 28, 19]. The use of abstractions in autonomous systems is motivated by the need to systematically reduce the complexity of problem-solving, such as planning and decision making, by removing details that are considered irrelevant for the specific task [19, 38]. As a result, the abstraction process allows autonomous agents to simplify the problem by focusing on aspects of the domain deemed to be task-relevant, while removing (or rather, ignoring) those details that are irrelevant [19, 38]. Despite the fact that the ability to form task-relevant abstractions is considered central to the development of intelligent systems, many existing frameworks that leverage the power of hierarchical abstractions rely heavily on user-provided rules to form reduced representations of the problem domain [38].

Consider, for example, the task of planning a collision-free path in an autonomous driving scenario. Assuming that a graph-based planning algorithm is employed to generate a path, such as A* or Dijkstra, an autonomous system must balance environment complexity (e.g., resolution) with the on-board resources available to execute the graph search in order to complete the task successfully in the presence of computational constraints. To illustrate the trade-off between environment resolution and search complexity, consider the two environments shown in Fig. 1. For each of the two abstractions shown in Fig. 1, one may construct a corresponding reduced graph whose vertices are the grid cell centers of the multi-resolution depiction and the corresponding edges encode cell connectivity. As a result, executing graph search algorithms such as A* and Dijkstra on reduced graphs with fewer vertices, that is, those that correspond to lower-resolution representations, have better worst-case time complexity as compared with performing the graph-search on reduced graphs that are derived from higher-resolution depictions that contain a greater number of vertices [24]. However, designing such abstractions is not trivial in that it requires the autonomous system to balance the model (environment) complexity with retaining as much information in regards to what is relevant to the task (in the example presented in Fig. 1 this may be the occupancy probability of the finest-resolution grid cells).

Generating suitable compressed representations requires the formulation of a mathematical optimization problem that specifies which aspects of the problem are important (or, equivalently, relevant) and which can be neglected, or ignored, when forming a latent (e.g., compressed) representation of the problem. Such a mathematical formulation requires the specification of two components. The first component is a measure that captures the reduced model complexity, or rather, the degree of achieved compression compared to the original representation. The second component is a measure that quantifies the quality of the abstract representation. The terms driving compression and representation quality are in conflict with one another, in the sense that a higher-quality representation comes at the cost of a more complex

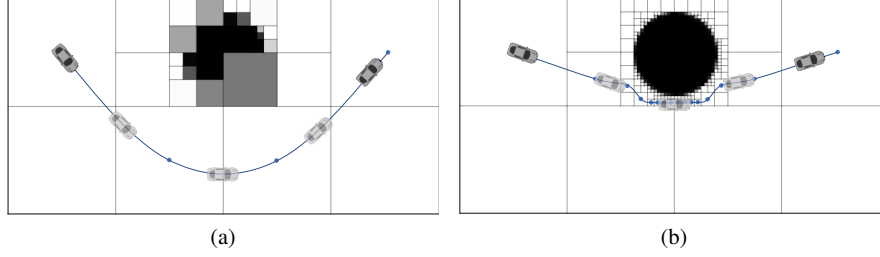


Fig. 1: Illustrative application of abstractions to the task of autonomous driving. The darker cells represent impassable obstacles in the environment.

abstract space, and vice-versa. The design of abstractions is then formally posed as an optimization problem where the goal is to maximize compression subject to constraints on the reduced model quality.

The construction of the abstract, or compressed, representations of the environment is then subject to the definition of the measures that capture model complexity and quality. By changing the measures that capture these two quantities, we arrive at different abstraction frameworks. It should be noted that the specification of measures that capture model complexity and quality are often not trivial for a given task, but are vital to the construction of a good abstract space, as they implicitly define what details should be retained, and which can be removed, when forming a compressed representation. To illustrate this trade-off, consider the framework of rate-distortion theory, which was developed by information theorists in order to encode signals for transmission over noisy communication channels [6]. Rate-distortion theory formulates an optimization problem where the goal is to maximize compression subject to an upper bound on the expected distortion (i.e., bound the desired model quality). Compressed representations that achieve lower values of expected distortion are considered to be of higher quality. In its formulation, the rate-distortion framework measures the model complexity by utilizing Shannon’s mutual information (MI) between the original and compressed signals, and quantifies the model quality through a non-negative distortion function. The use of mutual information as a measure of the model complexity is rooted in its use in communication systems, where it quantifies the rate of codes required to describe a signal to a given level of detail [6]. Crucially, however, rate-distortion theory does not specify the distortion measure, nor does it provide with any guidance how one should be constructed, assuming instead that the function is provided by the system designer [34]. The selection of the distortion function is crucial, as it defines what details of the original signal should be retained in order to achieve low values of expected distortion (high model quality). Specifying a distortion function for a given clustering task is difficult and non-intuitive, as it may not be clear from the problem formulation which features of the data should be preserved to accomplish a desired objective [34].

In order to combat some of the difficulties surrounding the specification of a distortion function, the work of [34] introduced the Information-Bottleneck (IB) method. In contrast to the approach employed by classical rate-distortion theory, the IB method does not rely on the specification of a distortion function, and instead introduces a third variable, referred to as the relevant random variable, that contains task-relevant information we wish to retain through the process of abstraction. Consequently, the IB formulation replaces the distortion measure with the mutual information between the relevant variable and compressed representation. Considering their similarities, it is unsurprising that the IB method has been informally viewed as a specific instance of the rate-distortion problem, where the distortion measure is taken to be that of the mutual information between the compressed and relevant random variables [13, 31, 34, 5]. By virtue of its formulation, the IB problem naturally captures the notion of designing abstractions that are maximally retentive regarding task-specific relevant information.

In the autonomous systems community, abstractions have seen widespread use as a way to reduce the complexity of decision-making and path-planning problems [35, 36, 16, 17, 10, 7, 9, 8]. For example, in the works of [35, 36, 16, 17, 10, 7, 9, 8], the authors create coarse-grained, multi-resolution, representations of an environment for path-planning to reduce the computational complexity of executing graph-search algorithms. The idea is that, by leveraging abstractions to create a reduced graph of the world containing fewer vertices than the full-resolution map, the execution time of off-the-shelf graph-search algorithms, such as A* and Dijkstra, can be improved at the cost of sub-optimal paths. In designing the abstractions, the works of [35, 36, 16, 17, 10, 7, 9, 8] consider the relevant information as the region nearest the autonomous agent, as autonomous agents do not generally need to represent environment information that is physically distant from the vehicle with high resolution, as it is not crucial for short-range obstacle avoidance and sensing. To strike a balance between (path) optimality and environment complexity the planning frameworks in [35, 36, 16, 17, 10, 7, 9, 8] sequentially re-solve the planning problem as the agent traverses the world. Crucially, the definition of information relevance employed by these works is somewhat arbitrary, and their reliance on the specification of tuning parameters that characterize the properties of the abstractions is a notable drawback.

Our goal in this chapter is to utilize the powerful approaches for signal compression to rigorously design task-relevant environment abstractions for autonomous systems by appealing to methods from information theory. We aim to develop frameworks for the emergence of environment abstractions in autonomous systems that are task-driven and require minimal domain knowledge or interaction with the system designer. To accomplish our goal, we propose to utilize concepts from information theory, specifically the IB principle, in order to formulate optimal encoder problems that give rise to task-relevant abstractions as a function of agent resource limitations. The proposed method aims to afford agents the ability to design abstractions autonomously as a function of the task, while considering system-level resource constraints in a principled and rigorous manner.

2 Notation

Throughout this chapter, let the set of real numbers be denoted $\mathbb{R}$, and, for any strictly positive integer n , let $\mathbb{R}^n$ denote the n -dimensional Euclidean space. For any vector $\mathbf{x} \in \mathbb{R}^n$, $[\mathbf{x}]_i$ denotes the i^{th} element of $\mathbf{x}$ for $i \in \{1, \dots, n\}$. Furthermore, for any two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, the notation $\mathbf{x} \leq \mathbf{y}$ is to be understood component-wise; that is, $[\mathbf{x}]_i \leq [\mathbf{y}]_i$ for all $i \in \{1, \dots, n\}$. The expected value of the discrete random variable X distributed according to $p(x)$ is denoted as $\mathbb{E}(X) = \sum_x p(x)x$. We assume throughout that logarithms have base e .

3 Preliminaries

Our objective in this chapter is to develop a framework for the design of multi-resolution, task-relevant, tree abstractions of operating environments for resource-constrained autonomous systems. To accomplish our goal, we first briefly review the fundamentals of information-theoretic signal compression by considering two notable frameworks: rate-distortion theory and the information bottleneck (IB) method; the latter will form the basis for the remainder of this chapter. We then review hierarchical tree structures, and detail the connection between trees and signal encoders. Facilitating a connection between hierarchical structures and encoders allows us to utilize information-theoretic signal compression theory to formalize an abstraction problem over the space of multi-resolution trees.

3.1 Information-Theoretic Signal Compression

A number of frameworks have been developed by information theorists in order to compress signals for transmission across capacity-limited communication channels. Two notable such frameworks are the classical rate-distortion theory [6, 29], and that of the information-bottleneck method [34]. Both of these approaches consider the design of optimal encoders by modeling signals probabilistically, employing concepts from probability theory to formulate rigorous optimization problems. Therefore, in order to describe these frameworks in more detail, we require the introduction of a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, with finite sample space Ω , σ -algebra $\mathcal{F}$ and probability measure $\mathbb{P} : \mathcal{F} \rightarrow [0, 1]$. Let $X : \Omega \rightarrow \mathbb{R}$ be a random variable defined on the sample space Ω with distribution $p(x) = \mathbb{P}(\{\omega \in \Omega : X(\omega) = x\})$. Similarly, we define the random variables $T : \Omega \rightarrow \mathbb{R}$ and $Y : \Omega \rightarrow \mathbb{R}$ with analogous definitions for the distributions $p(t)$ and $p(y)$. We will give concrete meaning to the random variables X , T , and Y when discussing the compression frameworks below, but will regard them as general random variables for the time being.

Given a random variable X with associated distribution $p(x)$, the (*Shannon*) *entropy* of X is defined according to

$$H(X) = - \sum_x p(x) \log p(x). \quad (1)$$

The Shannon entropy is non-negative and quantifies the uncertainty in the outcome of the random variable X . More specifically, the Shannon entropy attains its maximum value when $p(x)$ assigns equal probability mass to each outcome of X , and where its minimal value of $H(X) = 0$ is realized if $p(x)$ assigns all probability mass to a single outcome (i.e., X is deterministic). Notice that the Shannon entropy does not depend on the outcomes of the random variable X , but rather only on its distribution $p(x)$. For this reason, we will at times abuse notation and write $H(p)$ to mean $H(p) = - \sum_x p(x) \log p(x)$. While the entropy quantifies the uncertainty in X , it may also be useful to have a means to measure the average uncertainty in X when given the outcome of another random variable, say T . To this end, the conditional entropy of the random variable X given T is defined as

$$H(X|T) = - \sum_{t,x} p(t,x) \log p(x|t). \quad (2)$$

If the random variables X and T are independent, such that knowing the outcome of T does not influence our knowledge of X , then $H(X|T) = H(X)$. Moreover, it can be shown that $H(X|T) \leq H(X)$, representing the intuitive notion that additional information cannot increase the uncertainty in a random variable [6].

A related information-theoretic quantity is that of the Kullback-Liebler (KL) divergence, which quantifies the difference between two distributions $p(x)$ and $\nu(x)$. Formally, the KL-divergence is defined as

$$D_{\text{KL}}(p(x), \nu(x)) = \sum_x p(x) \log \frac{p(x)}{\nu(x)}. \quad (3)$$

The KL-divergence is, however, not a metric, as it is not symmetric and does not satisfy the triangle inequality [6, 29]. It should be noted that despite these technical considerations, the KL-divergence is zero, $D_{\text{KL}}(p(x), \nu(x)) = 0$, if and only if $p(x) = \nu(x)$, and is strictly positive otherwise. The mutual information between two random variables X and T is defined in terms of the KL-divergence as

$$I(T; X) = D_{\text{KL}}(p(t,x), p(t)p(x)) = \sum_{t,x} p(t,x) \log \frac{p(t,x)}{p(t)p(x)}. \quad (4)$$

From the definition of KL-divergence, we see that $I(T; X) \geq 0$ and that $I(T; X) = 0$ if and only if $p(t,x) = p(t)p(x)$. Consequently, if $I(T; X) = 0$, it follows that the two random variables T and X are independent. The mutual information $I(T; X)$ can be expressed in terms of the Shannon entropy as

$$I(T; X) = H(T) - H(T|X) = H(X) - H(X|T). \quad (5)$$

Relation (5) shows that the mutual information is the *reduction in uncertainty* regarding the random variable T when X is known, and vice-versa [6]. Consequently,

if $I(X; T) = 0$ then there is no reduction in uncertainty of T when given X , corresponding to the case when T and X are independent; that is, $I(T; X) = 0$ means that T conveys no information regarding X . The above arguments can be equivalently interpreted by replacing X with T , as the mutual information is symmetric, i.e., $I(T; X) = I(X; T)$. Furthermore, equation (5) along with the non-negativity of entropy furnish the inequality conditions $0 \leq I(T; X) \leq H(T)$ and, similarly, $0 \leq I(T; X) \leq H(X)$. As we will see, the mutual information plays a central role in signal compression problems developed by information theorists.

The development of signal compression problems requires two components: (i) a measure that quantifies the degree of compression achieved by the reduced model, and (ii) a method to quantify the quality of the reduced model which captures how well the compressed representation represents the original model. These two terms, the model complexity and quality of compression, are in direct conflict in the sense that a higher-quality model comes at the expense of a lower degree of achieved compression. Forming high-quality abstractions is then a matter of identifying which information in a signal is relevant and should be preserved, and what is irrelevant and can be safely discarded.

