Rate–Distortion Approach to Databases: Storage and Content-Based Retrieval

Ertem Tuncel, *Member, IEEE*, Prashant Koulgi, *Student Member, IEEE*, and Kenneth Rose, *Fellow, IEEE*

***Abstract*—This paper investigates the relationship between rate–distortion theory and efficient content-based data retrieval from high-dimensional databases. We consider database design as the encoding of a data object sequence, and retrieval from the database as the decoding of the sequence using side information (i.e., the query) available only at the decoder. We show that, in this setting, the optimal asymptotic tradeoff between the search time R_s (bits per data object read from the storage device) and the expected search accuracy D_s (relevance of the retrieved data set) is given by the Wyner–Ziv solution with a side-information-dependent distortion measure. Moreover, the data indexing and retrieval problem is, in general, inseparable from the data compression problem. Data items selected by the search procedure, which can be stored in the disk with a limited total rate of $R_r \geq R_s$, need to be presented at a prescribed expected reconstruction quality D_r. This is, hence, a problem of scalable source coding or successive refinement, albeit with differing layer distortion measures to quantify search and reconstruction quality, respectively. We derive a single-letter characterization of all achievable quadruples $\{R_s, R_r, D_s, D_r\}$, and prove conditions for "successive refinability" without rate loss. Finally, we show that the special case $D_s = D_r = 0$ is nontrivial and of practical interest in this context, as it can impose "acceptable" search and reconstruction qualities for each individual data item and for the entire query space with high probability, in contradistinction with standard average distortion requirements. The region of achievable $\{R_s, R_r\}$ is obtained by adapting Rimoldi's characterization to a new regular scalable coding problem.**

***Index Terms*—Approximate similarity searching, content-based retrieval, databases, scalable coding, successive refinability without rate loss, Wyner–Ziv problem, zero–one distortion measures.**

I. Introduction

A. Motivation

THIS work was originally motivated by several observations regarding central problems in database management and their relation to problems of source coding as well as to fundamental results in rate–distortion theory. An immediate first observation is that the design of coding systems for static databases eliminates one of the pitfalls experienced in virtually all other compression applications, namely, performance generalization outside the training set. Simply stated, in the case of compression for storage in a database one may train the code on exactly the same set of data that will actually be compressed.

More important to this work is the fact that researchers working on efficient retrieval from databases have come to realize that their main challenge is the handling of very large and high-dimensional databases, e.g., multimedia databases. A highly motivating observation is that the search and retrieval (or indexing) problem bears resemblance to the rate–distortion problem in that it seeks an optimal tradeoff between the amount of data that needs to be read during the search ("rate") and the quality of the retrieved data in terms of its relevance to the query ("distortion"). Moreover, since very large databases are at the focus of intense database research, it is of great interest to identify and characterize the asymptotic (in size of the database) performance bounds, which are naturally related to standard rate–distortion theoretic results.

We further observe that, in real-world applications, one must jointly handle search performance and compression performance. On the one hand, it is desired to have the search as efficient as possible in terms of the search time and quality tradeoff. On the other hand, there is the question of the reproduction quality of the individual data items. (The data cannot be reproduced losslessly as there is an inherent storage capacity barrier.) It turns out that this combined problem is equivalent to a scalable coding problem. The base layer is the "search" layer and expends the minimal rate needed to secure a certain level of search quality. The enhancement layer uses more rate to refine the reproduction of the data items. Note that this scalable coding problem involves different distortion measures at the two levels.

Based on the preceding initial observations, it is the premise of this paper that information-theoretic approaches may offer highly valuable insight into and performance bounds for important problems in storage and retrieval in large databases. Moreover, the database context offers a new setting and a variety of problems that would be of interest to researchers in rate–distortion theory.

B. Approximate Similarity Searching and Rate–Distortion Theory

Similarity search refers to the task of seeking in a database the entries that are most similar to a given query object. This problem is central in a wide range of applications in multimedia

Manuscript received October 24, 2002; revised July 31, 2003. This work was supported in part by the National Science Foundation under Grants EIA-9986057 and EIA-0080134, the University of California MICRO program, Dolby Laboratories, Inc., Lucent Technologies, Inc., Microsoft Corporation, Mindspeed Technologies, and Qualcomm, Inc. The material in this paper was presented in part at the IEEE International Symposium on Information Theory, Lausanne, Switzerland, June/July 2002.

E. Tuncel is with the Department of Electrical Engineering, University of California, Riverside, CA 92521 USA (e-mail: ertem@ee.ucr.edu).

P. Koulgi was with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106-9560 USA (e-mail: prashant_koulgi@yahoo.co.in).

K. Rose is with the Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106-9560 USA (e-mail: rose@ece.ucsb.edu).

Communicated by R. Zamir, Associate Editor for Source Coding.