To rigorously formalize these notions, we let the random variable X represent the original, uncompressed, signal and take the random variable T to represent the compressed representation of X . In addition, let $\Omega_X = \{x \in \mathbb{R} : X(\omega) = x, \omega \in \Omega\}$ and $\Omega_T = \{t \in \mathbb{R} : T(\omega) = t, \omega \in \Omega\}$ be the outcome spaces of X and T , respectively. Given the distribution $p(x)$ of the original signal, the framework of rate-distortion theory formalizes the optimal encoder selection problem as

$$\min_{p(t|x)} I(T; X), \quad (6)$$

subject to the constraint

$$\mathbb{E}(d(T, X)) \leq \tilde{D}, \quad (7)$$

where $d : \Omega_T \times \Omega_X \rightarrow [0, \infty)$ is a provided distortion function, which is assumed to take on small values when T is a good representation of X , and $\tilde{D} \geq 0$ is a given non-negative constant. The problem (6) subject to (7) is over all normalized distributions (encoders) $p(t|x)$ which map outcomes of X to outcomes of T , and where the expectation in the constraint relation (7) is with respect to the joint distribution $p(t, x) = p(t|x)p(x)$. The constrained optimization problem (6)-(7) has Lagrangian

$$\min_{p(t|x)} I(T; X) + \lambda \mathbb{E}(d(T, X)), \quad (8)$$

where $\lambda \geq 0$ is a Lagrange multiplier whose value depends on $\tilde{D}$. For a given value of $\lambda \geq 0$, the problem (8) can be solved analytically, resulting in the pair of formal, self-consistent equations given by

$$p(t|x) = \frac{p(t)}{\bar{Z}(x; \lambda)} \exp(-\lambda d(t, x)), \quad (9)$$

$$p(t) = \sum_x p(t|x)p(x), \quad (10)$$

where $\bar{Z}(x; \lambda) = \sum_t p(t) \exp(-\lambda d(t, x))$ [34, 6, 29]. The relations (9)-(10) are formal due to the dependence of $p(t)$ on the conditional distribution $p(t|x)$. Despite the fact that the relations do not yield an explicit solution, a numerical approach can be employed, known as the Blahut-Arimoto (BA) algorithm, to obtain a global solution to the rate-distortion problem [6].

In the rate-distortion framework, the degree of achieved compression is quantified by the mutual information $I(T; X)$ between the compressed representation and original signal. The use of $I(T; X)$ as a measure of model complexity is rooted in communication theory, where $I(T; X)$ is the rate, or rather, the average number of bits per message required to unambiguously specify an element in the codebook [34]. Consequently, lower rates imply that T is more compressed version of X . Moreover, observe that an encoder $p(t|x)$ resulting in full compression $I(T; X) = 0$ can always be obtained by removing all the details of a given signal X . An encoder $p(t|x)$ achieving full compression renders T and X statistically independent, and incurs a high distortion cost. Thus, by constraining the amount of allowable distortion, the framework must identify which aspects of the original signal are the most crucial to retain in order to maintain low distortion when designing the encoder $p(t|x)$.

To more clearly understand how the distortion function conveys which aspects of the signal are relevant, we inspect the formal solution in equation (9) for a given value of $\lambda \geq 0$ and $x \in \Omega_X$. Here, we see that those outcomes $t \in \Omega_T$ that result in lower distortion $d(t, x)$ are assigned greater probability mass as compared to those outcomes $t' \in \Omega_T$ that result in higher distortion $d(t, x) < d(t', x)$. Thus, the encoder $p(t|x)$ will assign greater probability mass to those outcomes $t \in \Omega_T$ which incur smaller values of $d(t, x)$ for each $x \in \Omega_X$ and $\lambda \geq 0$. It should be noted that the value of $\lambda \geq 0$ is dependent on $\bar{D}$ in the constraint (7) [6]. A notable drawback of the rate-distortion framework, however, lies in the need to specify the distortion function $d(t, x)$, which implicitly specifies which aspects of a signal are relevant, and should be preserved through the process of compression. Furthermore, the specification of a distortion function for a given task may be difficult and non-intuitive [34].

In contrast, the information-bottleneck method [34, 13] formulates an optimal encoder problem that does not require the specification of a distortion function. Instead, the IB problem aims to maximize the compression while explicitly constraining the amount of task-relevant information that is retained in the compressed representation. To formalize these notions, we take the random variable X to represent the original signal and denote its compressed representation by T . In addition, the IB problem introduces a third, relevant random variable, which we will denote by Y . The random variable Y is assumed to be correlated with X and contain the information that we wish to preserve when forming the compressed representation T . The IB method assumes that the joint distribution $p(x, y)$ is provided, and considers the problem

$$\min_{p(t|x)} I(T; X), \quad (11)$$

subject to the constraint

$$I(T; Y) \geq \hat{D}, \quad (12)$$

where optimization is over normalized distributions $p(t|x)$ and $\hat{D} \geq 0$ is a given constant. The joint distribution $p(t, x, y)$ factors according to $p(t, x, y) = p(t|x)p(x, y)$, which implies¹ $T \leftrightarrow X \leftrightarrow Y$ and, consequently, $I(T; Y) \leq I(X; Y)$, which follows from the data-processing inequality [6]. The IB problem was first introduced in [34], where the authors considered the problem

$$\min_{p(t|x)} I(T; X) - \beta I(T; Y), \quad (13)$$

where $\beta \geq 0$ is a scalar whose value depends on $\hat{D}$ [13]. For a given value of $\beta \geq 0$, the problem (13) can be solved analytically, resulting in the following formal relations that characterize the optimal solution

$$p(t|x) = \frac{p(t)}{Z(x; \beta)} \exp(-\beta D_{\text{KL}}(p(y|x), p(y|t))), \quad (14)$$

$$p(y|t) = \sum_x p(y|x)p(x|t), \quad (15)$$

$$p(t) = \sum_x p(t|x)p(x), \quad (16)$$

where $Z(x; \beta) = \sum_t p(t) \exp(-\beta D_{\text{KL}}(p(y|x), p(y|t)))$ [34, 29]. As in rate-distortion theory, the solution to the IB problem, characterized by relations (14)-(16), is formal as both the distributions $p(t)$ and $p(y|t)$ depend on the encoder $p(t|x)$. Moreover, comparing the relations (9) and (14), we see that the optimal solutions to the rate-distortion problem and IB method are very similar, and that the KL-divergence $D_{\text{KL}}(p(y|x), p(y|t))$ appears in place of a distortion function $d(t, x)$. Recall, however, that in contrast to rate-distortion theory, the IB method does not rely on the specification of a distortion function, instead requiring only the specification of the statistical distribution $p(x, y)$. It is therefore important to note that the KL-divergence $D_{\text{KL}}(p(y|x), p(y|t))$ in (14) *emerges* as a distortion-like function from the analytical derivation of the solution to (13), and is not specified or provided a priori [34, 29]. To conclude this section, we note that a solution to the IB problem (13) may be found as a function of β by employing a numerical approach that utilizes the relations (14)-(16) to construct an iterative method similar to the BA algorithm in rate-distortion theory, although convergence of the IB algorithm is guaranteed only to local solutions [34, 29].

¹ As T is a compressed representation of X , the IB problem searches for a mapping between outcomes of X and those of T . The resulting mapping (encoder) is not a function of the relevant variable Y , and so $p(t|x, y) = p(t|x)$ or $T \leftarrow X \leftarrow Y$. It can be shown that $T \leftarrow X \leftarrow Y$ implies $T \rightarrow X \rightarrow Y$ and thus $T \leftrightarrow X \leftrightarrow Y$.

3.2 Hierarchical Trees & Trees as Deterministic Encoders

Our goal in this chapter is to develop a framework for designing abstractions that endow autonomous systems the ability to trade environment complexity and model quality in a principled manner. To accomplish our goal, in the previous section we provided an overview of information-theoretic signal compression approaches that design abstract representations of general signals with fidelity guarantees. Importantly, the encoders that emerge as a result of the frameworks discussed in Section 3.1 do not impose any structural constraints on the resulting solution $p(t|x)$. However, as we will see, hierarchical multi-resolution tree abstractions can be realized as encoders $p(t|x)$ with very specific structure, and dealing with this additional structural constraint renders the resulting problem difficult to solve. In order to make these notions concrete, we introduce in this section the formalism of hierarchical tree structures and discuss their connection to encoder problems presented in the previous section.

Formally, a tree $\mathcal{T} = (\mathcal{N}(\mathcal{T}), \mathcal{E}(\mathcal{T}))$ is a connected acyclic graph defined by a set of nodes $\mathcal{N}(\mathcal{T})$ and a set of edges $\mathcal{E}(\mathcal{T})$, where the edges in $\mathcal{E}(\mathcal{T})$ describe the nodal interconnections [3]. Trees have been ubiquitously employed in the past in order to efficiently represent operating environments in robotics applications. A small sample of works that consider the use of multi-resolution trees in robotics include [17, 20, 12, 21, 18, 22]. In these works, environment abstractions in the form of multi-resolution hierarchical trees (specifically, quadrees and octrees) are employed in order to reduce the computational complexity of graph-search algorithms for autonomous planning, or to ease the on-board memory requirements to store environment representations. While tree structures are a general hierarchical data structure, we will focus our attention in this chapter on environment abstractions in the form of multi-resolution quadtree representations. It should be noted, however, that the methods discussed and presented in this chapter are applicable to general tree structures, beyond those of quadrees.

To formalize the connection between an environment and hierarchical tree structures, we will assume that there exists an integer $\ell > 0$ such that the world $\mathcal{W} \subset \mathbb{R}^2$ (generalizable to $\mathbb{R}^n$), is contained within a square (hypercube) of side length 2^ℓ . The environment $\mathcal{W}$ is assumed to be a two-dimensional grid-world where each element of $\mathcal{W}$ is a square (hypercube) of unit side length. We will take $\mathcal{T}^Q$ to denote the set of all valid, multi-resolution, quadtree representations of $\mathcal{W}$, and we let $\mathcal{T}_\mathcal{W} \in \mathcal{T}^Q$ be the finest resolution tree of the world $\mathcal{W}$. An illustrative example of $\mathcal{T}_\mathcal{W}$ for a 4×4 world $\mathcal{W}$ is shown in Fig. 2. Specific terminology is used to refer to some of the inter-node connections in $\mathcal{T}_\mathcal{W}$, which we describe in the following definition.

Definition 1 ([23]) Let $t \in \mathcal{N}(\mathcal{T}_\mathcal{W})$ be any node at depth $k \in \{0, \dots, \ell\}$. Then $t' \in \mathcal{N}(\mathcal{T}_\mathcal{W})$ is a *child* of t if the following hold:

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

Conversely, we say that t is the *parent* of t' if t' is a child of t . Furthermore, we let

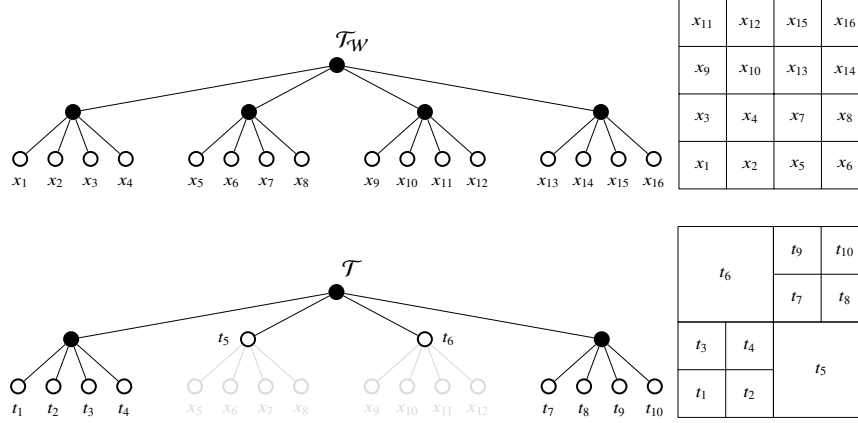


Fig. 2: Visual depiction of the tree $\mathcal{T}_W \in \mathcal{T}^Q$ along with a tree $\mathcal{T} \in \mathcal{T}^Q$ and corresponding multi-resolution (grid) representations of $\mathcal{W}$. Elements of the set $\mathcal{N}_{\text{int}}(\cdot)$ are shown in black, whereas elements of the set $\mathcal{N}_{\text{leaf}}(\cdot)$ are white and numbered. Notice that the tree $\mathcal{T}$ aggregates nodes x_5, x_6, x_7 , and x_8 to t_5 and the nodes x_5, x_6, x_7 , and x_8 to t_6 . The nodes x_1, x_2, x_3 , and x_4 as well as x_{13}, x_{14}, x_{15} , and x_{16} are not aggregated in this example. Aggregated nodes of $\mathcal{T}_W$ are shown in gray.

$$\mathcal{N}_k(\mathcal{T}) = \{t \in \mathcal{N}(\mathcal{T}) : t \text{ is at depth } k \text{ in } \mathcal{T}_W\},$$

to be the set of all nodes of the tree $\mathcal{T} \in \mathcal{T}^Q$ at depth k . $\triangle$

Accordingly, for any $t \in \mathcal{N}(\mathcal{T}_W)$, we let $C(t)$ denote the set of children of the node t . Furthermore, given any tree $\mathcal{T} \in \mathcal{T}^Q$, we say that the node t is a *leaf node* of $\mathcal{T}$ if $C(t) \cap \mathcal{N}(\mathcal{T}) = \emptyset$ and denote the set of all leaf nodes for any $\mathcal{T} \in \mathcal{T}^Q$ by $\mathcal{N}_{\text{leaf}}(\mathcal{T}) = \{t \in \mathcal{N}(\mathcal{T}_W) : C(t) \cap \mathcal{N}(\mathcal{T}) = \emptyset\}$.² The nodes in the set $\mathcal{N}_{\text{int}}(\mathcal{T}) = \mathcal{N}(\mathcal{T}) \setminus \mathcal{N}_{\text{leaf}}(\mathcal{T})$ are called *interior nodes*. An example showing the relation between the sets $\mathcal{N}_{\text{leaf}}(\mathcal{T})$ and $\mathcal{N}_{\text{int}}(\mathcal{T})$ is presented in Fig. 2.

The connection between hierarchical data structures, such as multi-resolution quadrees, and signal encoders $p(t|x)$ described in Section 3.1 was recently discussed in [23]. As part of their study, the authors of [23] discuss how any tree $\mathcal{T}_q \in \mathcal{T}^Q$ can be viewed as a deterministic encoder³ $p_q(t|x)$ that maps leaf nodes $x \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_W)$ to leaf nodes $t \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_q)$ of the tree $\mathcal{T}_q$. Specifically, given any tree $\mathcal{T}_q \in \mathcal{T}^Q$ a corresponding deterministic encoder $p_q(t|x)$ can be constructed such that $p_q(t|x) = 1$ if and only if $x \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_W)$ is aggregated to the leaf node $t \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_q)$, and $p(t|x) = 0$ otherwise. Consequently, by changing the tree $\mathcal{T}_q$ we change the corresponding encoder $p_q(t|x)$, which, in turn, changes the multi-resolution environment representation of $\mathcal{W}$ defined by the leaves of $\mathcal{T}_q$, as shown in

² Note that the set $\mathcal{N}(\mathcal{T}_W)$ contains all the available nodes.

³ A deterministic encoder is an encoder $p(t|x)$ such that $p(t|x) \in \{0, 1\}$ for all t and x . Deterministic encoders are also called hard encoders.

Fig. 2. By facilitating a connection between signal encoders and hierarchical trees allows us to utilize the compression frameworks discussed in Section 3.1 in order to develop problem formulations for the emergence of task-driven, multi-resolution, environment abstractions that can be tailored to agent resource constraints. We now turn to describe how such a problem formulation can be developed, and the technical considerations that one must observe when searching for a solution to the resulting problem.

4 Information-Theoretic Multi-Resolution Tree Abstractions

From the discussion in the previous section, it follows that if we are provided a grid-world $\mathcal{W}$, then the finest resolution tree corresponding to $\mathcal{W}$ is given by the tree $\mathcal{T}_{\mathcal{W}}$ whose leafs $x \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}})$ are the original grid-cells of the environment. Our objective in this section is to describe how one can utilize the information-theoretic ideas from Section 3.1 in order to formulate a problem whose solution is a multi-resolution, hierarchical tree representation of the world $\mathcal{W}$.

To this end, we assume that a grid-world $\mathcal{W}$ is provided along with a joint distribution $p(x, y)$. The joint distribution $p(x, y)$ captures the correlation between the original grid cells X and the relevant random variable Y . For example, if the world $\mathcal{W}$ is an occupancy grid and the objective is to form abstractions of $\mathcal{W}$ that are maximally predictive regarding the occupancy values, then $p(x, y)$ may be specified as $p(x, y) = p(y|x)p(x)$, where the occupancy information is described by the conditional distribution $p(y|x)$, and $p(x)$ is a distribution over world cells (e.g., distribution over possible locations of an autonomous agent). Since each tree $\mathcal{T}_q \in \mathcal{T}^Q$ can be represented by a *deterministic* encoder $p_q(t|x)$, it follows that by changing the tree, we alter the joint distribution $p_q(t, x, y)$ according to $p_q(t, x, y) = p_q(t|x)p(x, y)$. Consequently, a random variable $T_q : \Omega \rightarrow \mathbb{R}$ can be constructed for each tree $\mathcal{T}_q \in \mathcal{T}^Q$, whose outcomes are the leafs of $\mathcal{T}_q$ and has distribution $p_q(t) = \sum_x p_q(t|x)p(x)$. The random variable T_q for each $\mathcal{T}_q \in \mathcal{T}^Q$ can be viewed as a compressed representation of the random variable X created by aggregating finest-resolution grid-cells of the world $\mathcal{W}$. Thus, for each tree $\mathcal{T}_q \in \mathcal{T}^Q$ we define the mappings $I_Y : \mathcal{T}^Q \rightarrow [0, \infty)$ and $I_X : \mathcal{T}^Q \rightarrow [0, \infty)$ according to the rules

$$I_Y(\mathcal{T}_q) = I(T_q; Y) = \sum_{t,y} p_q(t, y) \log \frac{p_q(t, y)}{p_q(t)p(y)}, \quad (17)$$

and

$$I_X(\mathcal{T}_q) = I(T_q; X) = \sum_{t,x} p_q(t, x) \log \frac{p_q(t, x)}{p_q(t)p(x)}, \quad (18)$$

respectively. We note that in equations (17) and (18) the distributions $p(x)$ and $p(y)$ are not a function of the tree $\mathcal{T}_q \in \mathcal{T}^Q$, as they can be obtained by marginalizing the input distribution $p(x, y)$. By changing the tree $\mathcal{T}_q \in \mathcal{T}^Q$, we change the en-

coder $p_q(t|x)$, thereby changing the amount of relevant information captured by the compressed representation T_q as well as the multi-resolution depiction of $\mathcal{W}$.

Given the previous connection between encoders and trees, we observe that formalizing the problem of obtaining multi-resolution tree abstractions of $\mathcal{W}$ that are maximally informative regarding the task-relevant variable Y , can be done by employing an information-theoretic framework for signal compression with additional constraints on the structure of the encoder $p(t|x)$. The additional constraints placed on the encoder structure are necessary to ensure that the resulting solution to the compression problem corresponds to some tree $\mathcal{T} \in \mathcal{T}^Q$. The problem we consider in the remainder of this chapter is now formally stated as follows.

Problem 1 *Given a joint distribution $p(x, y)$, a scalar $\hat{D} \geq 0$, and the world $\mathcal{W}$, we consider the problem*

$$\min_{\mathcal{T}_q \in \mathcal{T}^Q} I_X(\mathcal{T}_q), \quad (19)$$

subject to the constraint

$$I_Y(\mathcal{T}_q) \geq \hat{D}. \quad (20)$$

Problem 1 is equivalent to the problem (11) subject to (12) with the added constraint $\mathcal{T}_q \in \mathcal{T}^Q$ that the encoder corresponds to a valid tree representation of $\mathcal{W}$. Imposing the constraint $\mathcal{T}_q \in \mathcal{T}^Q$ introduces a host of additional challenges, as existing methods for obtaining a solution to the problem (11)-(12) do not consider structural constraints on the encoder $p(t|x)$. In fact, the solution to the IB problem is, in general, a stochastic mapping $p(t|x)$ [34]. Other approaches, such as the agglomerative information bottleneck (AIB) [30, 29], design deterministic encoders by utilizing the IB principle and iteratively merging elements of the compressed representation T until only the trivial abstraction, where T only has a single outcome, remains. Although the end result of AIB is a family of deterministic encoders, the method does not consider the constraints requiring the encoder $p(t|x)$ to be a member of some space, for example correspond to some $\mathcal{T} \in \mathcal{T}^Q$. Our problem is fundamentally different: we employ the IB principle to obtain a tree $\mathcal{T}_q \in \mathcal{T}^Q$, which is a solution to (19)-(20) as a function of $\hat{D} \geq 0$, by enforcing the constraint $\mathcal{T}_q \in \mathcal{T}^Q$. Notice that the Problem 1, which aims to maximize compression subject to a lower-bound on the retained relevant information, is equivalent to the following problem [13].

Problem 2 *Given the joint distribution $p(x, y)$, a scalar $D \geq 0$, and the world $\mathcal{W}$, we consider the problem*

$$\max_{\mathcal{T}_q \in \mathcal{T}^Q} I_Y(\mathcal{T}_q), \quad (21)$$

subject to the constraint

$$I_X(\mathcal{T}_q) \leq D. \quad (22)$$

As opposed to maximizing compression subject to a lower bound on the retained relevant information, Problem 2 aims to maximize the relevant information subject to a bound on the achieved compression. Specifically, the result of solving problem (19) subject to (20) as a function of $\hat{D} \geq 0$ is a multi-resolution tree $\mathcal{T}_q \in \mathcal{T}^Q$ that

maximizes compression of $\mathcal{W}$ while ensuring the resulting representation contains at least $\hat{D}$ units of relevant information. Alternatively, Problem 2 allows us to consider the emergence of tree abstractions of $\mathcal{W}$ that are maximally informative regarding the relevant variable Y while constraining the allowable compression to at least $D \geq 0$ units. As the value of $D \geq 0$ in (22) increases, one obtains abstractions that are of higher leaf node cardinality, provided that they relay more task-relevant information. The two formulations are equivalent, and provide different interpretations of the same information-theoretic compression principle [13].

The most closely related work to our formulation in Problems 1 and 2 is the previous work by the authors of [23, 25]. In [23] the authors considered the problem

$$\max_{\mathcal{T}_q \in \mathcal{T}^Q} I_Y(\mathcal{T}_q) - \frac{1}{\beta} I_X(\mathcal{T}_q), \quad (23)$$

as a function of $\beta > 0$, assuming $\mathcal{W}$ and $p(x, y)$ are provided. The optimization problem (23) is equivalent to the problem (13) restricted to the space $\mathcal{T}_q \in \mathcal{T}^Q$, as can be seen by multiplying (13) by the constant $-1/\beta$. The authors of [23] develop an algorithm to solve (23) as a function of $\beta > 0$, termed *Q-tree search*, which utilizes a function similar to Q-functions employed in reinforcement learning, in order to decide which nodes in the tree to prune [2, 32]. However, a notable drawback to the Q-tree search algorithm is its dependence on $\beta > 0$, which is, in turn, a function of the distribution $p(x, y)$. Consequently, selecting a value of $\beta > 0$ for a given level of desired retained relevant information $I_Y(\mathcal{T})$ is non-trivial and is further exacerbated when the environment distribution $p(x, y)$ is time-varying, as is the case in many autonomous system applications. In contrast, our formulations in Problems 1 and 2 are independent of β , instead allowing for the specification of the constraints directly in terms of constants ($\hat{D}$ and D , respectively) that do not depend on $p(x, y)$. Moreover, we argue that agent resource constraints, such as on-board memory or capacity of communication channels available to the agent, are more naturally specified in terms of constraints of the form of (22) that have an information-theoretic interpretation relating to channel capacity and data-rate, as compared to specifying the same resource constraints in terms of a scalar $\beta > 0$.

While our formulation is a well-posed optimization problem, it is not obvious how to obtain the solution. A brute-force solution approach can be employed that exhaustively generates and enumerates all feasible trees in the set $\mathcal{T}^Q$. However, exhaustive search methods do not scale well to large environments, as the set $\mathcal{T}^Q$ quickly grows so as to render the computational cost of exhaustively generating each of the trees in $\mathcal{T}^Q$ prohibitively expensive. Instead, alternative solution approaches must be developed that exploit the structural properties of the problem. In the next section, we show how Problem 1 (equivalently, Problem 2) can be posed as an integer linear program.

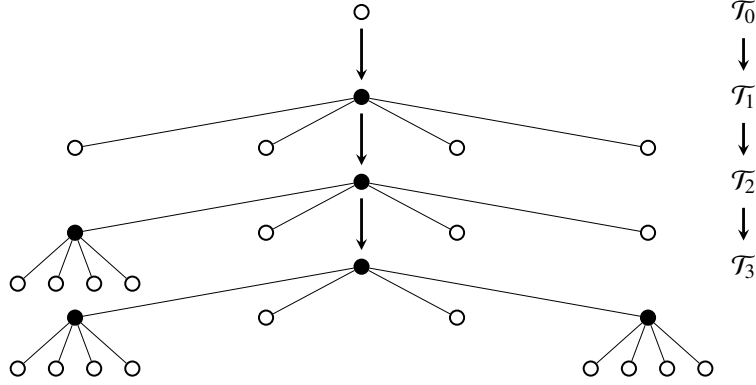


Fig. 3: Sequence of trees $\{\mathcal{T}_i\}_{i=0}^3 \subseteq \mathcal{T}^Q$ leading from $\mathcal{T}_0 = \mathcal{R}_W$, the root of $\mathcal{T}_W$, to the tree $\mathcal{T}_3$. Note that $\mathcal{N}(\mathcal{T}_{i+1}) \setminus \mathcal{N}(\mathcal{T}_i) = C(t)$ for some $t \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_i)$ holds for all $i \in \{0, 1, 2\}$. For each tree $\mathcal{T}_i$, $i \in \{0, 1, 2, 3\}$, the set of interior nodes, $\mathcal{N}_{\text{int}}(\mathcal{T}_i)$, is colored black whereas nodes in $\mathcal{N}_{\text{leaf}}(\mathcal{T}_i)$ are white. For the tree $\mathcal{R}_W$, the set of interior nodes is empty.

5 Information-Theoretic Abstractions via Linear Programming

In order to formulate a precise mathematical optimization problem, we look to exploit the structural properties of Problem 1. To this end, our overall approach is to write the mutual information terms (19) and (20) in terms of local changes in the tree, and show how the information content in any tree $\mathcal{T}_q \in \mathcal{T}^Q$ can be expressed as the sum of all local changes required to obtain $\mathcal{T}_q$ when starting from the root node of $\mathcal{W}$.

To begin, we write the information terms (19) and (20) of any tree $\mathcal{T}_q \in \mathcal{T}^Q$ as

$$I_X(\mathcal{T}_q) = I_X(\mathcal{T}_0) + \sum_{i=0}^{q-1} [I_X(\mathcal{T}_{i+1}) - I_X(\mathcal{T}_i)], \quad (24)$$

and

$$I_Y(\mathcal{T}_q) = I_Y(\mathcal{T}_0) + \sum_{i=0}^{q-1} [I_Y(\mathcal{T}_{i+1}) - I_Y(\mathcal{T}_i)], \quad (25)$$

respectively. Relations (24) and (25) allow us to write the information content of the tree $\mathcal{T}_q \in \mathcal{T}^Q$ in terms of a sequence $\{\mathcal{T}_0, \dots, \mathcal{T}_{q-1}, \mathcal{T}_q\}$, where $\mathcal{T}_i \in \mathcal{T}^Q$ for all $i \in \{0, \dots, q\}$. The validity of equations (24) and (25) does not depend on the specific sequence $\{\mathcal{T}_0, \dots, \mathcal{T}_{q-1}, \mathcal{T}_q\}$, instead allowing us to express the value of $I_X(\mathcal{T}_q)$ and $I_Y(\mathcal{T}_q)$ as a function of other trees $\mathcal{T}_0, \dots, \mathcal{T}_{q-1}$, irrespective of how the intermediate trees are selected. A key observation is that by selecting the intermediate trees $\mathcal{T}_0, \dots, \mathcal{T}_{q-1}$ in a specific way, we arrive at tractable expressions for $I_X(\mathcal{T}_q)$ and $I_Y(\mathcal{T}_q)$, as follows.

To this end, we notice that any tree $\mathcal{T}_q \in \mathcal{T}^Q$ can be expressed as a sequence of trees $\{\mathcal{T}_0, \dots, \mathcal{T}_{q-1}\}$ by starting at the root tree⁴ $\mathcal{T}_0 = \mathcal{R}_W$ and expanding a leaf node of each $\mathcal{T}_i, i \in \{0, \dots, q-1\}$, until we reach $\mathcal{T}_q$. An illustrative example is shown in Fig. 3. By considering sequences of this form, each tree $\mathcal{T}_i$ in the sequence $\{\mathcal{T}_i\}_{i=0}^{q-1}$ differs from the tree $\mathcal{T}_{i+1}$ by only a single leaf node expansion. In other words, for all $i \in \{0, \dots, q-1\}$ it holds that $\mathcal{N}(\mathcal{T}_{i+1}) \setminus \mathcal{N}(\mathcal{T}_i) = C(t)$ for some $t \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_i)$. We will show that, in this case, the difference in information $I_X(\mathcal{T}_{i+1}) - I_X(\mathcal{T}_i)$ and $I_Y(\mathcal{T}_{i+1}) - I_Y(\mathcal{T}_i)$ in (24) and (25) have a special form.