Digital Object Identifier 10.1109/TIT.2004.828068

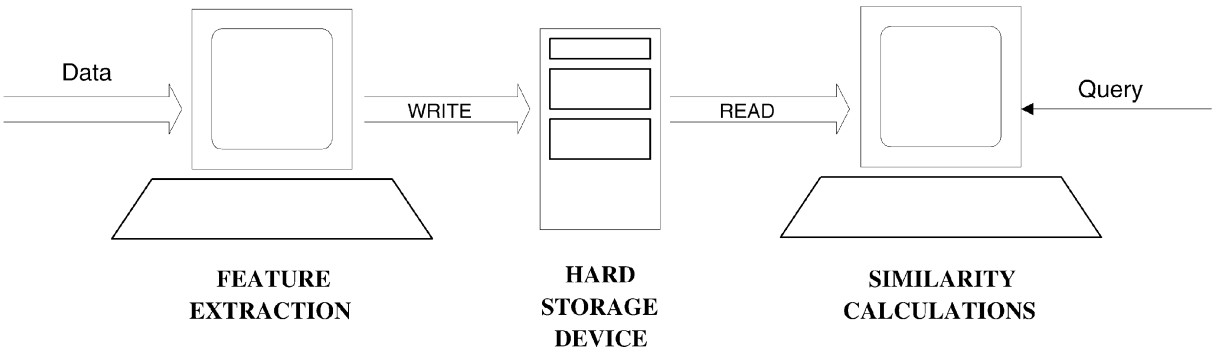


Fig. 1. The block diagram of the similarity search system.

databases, which may contain images, video, music, etc. The degree of similarity between two objects is often quantified by a distance measure, e.g., the Euclidean distance, operating on *feature vectors* extracted from the data. The user submits a query object to a search engine, and may either provide a distance threshold ϵ or the number of objects k to be returned. These types of queries are called the ϵ-range query, and the k-nearest-neighbor (k-NN) query, respectively.

In typical multimedia applications, the volume of data is huge, and the feature vectors are of high dimensionality. For example, for color-based similarity queries, a color histogram of B^3 dimensions (i.e., B bins per color component) is used. Therefore, even a modest value of B, e.g., $B = 5$, results in 125-dimensional feature vectors. It is in fact impractical to store all the extracted feature vectors in a random access memory (RAM) and, therefore, it is necessary to read them from a hard storage medium, typically a hard disk, during the search operation. Since input/output (I/O) operations for hard storage devices are slow, the time spent accessing the feature vectors overwhelmingly dominates the time complexity of the search. A sketch of the query processing system is provided in Fig. 1.

A powerful tool for time complexity reduction is *indexing*: the feature space is divided into subregions (usually using a tree structure), boundaries of which are also stored on the disk (see, for example, R-trees [9], kdb-trees [13], X-trees [1]). Fig. 2(a) depicts the main working principle of indexing. Basically, the purpose of the index is to narrow the scope of the search by pruning irrelevant data objects based on the boundaries of the subregions they fall into. However, these pruning techniques work well only for low-dimensional applications, e.g., searching on a geographic or road map, and tend to scale poorly to higher dimensional applications [3], [17], such as those involving multimedia databases. It was in fact shown in [17] that even the simple sequential scan shown in Fig. 2(b) outperforms all existing indexing techniques, as the dimensionality increases beyond even moderate values (around 10).[1]

On the other hand, it was observed that significant savings in disk I/O costs compared to the system in Fig. 2(b) are possible if one allows for *approximate* search results (see [7], [14], [16].) Usually, the extraction of feature vectors from the data objects is itself a heuristic process that attempts to approximately capture relevant information. Moreover, even if the feature vectors represented the original data perfectly and losslessly, users would still differ in their perception systems, and hence in their similarity expectations. Thus, rather than incur the extremely high cost of an exact result, it is more cost-effective to develop a fast search engine that outputs an approximate set.

We adopt in this work a widely accepted and very efficient approach for approximate similarity searching. Without building an indexing mechanism, the search engine simply accesses *partial* information about *all* the feature vectors. Popular examples of this approach are the VA-file algorithm [16], and the dimensionality reduction techniques [4], [10]. Feature vectors are approximated using the accessed partial information, thereby trading search quality for processing time. From the source coding point of view, this corresponds to quantization. Since the bottleneck for query processing is the number of disk I/O operations, the processing time becomes approximately proportional to the rate R at which the sequence of feature vectors are quantized. Fig. 2(c) shows the working principle of the adopted approximate searching scheme. Note that the original feature vectors do not have to be stored, as they will never be used.