To elucidate this special form, let us define, for any two trees $\mathcal{T}_{i+1}, \mathcal{T}_i \in \mathcal{T}^Q$ such that $\mathcal{N}(\mathcal{T}_{i+1}) \setminus \mathcal{N}(\mathcal{T}_i) = C(t)$ for some $t \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_i)$, the function ΔI_X

$$\Delta I_X(\mathcal{T}_{i+1}, \mathcal{T}_i) = I_X(\mathcal{T}_{i+1}) - I_X(\mathcal{T}_i). \quad (26)$$

From the definition of the function I_X in (18), as well as the properties of mutual information (5), we see that (26) can be written as

$$\Delta I_X(\mathcal{T}_{i+1}, \mathcal{T}_i) = H(T_{i+1}) - H(T_i) - H(T_{i+1}|X) + H(T_i|X). \quad (27)$$

Since each tree $\mathcal{T}_q \in \mathcal{T}^Q$ defines a deterministic encoder $p_q(t|x)$ (see Section 3.2), it follows that there is no uncertainty in the random variable T_q when X is known (i.e., X specifies the outcome of T_q with certainty), leading to $H(T_q|X) = 0$ for all $\mathcal{T}_q \in \mathcal{T}^Q$. As a result, equation (27) becomes

$$\Delta I_X(\mathcal{T}_{i+1}, \mathcal{T}_i) = H(T_{i+1}) - H(T_i). \quad (28)$$

Strictly speaking, (28) is valid for any deterministic encoding of X , as we have not yet utilized the fact that $\mathcal{N}(\mathcal{T}_{i+1}) \setminus \mathcal{N}(\mathcal{T}_i) = C(t)$ for some $t \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_i)$ to arrive at this relation. Now let $C(t) = \{t'_1, \dots, t'_4\}$ be the set of children of the node $t \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_i)$, and utilize the definition of entropy (1) to write

$$\Delta I_X(\mathcal{T}_{i+1}, \mathcal{T}_i) = H(T_{i+1}) - H(T_i) = - \sum_{i=1}^4 p(t'_i) \log p(t'_i) + p(t) \log p(t), \quad (29)$$

which follows since the only difference between the trees $\mathcal{T}_i$ and $\mathcal{T}_{i+1}$ is the aggregation of the nodes in $C(t) \subseteq \mathcal{N}_{\text{leaf}}(\mathcal{T}_{i+1})$ to their parent $t \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_i)$, where all other nodes remain unchanged. Note that

$$p(t) = \sum_{i=1}^4 p(t'_i), \quad (30)$$

which follows immediately from the properties of the tree structure and of deterministic encoders. Using (30), relation (29) is equivalently written as

⁴ The root tree is the tree $\mathcal{R}_W \in \mathcal{T}^Q$ for which $\mathcal{N}_{\text{int}}(\mathcal{R}_W) = \emptyset$. That is, $\mathcal{R}_W$ consists of only a single leaf node.

$$\Delta I_X(\mathcal{T}_{i+1}, \mathcal{T}_i) = p(t) \left[- \sum_{i=1}^4 \frac{p(t'_i)}{p(t)} \log \frac{p(t'_i)}{p(t)} \right] = p(t)H(\Pi), \quad (31)$$

where $\Pi \in \mathbb{R}^4$ is the distribution defined by

$$\Pi = \left[\frac{p(t'_1)}{p(t)}, \dots, \frac{p(t'_4)}{p(t)} \right]. \quad (32)$$

Equation (31) implies that when the trees $\mathcal{T}_{i+1}$ and $\mathcal{T}_i$ differ by only a single leaf node expansion, the change in information $\Delta I_X(\mathcal{T}_{i+1}, \mathcal{T}_i)$ is only a function of the node $t \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_i)$ expanded to create $\mathcal{T}_{i+1}$, and it does not depend on the other nodes in the tree $\mathcal{T}_i$. As a result, while we write $\Delta I_X(\mathcal{T}_{i+1}, \mathcal{T}_i)$ as a function of trees, it can be equivalently written as a function of nodes $\Delta \hat{I}_X : \mathcal{N}_{\text{int}}(\mathcal{T}_{\mathcal{W}}) \rightarrow [0, \infty)$ defined by⁵

$$\Delta \hat{I}_X(t) = p(t)H(\Pi). \quad (33)$$

Relation (33) quantifies the difference $\Delta I_X(\mathcal{T}_{i+1}, \mathcal{T}_i)$ between *any* two trees $\mathcal{T}_{i+1}, \mathcal{T}_i \in \mathcal{T}^Q$ that differ only by the expansion of $t \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_i) \subseteq \mathcal{N}_{\text{int}}(\mathcal{T}_{\mathcal{W}})$.

Having obtained a simplified expression for $\Delta I_X(\mathcal{T}_{i+1}, \mathcal{T}_i)$ in (33), we now turn our attention to $\Delta I_Y(\mathcal{T}_{i+1}, \mathcal{T}_i)$. In a similar spirit, we define the function ΔI_Y between trees $\mathcal{T}_{i+1}$ and $\mathcal{T}_i$ such that $\mathcal{N}(\mathcal{T}_{i+1}) \setminus \mathcal{N}(\mathcal{T}_i) = C(t)$ for some $t \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_i)$, according to the rule

$$\Delta I_Y(\mathcal{T}_{i+1}, \mathcal{T}_i) = I_Y(\mathcal{T}_{i+1}) - I_Y(\mathcal{T}_i). \quad (34)$$

Employing the definition of mutual information (4) and observing that the trees $\mathcal{T}_{i+1}$ and $\mathcal{T}_i$ are such that $\mathcal{N}(\mathcal{T}_{i+1}) \setminus \mathcal{N}(\mathcal{T}_i) = C(t)$ for some $t \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_i)$, one obtains

$$\Delta I_Y(\mathcal{T}_{i+1}, \mathcal{T}_i) = p(t) \left[\sum_{i=1}^4 \frac{p(t'_i)}{p(t)} \sum_y p(y|t'_i) \log p(y|t'_i) - \sum_y p(y|t) \log p(y|t) \right]. \quad (35)$$

From the tree structure, one can show that for any $t \in \mathcal{N}_{\text{int}}(\mathcal{T}_{\mathcal{W}})$

$$p(y|t) = \sum_{i=1}^4 [\Pi]_i p(y|t'_i), \quad (36)$$

where Π is defined as in (32). As a result of (35) and (36), it follows that (34) can be written as

$$\Delta I_Y(\mathcal{T}_{i+1}, \mathcal{T}_i) = p(t) \sum_{i=1}^4 [\Pi]_i \text{D}_{\text{KL}}(p(y|t'_i), p(y|t)). \quad (37)$$

Relation (37) is equivalent to

$$\Delta I_Y(\mathcal{T}_{i+1}, \mathcal{T}_i) = p(t) \text{JS}_{\Pi}(p(y|t'_1), \dots, p(y|t'_4)), \quad (38)$$

⁵ To evaluate the function $\Delta \hat{I}_X(t)$, we require the node t to have children. The collection of all nodes that have children are those in the set $\mathcal{N}_{\text{int}}(\mathcal{T}_{\mathcal{W}})$.

where $\text{JS}_\Pi(p(y|t'_1), \dots, p(y|t'_4))$ is the *Jensen-Shannon (JS) divergence* among the distributions $p(y|t'_1), \dots, p(y|t'_4)$ with weights Π [26].

We note that equation (37) is only a function of the nodes that differ between the two trees $\mathcal{T}_{i+1}$ and $\mathcal{T}_i$, as was the case for ΔI_X . Thus, although $\Delta I_Y(\mathcal{T}_{i+1}, \mathcal{T}_i)$ is written as a function of trees, it can be expressed as a function of nodes according to the mapping $\Delta \hat{I}_Y : \mathcal{N}_{\text{int}}(\mathcal{T}_W) \rightarrow [0, \infty)$ defined by

$$\Delta \hat{I}_Y(t) = p(t) \text{JS}_\Pi(p(y|t'_1), \dots, p(y|t'_4)), \quad (39)$$

where Π is defined in (32) and $C(t) = \{t'_1, \dots, t'_4\}$. Consequently, $\Delta \hat{I}_Y(t)$ captures the incremental change in relevant information between *any* two trees $\mathcal{T}_{i+1}, \mathcal{T}_i \in \mathcal{T}^Q$ that are such that $C(t) = \mathcal{N}(\mathcal{T}_{i+1}) \setminus \mathcal{N}(\mathcal{T}_i)$ for $t \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_i)$.

Returning to relations (24) and (25), we notice that (31) and (38) furnish tractable expressions for the terms $I_X(\mathcal{T}_{i+1}) - I_X(\mathcal{T}_i)$ and $I_Y(\mathcal{T}_{i+1}) - I_Y(\mathcal{T}_i)$, respectively, when given a sequence of trees $\{\mathcal{T}_i\}_{i=1}^q$ satisfying $\mathcal{N}(\mathcal{T}_{i+1}) \setminus \mathcal{N}(\mathcal{T}_i) = C(t)$ for some $t \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_i)$ and all $i \in \{0, \dots, q-1\}$. Furthermore, from (5) and the non-negativity of entropy, we have $0 \leq I_X(\mathcal{T}_0) \leq H(\mathcal{T}_0)$ and $0 \leq I_Y(\mathcal{T}_0) \leq H(\mathcal{T}_0)$. As a result, selecting the tree $\mathcal{T}_0$ to be the root tree $\mathcal{R}_W$ results in $I_Y(\mathcal{T}_0) = I_X(\mathcal{T}_0) = 0$, as, in this case, the random variable T_0 has only a single outcome (i.e., T_0 is deterministic), and thus $H(\mathcal{T}_0) = 0$. Lastly, we observe that for any tree $\mathcal{T}_q \in \mathcal{T}^Q$, all sequences $\{\mathcal{T}_0, \dots, \mathcal{T}_q\}$ leading from $\mathcal{T}_0 = \mathcal{R}_W$ to $\mathcal{T}_q$ and satisfying $\mathcal{N}(\mathcal{T}_{i+1}) \setminus \mathcal{N}(\mathcal{T}_i) = C(t)$ for some $t \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_i)$ and all $i \in \{0, \dots, q-1\}$, must expand all the interior nodes of $\mathcal{T}_q$. Accordingly, equations (24) and (25) can be written, for any $\mathcal{T}_q \in \mathcal{T}^Q$, as

$$I_X(\mathcal{T}_q) = \sum_{t \in \mathcal{N}_{\text{int}}(\mathcal{T}_q)} \Delta \hat{I}_X(t), \quad (40)$$

and

$$I_Y(\mathcal{T}_q) = \sum_{t \in \mathcal{N}_{\text{int}}(\mathcal{T}_q)} \Delta \hat{I}_Y(t), \quad (41)$$

respectively. It should also be noted that relations (40) and (41) are valid for all $\mathcal{T}_q \in \mathcal{T}^Q$, since every tree in the space $\mathcal{T}^Q$ can be obtained by starting at the root tree $\mathcal{R}_W \in \mathcal{T}^Q$ and performing a sequence of expansions according to the process described above.

At this point, we mention two important observations. First, notice that the quantities $I_X(\mathcal{T}_q)$ and $I_Y(\mathcal{T}_q)$ are both fully determined once the interior node set $\mathcal{N}_{\text{int}}(\mathcal{T}_q)$ is provided. Secondly, the interior node set also fully characterizes the tree $\mathcal{T}_q$. In other words, given the set $\mathcal{N}_{\text{int}}(\mathcal{T}_q)$, we are able to uniquely determine which tree $\mathcal{T}_q \in \mathcal{T}^Q$ corresponds to this interior node set. As a result, the set $\mathcal{N}_{\text{int}}(\mathcal{T}_q)$ determines the multi-resolution structure of $\mathcal{W}$.

5.1 Integer Linear Program Formulation

Our goal is to exploit the previous observations and utilize the expressions (40) and (41) to pose Problem 1 as a search for interior nodes.

To this end, we introduce the vector $\mathbf{z} \in \mathbb{R}^{|\mathcal{N}_{\text{int}}(\mathcal{T}_W)|}$ whose entries $[\mathbf{z}]_t \in \{0, 1\}$ indicate whether or not the node $t \in \mathcal{N}_{\text{int}}(\mathcal{T}_W)$ is part of the interior node set $\mathcal{N}_{\text{int}}(\mathcal{T}_q)$ of the tree $\mathcal{T}_q$. The entries of $\mathbf{z}$ are such that, for any $t \in \mathcal{N}_{\text{int}}(\mathcal{T}_W)$, if $[\mathbf{z}]_t = 1$ then $t \in \mathcal{N}_{\text{int}}(\mathcal{T}_q)$, whereas when $[\mathbf{z}]_t = 0$ implies $t \notin \mathcal{N}_{\text{int}}(\mathcal{T}_q)$. We then define the vectors $\Delta_X \in \mathbb{R}^{|\mathcal{N}_{\text{int}}(\mathcal{T}_W)|}$ and $\Delta_Y \in \mathbb{R}^{|\mathcal{N}_{\text{int}}(\mathcal{T}_W)|}$ such that, for all $t \in \mathcal{N}_{\text{int}}(\mathcal{T}_W)$, $[\Delta_X]_t = \Delta \hat{I}_X(t)$ and $[\Delta_Y]_t = \Delta \hat{I}_Y(t)$. The relations (40) and (41) can then be written in terms of these vectors as $I_X(\mathcal{T}_q) = \mathbf{z}^\top \Delta_X$ and $I_Y(\mathcal{T}_q) = \mathbf{z}^\top \Delta_Y$, respectively. It then follows that Problem 1 can be expressed as the integer linear program (ILP)

$$\min_{\mathbf{z}} \mathbf{z}^\top \Delta_X, \quad (42)$$

subject to

$$\mathbf{z}^\top \Delta_Y \geq \hat{D}, \quad (43)$$

$$[\mathbf{z}]_{t'} - [\mathbf{z}]_t \leq 0, \quad t \in \mathcal{B}, \quad t' \in C(t), \quad (44)$$

$$[\mathbf{z}]_t \in \{0, 1\}, \quad t \in \mathcal{N}_{\text{int}}(\mathcal{T}_W), \quad (45)$$

where $\hat{D} \geq 0$ and $\mathcal{B} = \{t \in \mathcal{N}_{\text{int}}(\mathcal{T}_W) : C(t) \cap \mathcal{N}_{\text{int}}(\mathcal{T}_W) \neq \emptyset\}$. The constraint (44) is required as not all vectors $\mathbf{z}$ such that $[\mathbf{z}]_t \in \{0, 1\}$ for $t \in \mathcal{N}_{\text{int}}(\mathcal{T}_W)$ correspond to a valid tree representation in the space $\mathcal{T}^Q$. Thus, relation (44) ensures that the resulting vector $\mathbf{z}$ indeed corresponds to a feasible tree in the space $\mathcal{T}^Q$, and does so by enforcing that the children $t' \in C(t)$ of some $t \in \mathcal{N}_{\text{int}}(\mathcal{T}_W)$ cannot be expanded unless their parent t has been expanded as well (i.e., if $[\mathbf{z}]_t = 0$ then $[\mathbf{z}]_{t'} = 0$). We need only to enforce the constraint (44) for those nodes $t \in \mathcal{N}_{\text{int}}(\mathcal{T}_W)$ that have children that are expandable, which are those in the set $\mathcal{B}$.

Similarly, it follows that Problem 2 can be formulated as the following ILP

$$\max_{\mathbf{z}} \mathbf{z}^\top \Delta_Y, \quad (46)$$

subject to the constraints

$$\mathbf{z}^\top \Delta_X \leq D, \quad (47)$$

$$[\mathbf{z}]_{t'} - [\mathbf{z}]_t \leq 0, \quad t \in \mathcal{B}, \quad t' \in C(t), \quad (48)$$

$$[\mathbf{z}]_t \in \{0, 1\}, \quad t \in \mathcal{N}_{\text{int}}(\mathcal{T}_W), \quad (49)$$

where $D \geq 0$ is a given constant.