Since the database is very large, an additional *storage complexity* constraint arises, and it becomes necessary to store the data also in compressed form. One way to do this is to compress the data separately from the feature vectors. However, as the compressed feature vectors already carry some information about the data they are extracted from, their description could be embedded into the description of the data and be viewed as the base layer of a corresponding scalable coder. This scalable coding approach, also demonstrated in Fig. 2(c), will certainly improve the total storage performance. It is important to keep in mind that the quality of compression is measured very differently in the two layers: the first layer performance is measured by the accuracy of the search, while the second layer objective is accurate reconstruction. Also, the first-layer rate reflects the time complexity, whereas the second layer rate determines the total storage size.

Practical databases are of finite size for obvious reasons. However, we idealize the database as a random and infinite sequence of data objects, drawn independently from a fixed distribution, and measure the time complexity and the total

[1]Even when the index structure cannot prune any data, and therefore the system has to retrieve all data objects, sequential scan is still advantageous because the tree-structured nature of the index dictates many *random seek* operations on the disk. In terms of time complexity, one random seek operation is equal to retrieving around 100 kbytes sequentially.

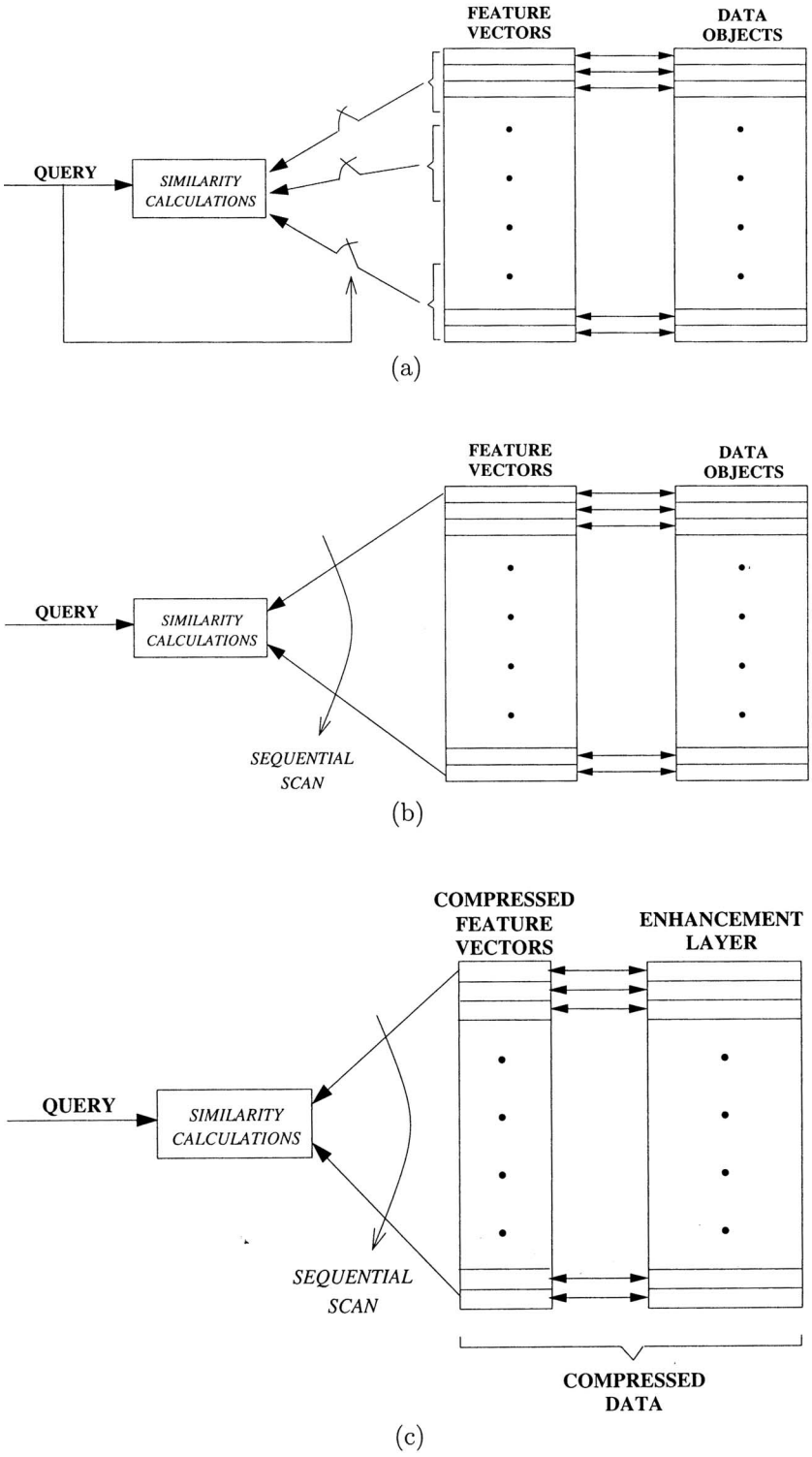


Fig. 2. (a) The indexing mechanism where the decision of whether or not to retrieve a feature vector is solely based on which group it belongs to. (b) Sequential scan of all feature vectors. (c) Sequential scan of compressed feature vectors, which constitute the first layer of the proposed scalable coding scheme.