The two ILPs offer different perspectives of the same information-theoretic principle [34, 13]. In the context of resource constrained agents, the formulation (46)-(49) can be viewed as a method for designing multi-resolution tree abstractions for agents that have memory or communication constraints, as the constraint (47) can be inter-

puted from a data-rate constraint, as discussed in Section 3.1. In these scenarios, the value of D is chosen so as to reflect the capacity of available communication channels or the amount of available memory resources. In contrast, the formulation (42)-(45) is for agents who wish to compress the environment representation as much as possible while ensuring at least $\hat{D} \geq 0$ units of relevant information are retained in the resulting abstraction.

Our approach allows for a rich set of task-driven, multi-resolution, tree abstractions to be obtained for a given operating environment. There are, however, many technical considerations pertaining to quantifying the quality of the solutions obtained by solving these programs. Furthermore, it is not clear how one should compare trees obtained by employing the approach developed in this chapter to existing methods for tree abstraction. In the next section, we discuss these questions and also introduce an alternative way of obtaining multi-resolution tree abstractions that does not require solving an integer programming problem.

6 The Information Plane, Pareto Efficiency and LP Relaxation

This section discusses a number of technical details regarding the solutions to the integer programs formulated in the previous section. In addition, we also briefly discuss an alternate solution approach to the ILPs (42)-(45) and (46)-(49) that leverages ideas from convex optimization. We first consider our formulation in the context of multi-objective optimization theory, investigating whether or not the multi-resolution tree abstractions we obtain by solving the ILPs (42)-(45) and (46)-(49) are the most informative trees possible for a given level of compression. To this end, we provide a detailed discussion regarding Pareto efficiency of solutions and introduce a numerical method that can be employed in order to trace the set of Pareto optimal values. A central component of our discussion of Pareto optimality is the so-called *information plane*, which graphically shows the compression vs. relevant information trade-offs of the solutions to (42)-(45). We then close this section by presenting a relaxation approach towards solving the integer programs, which is obtained by removing the integrality constraints (45) and (49), respectively. The resulting optimization problem is then a linear program. As the solution to the linear program may not be integer, we provide an approach to obtain an integer vector from the linear program solution that is guaranteed to result in a valid multi-resolution tree representation of $\mathcal{W}$.

6.1 Pareto Efficiency of Solutions

When discussing the quality of solutions obtained by the IB method, it is common to plot solutions in the so-called information plane. Specifically, the term *information plane* refers to the graphical display of compression-relevance data on a plot with abscissa $I(T; X)$ and ordinate $I(T; Y)$ [29, 34]. An example is shown in Fig. 4.

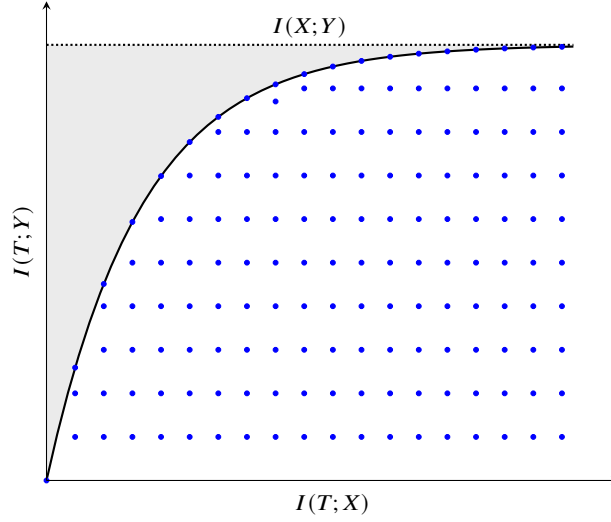


Fig. 4: Example information plane with the discrete feasible set of trees (blue points). The information plane places relevant information on the ordinate axis and compression on the abscissa (compression decreasing from left to right). The curve traces the trade-off between maximal retention of relevant information vs. compression. In the standard IB problem (13), this curve is characterized by a concave function, but is a discrete set of points in the setting of hierarchical tree abstractions of $\mathcal{W}$. The grey shaded region above the curve is not achievable by any tree. The dashed horizontal line is the value of $I(X; Y)$, which is determined by the input distribution $p(x, y)$. The solution approaches this value of $I(X; Y)$ as either $\beta \rightarrow \infty$ in (23) or $\hat{D} \rightarrow I(X; Y)$ in (43).

Before we formalize our problem in this section, we discuss a number of important characteristics of the information shown in Fig. 4. Firstly, it should be noted that the information plane is two dimensional regardless of the encoder structure. In other words, the two scalars that quantify the attractiveness of a valid encoder are: (i) its degree of achieved compression $I(T; X)$, and (ii) the amount of relevant information it retains $I(T; Y)$. Secondly, it is important to note that our problem is discrete, as the space of feasible multi-resolution tree representations of $\mathcal{W}$ is a finite set. This is in stark contrast to the standard information-bottleneck method, where the decision variables of the optimization problem are the (real-valued) probability mass values of the conditional distribution $p(t|x)$. As a result, the curve characterizing the trade-off between the maximal amount of retainable information and compression in the standard information-bottleneck method is a concave function, whereas in our problem it is a discrete set of points.

In light of the above observations, it is natural to ask whether or not solutions obtained by solving the integer program (42)-(45) are members of the set of trees that characterize the curve in Fig. 4. Unfortunately, this cannot be guaranteed, in

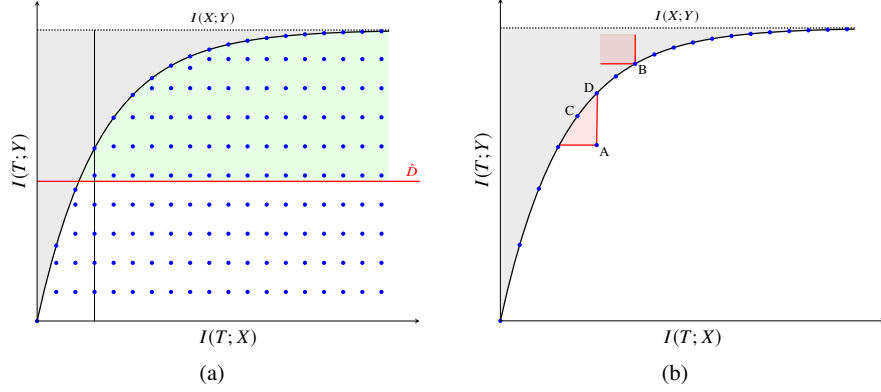


Fig. 5: (a) Information plane showing the feasible region $\{(\mathbf{z}^\top \Delta_X, \mathbf{z}^\top \Delta_Y) : \mathbf{z} \in \mathcal{D}_{\hat{D}}\}$ (green), and a horizontal line (red) drawn at the given value of $\hat{D}$. (b) Set of Pareto optimal values (blue points along black curve) with select solutions (trees) shown.

general, as the integer program does not order solutions based on retained relevant information. To understand why this is the case, consider solving the problem (42)-(45) for a given value of $\hat{D} \geq 0$, where the set of feasible solutions is given by

$$\mathcal{D}_{\hat{D}} = \{\mathbf{z} \in \mathbb{R}^{|\mathcal{N}_{\text{int}}(\mathcal{T}_W)|} : (43)-(45) \text{ are satisfied}\}. \quad (50)$$

The feasible set in the information plane for this instance of the problem is shown in Fig. 5a. Notice that the value of $\mathbf{z}^\top \Delta_X$ is constant along vertical lines in the information plane. As a result, minimizing the cost $\mathbf{z}^\top \Delta_X$ as in (42) amounts to finding an element $\mathbf{z} \in \mathcal{D}_{\hat{D}}$ such that the vertical line passing through $\mathbf{z}$ in the information plane is as close to the origin as possible. However, as seen in Fig. 5a, there may be multiple feasible trees that minimize $\mathbf{z}^\top \Delta_X$. More specifically, we see in the example in Fig. 5a that there are two feasible points along the vertical line intersecting the x-coordinate at the minimum value of $\mathbf{z}^\top \Delta_X$. Since the optimization problem (42)-(45) does not order solutions based on their retained relevant information, the two feasible solutions along the vertical line in Fig. 5a are equivalent with respect to the cost function of the integer program (42)-(45). Consequently, either one of these two solutions may be returned when executing a suitable search algorithm, and thus the result of the ILP need not be the most informative tree among those that attain the minimum of (42). Despite this, we aim throughout the remainder of this section to develop an approach toward characterizing the set of trees that maximally trade information retention and compression for a given environment $\mathcal{W}$. Devising a method for obtaining this set is useful, as it allows for a more meaningful understanding of the overall performance of various abstraction algorithms applied to generate abstractions of $\mathcal{W}$, and also to elucidate the intrinsic trade-off between information retention and compression of a given environment (i.e., model complexity vs. model quality). In our endeavor to

characterize the set of trees that best trade information retention and compression, we leverage ideas from multi-objective optimization theory, which we turn to discuss next.

As the preceding discussion exemplified, it is insufficient to consider only the degree of achieved compression ($\mathbf{z}^\top \Delta_X$) or the amount of retained information ($\mathbf{z}^\top \Delta_Y$) as the sole representative of the quality of a candidate solution when searching for solutions that maximize compression and relevant information retention. Instead, we aim to quantify the goodness of a solution based on *two* measures of quality, namely *both* the compression *and* retained relevant information. To this end, let the set

$$\mathcal{I}_{\hat{D}} = \{(\mathbf{z}^\top \Delta_X, \mathbf{z}^\top \Delta_Y) : \mathbf{z} \in \mathcal{D}_{\hat{D}}\}, \quad (51)$$

be the collection of all feasible compression-relevance pairs for a given value of $\hat{D}$. Our goal is then to select a feasible tree $\mathbf{z} \in \mathcal{D}_{\hat{D}}$ such that the pair $(\mathbf{z}^\top \Delta_X, \mathbf{z}^\top \Delta_Y) \in \mathcal{I}_{\hat{D}}$ simultaneously maximizes compression (minimizes $\mathbf{z}^\top \Delta_X$) and the retained relevant information (maximize $\mathbf{z}^\top \Delta_Y$). However, searching for an optimal realization of the vector $(\mathbf{z}^\top \Delta_X, \mathbf{z}^\top \Delta_Y) \in \mathcal{I}_{\hat{D}}$ over $\mathbf{z} \in \mathcal{D}_{\hat{D}}$ is a multi-objective optimization problem, where feasible solutions $\mathbf{z} \in \mathcal{D}_{\hat{D}}$ are not totally ordered with respect to the vector-valued objective $(\mathbf{z}^\top \Delta_X, \mathbf{z}^\top \Delta_Y)$ [4]. For example, consider two candidate solutions $\mathbf{z}, \hat{\mathbf{z}} \in \mathcal{D}_{\hat{D}}$ where $\mathbf{z}^\top \Delta_X < \hat{\mathbf{z}}^\top \Delta_X$ and $\mathbf{z}^\top \Delta_Y < \hat{\mathbf{z}}^\top \Delta_Y$. In this scenario, the solution $\mathbf{z}$ achieves a greater degree of compression but retains less information as compared to $\hat{\mathbf{z}}$. As a result, the solutions $\mathbf{z}$ and $\hat{\mathbf{z}}$ are incomparable, since which of these solutions we prefer depends on our relative preference of compression and information retention.

Being that solutions in multi-objective optimization problems are generally partially ordered, the goal of multi-objective problems must be modified in comparison to their single (scalar) objective optimization counterparts [4]. To this end, methods for multi-objective optimization problems aim to obtain solutions that are *Pareto efficient* [4, 11]. In our problem setting, a feasible tree $\mathbf{z} \in \mathcal{D}_{\hat{D}}$ is Pareto efficient if there does not exist a $\hat{\mathbf{z}} \in \mathcal{D}_{\hat{D}}$ such that $(\hat{\mathbf{z}}^\top \Delta_X, -\hat{\mathbf{z}}^\top \Delta_Y) \leq (\mathbf{z}^\top \Delta_X, -\mathbf{z}^\top \Delta_Y)$ holds with strict inequality in at least one component (i.e., either $\hat{\mathbf{z}}^\top \Delta_X < \mathbf{z}^\top \Delta_X$ or $-\hat{\mathbf{z}}^\top \Delta_Y < -\mathbf{z}^\top \Delta_Y$, or both). The collection of vectors $(\mathbf{z}^\top \Delta_X, \mathbf{z}^\top \Delta_Y) \in \mathbb{R}^2$ for all Pareto efficient solutions $\mathbf{z} \in \mathcal{D}_{\hat{D}}$ is called the *set of Pareto optimal values* [4]. An example is shown in Fig. 5b. From Fig. 5b, we observe that trees that have information plane points in the red region offer improvement in at least one attribute (compression or information retention) as compared to the solution at the intersection of the red lines. For example, the tree A offers less compression and less retained information than the tree C, and the same compression, but less information retained, as compared with tree D. Notice that Pareto efficient trees, such as the tree corresponding to B, do not have any feasible points in the red region other than themselves and possibly other solutions (trees) that are equivalent in the information plane (i.e., trees that are the same in terms of compression and information retention). Trees corresponding to points B, C, and D are all Pareto efficient and are not comparable.

To formalize the notion of Pareto efficiency consider any $\mathbf{z} \in \mathcal{D}_{\hat{D}}$, and define the set

$$\mathcal{S}_{\hat{D}}(\mathbf{z}) = \{\hat{\mathbf{z}} \in \mathcal{D}_{\hat{D}} : \hat{\mathbf{z}}^\top \Delta_Y \geq \mathbf{z}^\top \Delta_Y \text{ and } \hat{\mathbf{z}}^\top \Delta_X \leq \mathbf{z}^\top \Delta_X\}. \quad (52)$$

For any tree $\mathbf{z} \in \mathcal{D}_{\hat{D}}$, the set $\mathcal{S}_{\hat{D}}(\mathbf{z})$ contains those trees $\hat{\mathbf{z}}$ that satisfy any one of the following conditions: (i) contain the same amount of relevant information for the given amount of compression as compared to the tree $\mathbf{z}$, (ii) provide more relevant information for the same level of compression, (iii) achieve a greater degree of compression for the same amount of relevant information as the tree $\mathbf{z}$, or (iv) provide both more relevant information and greater degree of compression as compared to the tree $\mathbf{z}$. Also, notice that the set $\mathcal{S}_{\hat{D}}(\mathbf{z})$ contains those trees that have the same amount of relevant information and achieve an identical degree of compression (i.e., share a common point in the information plane). To identify the trees that share a common point in the information plane, we define for any $\mathbf{z} \in \mathcal{D}_{\hat{D}}$ the set

$$\mathcal{U}_{\hat{D}}(\mathbf{z}) = \{\hat{\mathbf{z}} \in \mathcal{D}_{\hat{D}} : \hat{\mathbf{z}}^\top \Delta_X = \mathbf{z}^\top \Delta_X \text{ and } \hat{\mathbf{z}}^\top \Delta_Y = \mathbf{z}^\top \Delta_Y\}. \quad (53)$$

Thus, the trees $\hat{\mathbf{z}} \in \mathcal{S}_{\hat{D}}(\mathbf{z}) \setminus \mathcal{U}_{\hat{D}}(\mathbf{z})$ achieve a greater degree of compression or retain more relevant information (or both), while being no worse in either attribute as compared to $\mathbf{z} \in \mathcal{D}_{\hat{D}}$. The set of points $\mathbf{z} \in \mathcal{D}_{\hat{D}}$ for which there does not exist any tree $\hat{\mathbf{z}} \in \mathcal{D}_{\hat{D}}$ that offers improvement in either compression or information retention, while being no worse in either attribute, are those trees in the set

$$\mathcal{P}_{\hat{D}} = \{\mathbf{z} \in \mathcal{D}_{\hat{D}} : \mathcal{S}_{\hat{D}}(\mathbf{z}) \setminus \mathcal{U}_{\hat{D}}(\mathbf{z}) = \emptyset\}. \quad (54)$$

Consequently, the collection of all Pareto efficient solutions is given by

$$\mathcal{P} = \{\mathbf{z} : \mathbf{z} \in \mathcal{P}_{\hat{D}}, \hat{D} \geq 0\}, \quad (55)$$

and the corresponding set of Pareto optimal values is

$$\mathcal{V} = \{(\mathbf{z}^\top \Delta_X, \mathbf{z}^\top \Delta_Y) : \mathbf{z} \in \mathcal{P}\}. \quad (56)$$

Given the mathematical description of the set of Pareto optimal values, we are now interested in the computational aspects of obtaining the set $\mathcal{V}$. Interestingly, obtaining Pareto efficient trees and tracing the set of Pareto optimal values for our problem can be done in a principled manner, as follows.

6.2 Obtaining Pareto Efficient Trees

Given value of $\hat{D} \geq 0$, consider the optimization problem

$$D^* \triangleq \min_{\mathbf{z}} \mathbf{z}^\top \Delta_X, \quad (57)$$

subject to

$$\mathbf{z}^\top \Delta_Y \geq \hat{D}, \quad (58)$$

$$[\mathbf{z}]_{t'} - [\mathbf{z}]_t \leq 0, \quad t \in \mathcal{B}, \quad t' \in C(t), \quad (59)$$

$$[\mathbf{z}]_t \in \{0, 1\}, \quad t \in \mathcal{N}_{\text{int}}(\mathcal{T}_W), \quad (60)$$

which is the same as (42)-(45) and repeated here for convenience. As discussed previously, the solution to the above optimization problem is not necessarily a Pareto efficient solution, as there may be a more informative tree for the same level of compression available in the feasible set $\mathcal{D}_{\hat{D}}$. We are, however, guaranteed that there are no trees containing at least $\hat{D}$ units of relevant information that offer more compression than D^* units. Thus, in order to obtain a Pareto efficient tree, we must check to see if there are any other solutions that contain more relevant information with D^* units of compression. To this end, we consider the following equality constrained integer program

$$\hat{D}^* \triangleq \max_{\mathbf{z}} \mathbf{z}^\top \Delta_Y, \quad (61)$$

subject to the constraints

$$\mathbf{z}^\top \Delta_X = D^*, \quad (62)$$

$$[\mathbf{z}]_{t'} - [\mathbf{z}]_t \leq 0, \quad t \in \mathcal{B}, \quad t' \in C(t), \quad (63)$$

$$[\mathbf{z}]_t \in \{0, 1\}, \quad t \in \mathcal{N}_{\text{int}}(\mathcal{T}_W). \quad (64)$$

It is important to observe that the feasible set of the above linear program is not empty, since the existence of a tree $\mathbf{z} \in \mathcal{D}_{\hat{D}}$ for which $\mathbf{z}^\top \Delta_X = D^*$ is known from the solution of (42)-(45). The resulting pair $(D^*, \hat{D}^*) \in \mathcal{V}$ is in the set of Pareto optimal values, and the solution to the problem (61)-(64) is a member of the set $\mathcal{P}$.

One can then obtain the entire set of Pareto optimal values by performing the aforementioned process for $\hat{D} \in [0, I(X; Y)]$. Having a method for obtaining the set of Pareto optimal values, we are able to quantify the solutions from various abstraction methods (e.g., integer programming, Q-tree search, etc.) with respect to trees that offer the best trade-off between compression and retained (relevant) information, as we will show in the numerical example that follows. However, before we present some numerical results, we briefly discuss an alternate approach to solving the program (42)-(45) that is obtained by removing the integer constraints (45).

6.3 Linear Programing Relaxation

The optimization problems (42)-(45) and (46)-(49) fall within the realm of combinatorial optimization due to the integer constraint placed on the decision variable $\mathbf{z}$ [27, 1]. As a result, standard gradient and sub-gradient based methods for optimization are not applicable due to the discrete nature of the feasible set of solutions. Instead, one may observe that by removing (or relaxing) the integrality constraints (45) and (49) we obtain a linear program, which can be solved by em-

employing efficient convex optimization algorithms developed in the past decades. It is then natural to investigate if relaxing the integrality constraints, thereby circumnavigating the discrete, combinatorial aspects of optimization problem, can result in reasonable solutions (i.e., multi-resolution trees) to our problem. To this end, the *linear programming (LP) relaxation* [27, 1] of the problem (42)-(45) is given by

$$\min_{\mathbf{z}} \mathbf{z}^\top \Delta_X, \quad (65)$$

subject to the constraints

$$\mathbf{z}^\top \Delta_Y \geq \hat{D}, \quad (66)$$

$$[\mathbf{z}]_{t'} - [\mathbf{z}]_t \leq 0, \quad t \in \mathcal{B}, \quad t' \in C(t), \quad (67)$$

$$0 \leq [\mathbf{z}]_t \leq 1, \quad t \in \mathcal{N}_{\text{int}}(\mathcal{T}_W). \quad (68)$$

In the problem (65)-(68), the vector $\mathbf{z} \in \mathbb{R}^{|\mathcal{N}_{\text{int}}(\mathcal{T}_W)|}$ is not restricted to have integer entries, as is the case in the integer programming formulation (42)-(45). Consequently, the problem (65) subject to (66)-(68) is a linear (convex) program for which traditional approaches for optimization (e.g., the simplex method) may be applied in order to obtain a solution [27]. The LP relaxation for the integer problem (46)-(49) can be formulated analogously.

Notice that since the LP relaxation removes the integer constraint (45), we can no longer guarantee that the solution to the relaxed problem (65)-(68) corresponds to a valid tree representation of $\mathcal{W}$. Therefore, we must post-process the solution vector from the LP relaxation in order to obtain a valid multi-resolution tree depiction of the world $\mathcal{W}$. To this end, assume $\mathbf{z}^* \in \mathbb{R}^{|\mathcal{N}_{\text{int}}(\mathcal{T}_W)|}$ is a solution to the LP relaxation (65)-(68), where the entries $[\mathbf{z}^*]_t$ are not necessarily integer. Now, we construct a multi-resolution tree representation from the vector $\mathbf{z}^*$ by considering the function $f_\delta : \mathbb{R} \rightarrow \{0, 1\}$ defined according to the rule

$$f_\delta(z) = \begin{cases} 1, & \text{if } z \geq \delta, \\ 0, & \text{otherwise,} \end{cases} \quad (69)$$

and let the *extension* of f_δ to $\mathbb{R}^n$ be the function $\tilde{f}_\delta : \mathbb{R}^n \rightarrow \{0, 1\}^n$ defined by

$$\tilde{f}_\delta(\mathbf{z}) = (f_\delta([\mathbf{z}]_1), \dots, f_\delta([\mathbf{z}]_n)). \quad (70)$$

An integer vector $\tilde{\mathbf{z}}^*$ can then be constructed from $\mathbf{z}^*$ according to $\tilde{\mathbf{z}}^* = \tilde{f}_\delta(\mathbf{z}^*)$. Observe that creating the vector $\tilde{\mathbf{z}}^*$ from $\mathbf{z}^*$ according to (70) not only yields an integer vector, but also guarantees that $\tilde{\mathbf{z}}^*$ corresponds to a valid tree representation of $\mathcal{W}$. To see why this is the case, we note that the constraint (67) implies that the values $[\mathbf{z}^*]_{t'} \in \mathbb{R}$ and $[\mathbf{z}^*]_t \in \mathbb{R}$ satisfy $[\mathbf{z}^*]_{t'} \leq [\mathbf{z}^*]_t$ for all $t \in \mathcal{B}$, $t' \in C(t)$. Then, since the function f_δ is monotonic, we have $[\tilde{\mathbf{z}}^*]_{t'} = f_\delta([\mathbf{z}^*]_{t'}) \leq f_\delta([\mathbf{z}^*]_t) = [\tilde{\mathbf{z}}^*]_t$, and thus $[\tilde{\mathbf{z}}^*]_{t'} - [\tilde{\mathbf{z}}^*]_t \leq 0$ holds for all $t \in \mathcal{B}$ and $t' \in C(t)$. As a result, the vector $\tilde{\mathbf{z}}^* = \tilde{f}_\delta(\mathbf{z}^*)$ is integer and represents a valid tree representation, since the constraints $[\tilde{\mathbf{z}}^*]_{t'} - [\tilde{\mathbf{z}}^*]_t \leq 0$, $t \in \mathcal{B}$, $t' \in C(t)$ and $[\tilde{\mathbf{z}}^*]_t \in \{0, 1\}$, $t \in \mathcal{N}_{\text{int}}(\mathcal{T}_W)$

are both satisfied. However, creating the integer solution vector $\tilde{\mathbf{z}}^*$ according to $\tilde{\mathbf{z}}^* = \tilde{f}_\delta(\mathbf{z}^*)$ from the solution $\mathbf{z}^*$ to the LP relaxation (65)-(68) does not guarantee that the resulting multi-resolution representation contains at least $\hat{D}$ units of relevant information. In other words, it is not assured that the integer vector $\tilde{\mathbf{z}}^* = \tilde{f}_\delta(\mathbf{z}^*)$ satisfies the constraint (66). Despite this drawback, we observe that the LP relaxation method described above performs very well in simulation, as we shall see in the numerical examples we present next.

7 Numerical Examples and Discussion

In this section, we present two examples that demonstrate the utility of our framework for generating task-relevant abstractions. For each example, we provide a sample of abstractions obtained by the integer program and discuss the quality of the solutions in the information plane. In addition, the set of Pareto optimal values for each case is shown, and the integer program formulation developed in this chapter is compared to other approaches for tree abstraction, such as LP relaxation (see Section 6.3), Q-tree search [23] and k-class tree pruning [22]. The results are discussed in detail and brought into context of the topics discussed throughout this chapter. We employ the Gurobi CVX solver [15, 14] to solve the information-theoretic integer program formulated in Section 5.

We first consider the satellite image shown in Fig. 6a, which is of dimension 128×128 ($\ell = 7$). In this example, the relevant variable $Y : \Omega \rightarrow \{0, 1\}$ is assumed to be the color intensity of cells, where the outcome $Y = 0$ corresponds to low intensity (lighter color) and the outcome $Y = 1$ corresponds to high color intensity (darker color). In addition, we let the uncompressed (original signal) variable be $X : \Omega \rightarrow \{1, \dots, |\mathcal{N}_{\text{leaf}}(\mathcal{T}_W)|\}$, whose outcomes x are the individual grid cells of the original image. We assume the map represents the conditional distribution $p(y = 1|x)$, where each x is an outcome (grid cell) of X , and $p(y = 1|x) \in [0, 1]$ is the color intensity of the cell $x \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_W)$. In general, the color intensity encoded by the map may be representative of a number of important aspects of the world, such as elevation data or probabilistic occupancy information.⁶ In both of these cases, the agent may not be able to determine, with certainty, the elevation data or occupancy information of grid cells due to limited on-board computational resources in processing camera images in order to create the environment map [37]. Furthermore, agents may need to abstract, or compress, environment maps due a number of factors, including limited availability of on-board memory resources to store environment representations, or to transmit environment maps to other agents (e.g., satellite ground stations) across capacity-limited communication channels.

The input distribution $p(x, y)$ is then created according to $p(x, y) = p(y|x)p(x)$, where we assume throughout this chapter that the distribution $p(x)$ is uniform (i.e., $p(x) = 1/|\mathcal{N}_{\text{leaf}}(\mathcal{T}_W)|$). By choosing the distribution $p(x)$ to be uniform, we perform

⁶ In this case, $p(y = 1|x)$ represents the probability that the cell $x \in \mathcal{N}_{\text{leaf}}(\mathcal{T}_W)$ is occupied. See [23, 24, 33] for more details on occupancy grids and abstraction.

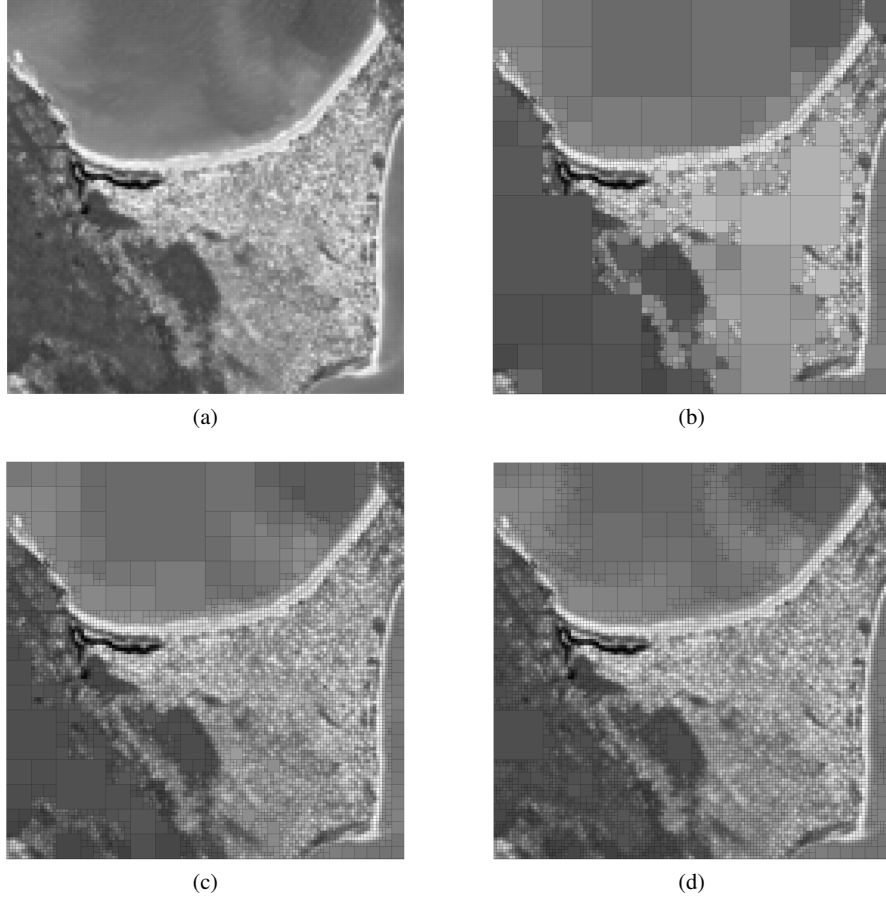


Fig. 6: (a) Original, full resolution, 128×128 image of the environment $\mathcal{W}$, (b)-(d) abstractions of $\mathcal{W}$ obtained by solving the integer program for a range of $\hat{D}$ values. (b) Abstraction for $\hat{D} = 0.0555$ nats ($\hat{D}/I(X;Y) = 0.965$), which retains $I_Y(\mathcal{T})/I(X;Y) = 0.9650$ of the available relevant information and contains 34.9% of the nodes compared to the full resolution image, (c) Abstraction for $\hat{D} = 0.0574$ nats ($\hat{D}/I(X;Y) = 0.998$), which retains $I_Y(\mathcal{T})/I(X;Y) = 0.9984$ of the available relevant information and contains 67.6% of the nodes compared to the full resolution image, (d) abstraction for $\hat{D} = 0.0575$ nats ($\hat{D}/I(X;Y) = 0.999$), which retains $\hat{D}/I(X;Y) = 0.999$ of the available relevant information and contains 76.9% of the nodes as compared to the full resolution image. Notice that, in this example, with only approximately 35% of the nodes we obtain an abstraction that retains 96.5% of the relevant information.

what we call *region agnostic* abstraction, whereby our framework will not favor any region of the map over another when generating the multi-resolution trees. However,

while we assume $p(x)$ is uniform in this chapter for the purposes of demonstrating the utility of our approach, *any* valid distribution $p(x)$ may be used in our framework. The use of a non-uniform $p(x)$ will lead to *region-specific* abstraction, as areas of the map with no support through $p(x)$ do not contain any relevant information, irrespective of the distribution $p(y|x)$. Consequently, employing a non-uniform $p(x)$ will result in refinement only in the regions for which $p(x) > 0$. For more information regarding region-specific abstraction in the context of our framework, we refer the interested reader to [23].

By solving the integer program (42)-(45) for a range of $\hat{D} \geq 0$ values, we obtain the abstractions shown in Figs. 6b-6d. From Figs. 6b-6d we see a number of interesting observations. First, notice that, as the value of $\hat{D} \geq 0$ is increased, the resolution of the environment representation increases. This is expected, since the terms driving compression and information retention are in direct conflict, and thus the emergence of more informative solutions comes at the expense of a greater amount of X -information (i.e., less compression). Secondly, observe that, at lower values of $\hat{D}$, our framework retains those features of the image that are the most salient, leaving less noteworthy, finer-scale, details of $\mathcal{W}$ to emerge at greater values of $\hat{D}$. To be more specific, consider the abstractions shown in Fig. 6b and Fig. 6d, where the former of these images is obtained at a lower value of $\hat{D}$ than the latter. While the abstraction Fig. 6b contains less relevant information than that in Fig. 6d, the two representations both show regions of the image where there is a stark difference in color intensity with great detail.

To understand why this is happening, we examine relations (36) and (38). The first of these equations, namely (36), shows how the values of $p(y|t)$ for any node $t \in \mathcal{N}_{\text{int}}(\mathcal{T}_{\mathcal{W}})$ is computed from knowledge of $p(y|t')$ of its children $t' \in \mathcal{C}(t)$. The second relation, that of (38), then displays how the distributions $p(y|t)$ and $p(y|t')$ for $t \in \mathcal{N}_{\text{int}}(\mathcal{T}_{\mathcal{W}})$, $t' \in \mathcal{C}(t)$ are utilized to compute the incremental change in relevant information contributed by node t , given by $\Delta \hat{I}_Y(t)$. From these relations, we see that two components are required to compute $\Delta \hat{I}_Y(t)$ for any $t \in \mathcal{N}_{\text{int}}(\mathcal{T}_{\mathcal{W}})$; namely $p(t)$ and $\text{JS}_{\Pi}(p(y|t'_1), \dots, p(y|t'_4))$. Observe that, from the properties of the JS-divergence, the more diverse the distributions $p(y|t')$ for $t' \in \mathcal{C}(t)$ are from their parent distribution $p(y|t)$, the greater the value of $\text{JS}_{\Pi}(p(y|t'_1), \dots, p(y|t'_4))$. Moreover, from the relation (38) we see that nodes $t \in \mathcal{N}_{\text{int}}(\mathcal{T}_{\mathcal{W}})$ that have the greatest changes in incremental information, $\Delta \hat{I}_Y(t)$, are those nodes t that have large values of $p(t)$ and a high divergence (diversity) of the distributions $p(y|t')$ of their children $t' \in \mathcal{C}(t)$, as measured by the JS-divergence. In the context of our example this means that merging cells that are diverse with respect to their grey-scale intensity will have large values of $\text{JS}_{\Pi}(p(y|t'_1), \dots, p(y|t'_4))$, whereas aggregating cells that are equal in their grey-scale intensity (i.e., are the same shade of grey), that is, $p(y|t'_1) = \dots = p(y|t'_4)$, will result in $\text{JS}_{\Pi}(p(y|t'_1), \dots, p(y|t'_4)) = 0$. Intuitively, when $p(y|t'_1) = \dots = p(y|t'_4)$, one's ability to predict the value of the color intensity is equally as good from the aggregated node as if the cells were refined and the resolution were increased. In contrast, when there is a greater color intensity difference between nodes that are aggregated (i.e., $\text{JS}_{\Pi}(p(y|t'_1), \dots, p(y|t'_4))$ is relatively large) then information regarding the color intensity of the merged nodes is lost, and can be

recovered by splitting the aggregate cell and by increasing the resolution in the region. It is for the aforementioned reasons that regions in which there is a large color intensity difference are represented with relatively high resolution at lower values of $\hat{D}$, whereas those areas with homogenous color intensities remain merged, or aggregated, to larger super nodes even at greater values of $\hat{D}$.

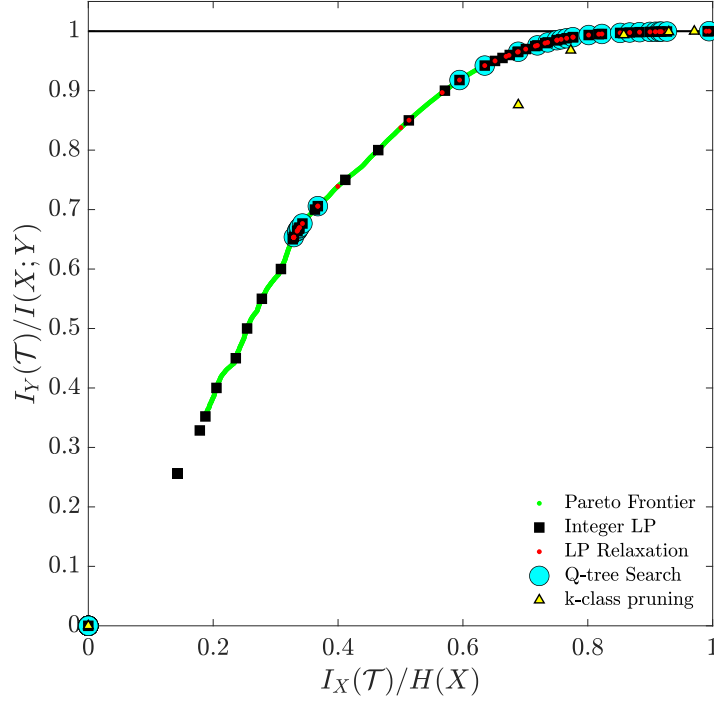


Fig. 7: Normalized information plane for the environment shown in Fig. 6a. The figure contains solutions (trees) obtained from the integer program, Q-tree search, LP relaxation and k-class tree pruning. In order to generate a valid multi-resolution tree representation of $\mathcal{W}$ from the LP relaxation result, we apply the rule (70) with $\delta = 0.5$.

From the results in Figs. 6b-6d, we also see that our framework finds multi-resolution trees that retain more than 95% of the relevant information and contain fewer than 35% of the nodes in the original representation. Moreover, nearly all the relevant information can be retained by employing an abstraction that contains 13% fewer nodes than the full resolution image (see, Fig. 6d). These results represent a significant reduction in the number of nodes employed in the reduced representation, while sacrificing little in terms of relevant information retention. To obtain a more holistic understanding of the trade-off between relevant information retention and compression for the environment in Fig. 6a, we show in Fig. 7 solutions obtained

from the integer program (42)-(45) in the normalized information plane. In addition to the results obtained from executing the integer program, we also show in Fig. 7 information plane points for other algorithms for abstraction, including Q-tree search, k-class trees and LP relaxation as well as the set of Pareto optimal values obtained by the method described in Section 6.1.

From Fig. 7 we observe that, in this example, the solutions returned by the LP relaxation, Q-tree search and the integer programming approaches⁷ are all Pareto efficient. The same cannot be said, however, regarding the k-class tree solutions, which lie below the collection of Pareto optimal values. As a result, we conclude that the k-class tree pruning abstractions are sub-optimal, since the trees that are obtained from the k-class pruning approach retain less relevant information than trees returned by the integer programming developed in this chapter. This is expected, as the k-class tree pruning method is an ad-hoc approach towards generating tree abstractions that is not driven by information-theoretic principles of information retention and compression.

Notice also that the solutions from the Q-tree search, integer programming and LP relaxation all show good agreement in the information plane. As we increase the value of $\hat{D}$, we move to the right in the information plane, obtaining solutions that retain more information regarding the relevant random variable, at the expense of achieving less compression in the resulting multi-resolution representation. Thus, when $\hat{D} = 0$, we obtain the trivial (root) tree $\mathcal{R}_{\mathcal{W}}$ (which retains no relevant information, but maximizes compression), and as $\hat{D} \rightarrow I(X;Y)$ trees that retain all the relevant information emerge as the solution to the integer program. We recover a family of solutions (trees) for values of $\hat{D}$ between these two extremes. Similarly, the Q-tree search solutions move to the right in the information plane as the value of $\beta > 0$ is increased, rendering trees that retain all the available relevant information as $\beta \rightarrow \infty$. In the Q-tree search method, the trivial (root) abstraction is returned as β approaches 0. Recall, however, that in contrast to the Q-tree search approach, the integer-programming method developed in this chapter does not require the guesswork of tuning the $\beta > 0$ parameter when designing abstractions to meet, for example, channel-capacity or memory resource constraints. Instead, in the integer programming approach, these constraints can be directly incorporated into the problem formulation a priori, allowing for task-relevant abstractions to be obtained that are guaranteed to meet the imposed resource constraints.

Lastly, we notice that there are points in the information plane where the solution to the integer program “jumps” from one point to another. For example, no integer programming solutions are found between the information plane points of the origin $(0, 0)$ (i.e., the trivial abstraction) and $(0.1429, 0.2559)$. The resulting “kink” is due to the discrete nature of the solution space. Specifically, in this example, no solutions are obtained between the origin and $(0.1429, 0.2559)$ as, from the root tree/node $\mathcal{R}_{\mathcal{W}}$, only one expansion is permitted – namely, expanding the root itself. Expanding

⁷ We note that the worst-case time complexity of the ILP and Q-tree search algorithms are as follows: ILP is NP-complete [27, Ch.15] and Q-tree search is $O(|\mathcal{N}_{\text{leaf}}(\mathcal{T}_{\mathcal{W}})|)$ [23, pp.1680-1681]. Comparable performance of the two algorithms was observed for the examples presented in this chapter.

the root results in a finite amount of information gain, captured by $\Delta\hat{I}_X(t)$ and $\Delta\hat{I}_Y(t)$, respectively, where $t = \mathcal{R}_W$. Thus, for $\hat{D} \in (0, \Delta\hat{I}_Y(t)]$ (where $t = \mathcal{R}_W$), the resulting solution to the integer program will be the root-expanded tree, resulting in a jump in the information plane from $(0, 0)$ to $(\Delta\hat{I}_X(t), \Delta\hat{I}_Y(t)) = (0.1429, 0.2559)$, as seen in Fig. 7.

We also apply our approach to the environment shown in Fig. 8a, which is of dimension 128×128 ($\ell = 7$). In order to provide a useful discussion comparing and contrasting the results obtained by employing our framework to design abstractions for this new environment, we again assume $p(x)$ is a uniform distribution and $p(x, y) = p(y|x)p(x)$, where the map encodes the conditional distribution $p(y|x)$. By solving the information-theoretic integer program for various values of $\hat{D}$, we obtain the abstractions shown in Figs. 8b-8d.

From the results in Figs. 8b-8d we see a number of trends that are similar to what it was observed in the prior example. To this end, we first notice that, as the value of $\hat{D}$ is increased, the resolution of the resulting abstract representation increases. Furthermore, from Fig. 8 we observe that employing an abstraction with only approximately 20% of the resolution of the original image, we are able to retain more than 95% of the relevant information. Consequently, many of the cells from the original image may be merged with little degradation in ones ability to predict the color intensity of cells. Moreover, we see that at lower values of $\hat{D}$, our framework finds trees that maintain high resolution in areas where there is a large change in color intensity between finest-resolution cells. For example, notice that the multi-resolution tree depicted in Fig. 8b aggregates cells in areas where there is relatively small change in color intensity (e.g., the homogeneously dark or lighter portions of the map) while regions along the boundaries of light to dark cells are shown in high resolution. As the value of $\hat{D}$ is increased, the priority of multi-resolution abstractions becomes retaining as much information as possible, resulting in refinement of those regions where there is a color intensity difference, but that were aggregated at lower values of $\hat{D}$. Lastly, notice that the dark region of the map remains aggregated into super cells even at larger values of $\hat{D}$. Per our previous discussion regarding the incremental change in relevant information, we note that this occurs as areas of homogeneous color intensity do not contain any relevant information, and thus the dark portion of the map remains aggregated into as large cells as possible while ensuring the resulting multi-resolution tree is a valid quadtree representation of $\mathcal{W}$.

Shown in Fig. 9 are the information plane points obtained from applying other aggregation algorithms, such as Q-tree search and k-class tree pruning, to the example environment in Fig. 8a. We again observe good agreement between the solutions obtained from the Q-tree search, LP relaxation and integer programming approaches. Interestingly, we see from the integer program solutions shown in Fig. 9 that there is one identifiable solution that is not Pareto efficient. Notice that the third and fourth integer solutions both have the same value of $I_X(\mathcal{T})$ (that is, they are along the same vertical line in the information plane), however, the fourth solution contains more relevant information than the third. As a result, the third solution is not Pareto efficient (see Section 6.1), since there is another feasible tree that, for the same

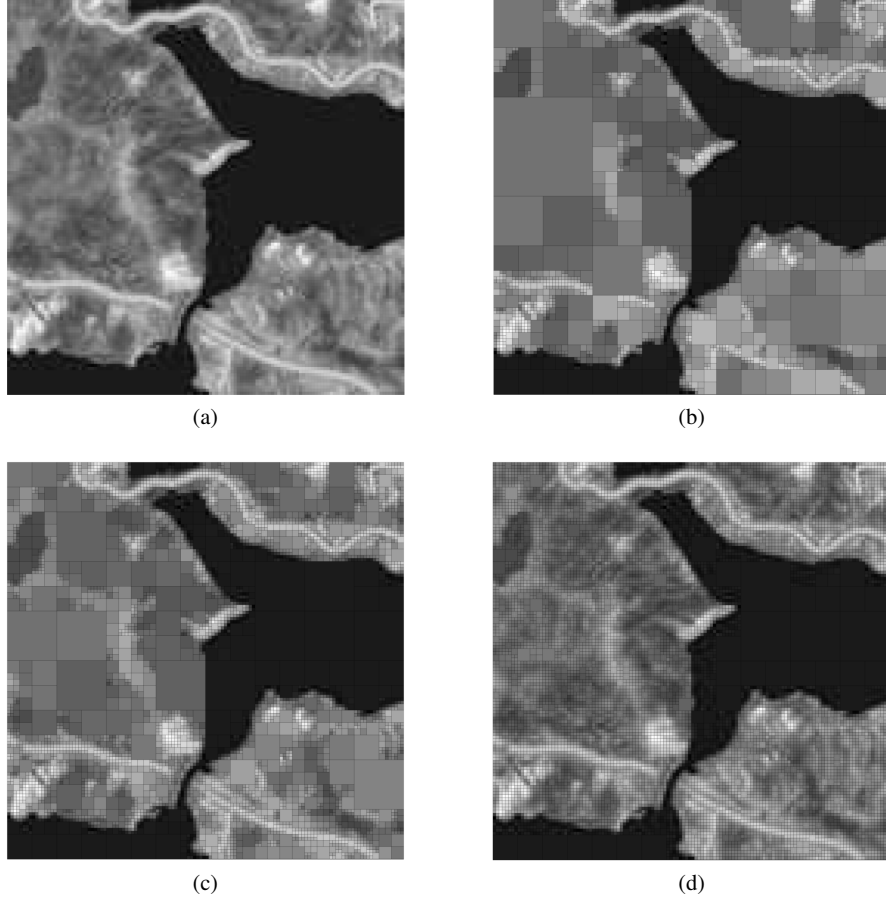


Fig. 8: (a) Original, full resolution, 128×128 image of the environment $\mathcal{W}$, (b)-(d) abstractions of $\mathcal{W}$ obtained by solving the integer program for a range of $\hat{D}$ values. (b) Abstraction for $\hat{D} = 0.0954$ nats ($\hat{D}/I(X;Y) = 0.959$), which retains $I_Y(\mathcal{T})/I(X;Y) = 0.959$ of the available relevant information and contains 20.2% of the nodes as compared to the full resolution image, (c) Abstraction for $\hat{D} = 0.0981$ nats ($\hat{D}/I(X;Y) = 0.985$), which retains $I_Y(\mathcal{T})/I(X;Y) = 0.985$ of the available relevant information and contains 35.5% of the nodes as compared to the full resolution image, (d) Abstraction for $\hat{D} = 0.0995$ nats ($\hat{D}/I(X;Y) = 0.99$), which retains $I_Y(\mathcal{T})/I(X;Y) = 0.99$ of the available relevant information and contains 79.4% of the nodes as compared to the full resolution image.

amount of compression, retains more relevant information. We note that this is the only one of the solutions obtained and displayed in Fig. 9 that are not Pareto efficient, apart from those obtained via the k-class tree pruning method. Furthermore, we once

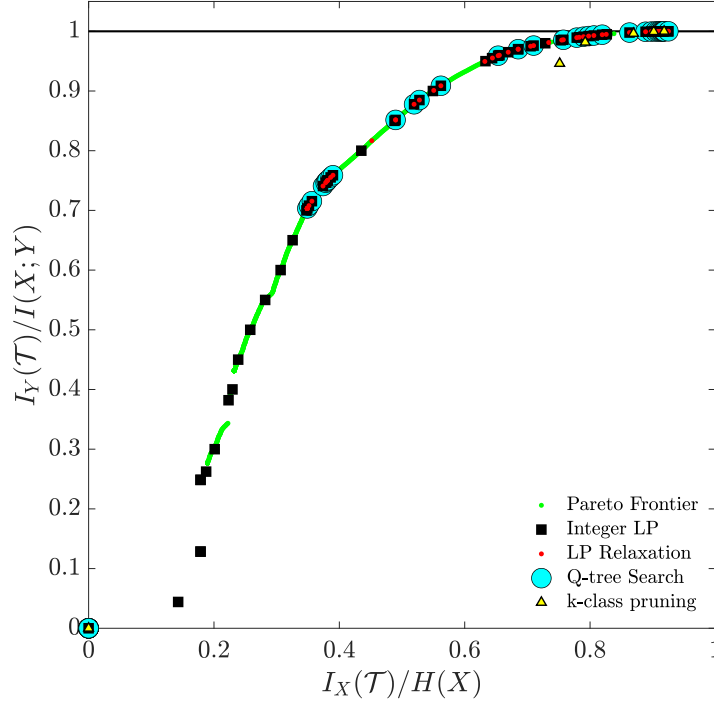


Fig. 9: Normalized information plane for the environment shown in Fig. 8a. The figure contains solutions (trees) obtained from the integer program, Q-tree search, LP relaxation and k-class tree pruning. In order to generate a valid multi-resolution tree representation of $\mathcal{W}$ from the LP relaxation result, we apply the rule (70) with $\delta = 0.5$.

again observe that the k-class approach results in trees that are sub-optimal in their balance between compression and information retention. Moreover, as was the case in Fig. 7, we notice that the solutions returned by the k-class tree method are grouped at the extreme ends of the information plane. That is, the k-class solutions jump from the origin (which is the trivial compression obtained when only a single class is employed) to the upper right region of the information plane, which is the area where high-cardinality, highly informative solutions lie. Consequently, while the k-class approach allows one to reduce memory storage requirements by creating abstractions of the environment, it does not allow for a diverse family of solutions to be obtained that gradually trade compression and information retention. In contrast, our approach allows for a more principled trade-off between information retention and compression.

8 Conclusion

In this chapter, we considered the problem of designing task-relevant abstractions of operating environments for autonomous systems. The abstractions emerge as a function of agent resource constraints and are in the form of multi-resolution, hierarchical, tree structures that are not provided a priori. To design the abstractions, we employ an information-theoretic approach that leverages the information bottleneck (IB) principle to formulate an optimization problem that searches for encodings (abstractions) of the environment that are maximally retentive regarding task-relevant information while maximizing compression. In order to rigorously formulate our problem, we elucidate the connections between signal encoders and multi-resolution tree structures and detail how a tree abstraction problem can be viewed as an optimal encoder search, akin to information-theoretic approaches of signal compression.

Moreover, we show how designing information-theoretically driven tree abstractions that are maximally retentive regarding task-relevant information can be realized as an integer linear program by exploiting the structure of the problem. The properties of the resulting integer program are discussed in detail, and a method for linear programming relaxation is presented, thereby allowing information-theoretic multi-resolution tree abstractions to be obtained by solving a (convex) linear optimization problem. Numerical examples are presented that show the utility of the approach. Furthermore, we compare and contrast our approach to existing methods for abstraction, including Q-tree search [23] and k-class tree pruning [22]. The numerical results are discussed in detail and show the usefulness of our approach for generating task-relevant abstractions for resource-constrained autonomous systems.

References

1. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: Network Flows. Pearson (1993)
2. Bertsekas, D.: Dynamic programming and optimal control. Athena Scientific (2005)
3. Bondy, J.A., Murty, U.S.R.: Graph Theory with Applications. Macmillan Education UK (1976)
4. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge university press (2004)
5. Chechik, G., Tishby, N.: Extracting relevant structures with side information. In: Conference on Neural Information Processing Systems, pp. 881–888. Vancouver, BC, CA (2002). December 9–14, 2002
6. Cover, T.M., Thomas, J.A.: Elements of Information Theory, 2nd edn. John Wiley & Sons (2006)
7. Cowlagi, R.V.: Hierarchical motion planning for autonomous aerial and terrestrial vehicles. Ph.D. thesis, Georgia Institute of Technology (2011)
8. Cowlagi, R.V., Tsiotras, P.: Multiresolution path planning with wavelets: A local replanning approach. In: American Control Conference, pp. 1220–1225. Seattle, WA, USA (2008). June 11–13, 2008
9. Cowlagi, R.V., Tsiotras, P.: Multi-resolution path planning: Theoretical analysis, efficient implementation, and extensions to dynamic environments. In: IEEE Conference on Decision and Control, pp. 1384–1390. Atlanta, GA, USA (2010). December 15–17, 2010
10. Cowlagi, R.V., Tsiotras, P.: Multiresolution motion planning for autonomous agents via wavelet-based cell decompositions. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) **42**(5), 1455–1469 (2012)

11. Das, I., Dennis, J.E.: A closer look at drawbacks of minimizing weighted sums of objectives for pareto set generation in multicriteria optimization problems. *Structural Optimization* **14**(1), 63–69 (1997)
12. Einhorn, E., Schröter, C., Gross, H.M.: Finding the adequate resolution for grid mapping - cell sizes locally adapting on-the-fly. In: *IEEE Conference on Robotics and Automation*, pp. 1843–1848. Shanghai, CN (2011). May 9–13, 2011
13. Gilad-Bachrach, R., Navot, A., Tishby, N.: An information theoretic tradeoff between complexity and accuracy. In: *Learning Theory and Kernel Machines*, pp. 595–609. Springer Berlin Heidelberg (2003)
14. Grant, M., Boyd, S.: Graph implementations for nonsmooth convex programs. In: V. Blondel, S. Boyd, H. Kimura (eds.) *Recent Advances in Learning and Control, Lecture Notes in Control and Information Sciences*, pp. 95–110. Springer-Verlag Limited (2008)
15. Grant, M., Boyd, S.: CVX: Matlab software for disciplined convex programming, version 2.1. <http://cvxr.com/cvx> (2014)
16. Hauer, F.: Path-Planning Algorithms in High-Dimensional Spaces. Ph.D. thesis, Georgia Institute of Technology (2019)
17. Hauer, F., Kundu, A., Rehg, J.M., Tsiotras, P.: Multi-scale perception and path planning on probabilistic obstacle maps. In: *IEEE International Conference on Robotics and Automation*, pp. 4210–4215. Seattle, WA, USA (2015). May 26–30, 2015
18. Hauer, F., Tsiotras, P.: Reduced complexity multi-scale path-planning on probabilistic maps. In: *IEEE International Conference on Robotics and Automation*, pp. 83–88. Stockholm, SE (2016). May 16–21, 2016
19. Holte, R.C., Choueiry, B.Y.: Abstraction and reformulation in artificial intelligence. *Philosophical Transactions of the Royal Society of London B: Biological Sciences* **358**(1435), 1197–1204 (2003)
20. Hornung, A., Wurm, K.M., Bennewitz, M., Stachniss, C., Burgard, W.: Ocotomap: An efficient probabilistic 3d mapping framework based on octrees. *Autonomous Robots* **34**, 189–206 (2013)
21. Kambhampati, S., Davis, L.S.: Multiresolution path planning for mobile robots. *IEEE Journal of Robotics and Automation* **RA-2**(3), 135–145 (1986)
22. Kraetzschmar, G.K., Gassull, G.P., Uhl, K.: Probabilistic quadrees for variable-resolution mapping of large environments. *IFAC Proceedings Volumes* **37**(8), 675–680 (2004)
23. Larsson, D.T., Maity, D., Tsiotras, P.: Q-tree search: An information-theoretic approach toward hierarchical abstractions for agents with computational limitations. *IEEE Transactions on Robotics* **36**(6), 1669–1685 (2020)
24. Larsson, D.T., Maity, D., Tsiotras, P.: Information-theoretic abstractions for planning in agents with computational constraints. *IEEE Robotics and Automation Letters* **6**(4), 7651–7658 (2021)
25. Larsson, D.T., Maity, D., Tsiotras, P.: Information-theoretic abstractions for resource-constrained agents via mixed-integer linear programming. In: *Proceedings of the Workshop on Computation-Aware Algorithmic Design for Cyber-Physical Systems*, pp. 1–6. Nashville, TN, USA (2021). May 19–21, 2021
26. Lin, J.: Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory* **37**(1), 145–151 (1991)
27. Papadimitriou, C.H., Steiglitz, K.: *Combinatorial Optimization: Algorithms and Complexity*. Courier Corporation (1998)
28. Ponsen, M., Taylor, M.E., Tuyls, K.: Abstraction and generalization in reinforcement learning: A summary and framework. In: *Adaptive and Learning Agents*, pp. 1–32. Springer Berlin Heidelberg (2010)
29. Slonim, N.: The information bottleneck: Theory and applications. Ph.D. thesis, The Hebrew University (2002)
30. Slonim, N., Tishby, N.: Agglomerative information bottleneck. In: *Conference on Neural Information Processing Systems*, pp. 617–623. Denver, CO, USA (1999). November 29 - December 4, 1999
31. Strouse, D., Schwab, D.J.: The deterministic information bottleneck. *Neural Computation* **29**(6), 1611–1630 (2017)

32. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press (1998)
33. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics. MIT Press (2006)
34. Tishby, N., Pereira, F.C., Bialek, W.: The information bottleneck method. In: Allerton Conference on Communication, Control and Computing, pp. 368–377. Monticello, IL, USA (1999). September 22–24, 1999
35. Tsiotras, P., Bakolas, E.: A hierarchical on-line path planning scheme using wavelets. In: European Control Conference, pp. 2806–2812. Kos, GR (2007). July 2–5, 2007
36. Tsiotras, P., Jung, D., Bakolas, E.: Multiresolution hierarchical path-planning for small UAVs using wavelet decompositions. *Journal of Intelligent & Robotic Systems* **66**(4), 505–522 (2011)
37. Wang, Y., Lai, Z., Huang, G., Wang, B.H., van der Maaten, L., Campbell, M., Weinberger, K.Q.: Anytime stereo image depth estimation on mobile devices. In: International Conference on Robotics and Automation, pp. 5893–5900. Montreal, QC, CA (2019). May 20–24, 2019
38. Zucker, J.: A grounded theory of abstraction in artificial intelligence. *Philosophical Transactions of the Royal Society of London, Series B: Biological Sciences* **358** (1435), 1293–1309 (2003)