storage size in *normalized* form, i.e., per data object. The source coding intuition promises an improved performance for the system if data vectors are compressed jointly, and the performance reaches its maximum as the number of jointly encoded data objects tends to infinity. Therefore, as shown in Fig. 3, we consider coders that operate on "data blocks" of length n, and analyze the performance of the system when $n \to \infty$. Of course, the limiting achievable performance region obtained by such an analysis is also useful as an outer bound on the performance of the system with finite n.

C. Summary of Results

In this paper, we look at the performance of the above system in a rate–distortion theoretic framework. Four competing per-

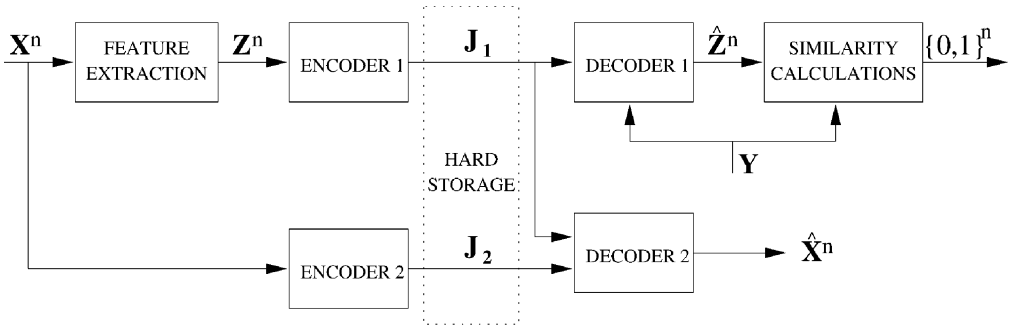


Fig. 3. A block-based scalable source coding look at the system in Fig. 2(c). Note the Wyner–Ziv setup of the first layer.

formance parameters[2] are the time complexity R_s, *total* storage complexity R_r, search distortion D_s, and reconstruction distortion D_r. Recall that time complexity is proportional to the amount of bits per feature vector retrieved from the disk, i.e., the first-layer rate in Fig. 3. We omit the constant factor of "seconds per bit," and measure time complexity directly in bits, in order to compare it with storage complexity. At the second layer of scalable coding applications, one can alternatively consider the *differential* rate $R_r - R_s$ as the relevant performance parameter. In the proposed application, however, the total rate R_r at the second layer is more appealing for our purposes as it has a direct physical meaning (storage complexity). We model the quality of the search by a distortion function $\Delta(z, \hat{z}, y)$, where z is the original feature vector, $\hat{z}$ is the quantized version of z, and y denotes the query point given by the user. Since feature extraction is a deterministic process, we can equivalently consider a distortion function $d_s(x, \hat{z}, y)$, where x is the data point from which z was extracted. The quality of the reconstruction at the second layer is captured by an ordinary distortion measure $d_r(x, \hat{x})$, where $\hat{x}$ denotes the reproduction for x.

We consider distortion measures $\Delta(z, \hat{z}, y)$ which penalize the undesirable cases where y is close to only one of the vectors z and $\hat{z}$. Effectively, this type of measure penalizes false hits, i.e., inclusion of irrelevant data in the answer set, or false dismissals, i.e., exclusion of relevant data from the answer set. The notion of "closeness" is quantified by a distance function $\rho(z, y)$, which is assumed to capture the similarity between the data and the query point. Fig. 4 illustrates this concept. For the feature vectors $z_1, \ldots, z_4$ and query y shown in the figure, $\Delta(z_1, \hat{z}_1, y)$ and $\Delta(z_2, \hat{z}_2, y)$ yield high values, and $\Delta(z_3, \hat{z}_3, y)$ and $\Delta(z_4, \hat{z}_4, y)$ yield low values. Note the contrast with classical (query-independent) distortion measures where $d(z, \hat{z}) \approx \rho(z, \hat{z})$, which would imply $d(z_3, \hat{z}_3) \gg d(z_1, \hat{z}_1) \approx d(z_4, \hat{z}_4)$.

We first analyze the rate–distortion performance in the first layer. We show that the minimum achievable asymptotic rate $R_s(D_s)$ for a prescribed search quality D_s is given by the rate–distortion function for a special case of the well-known Wyner–Ziv problem [18]: lossy source coding, where the decoder has access to side information. The correspondence of the first layer with the Wyner–Ziv setup is explicit in Fig. 3. The side information known to the decoder (but not to the encoder) is the query point y. The encoder only has statistical

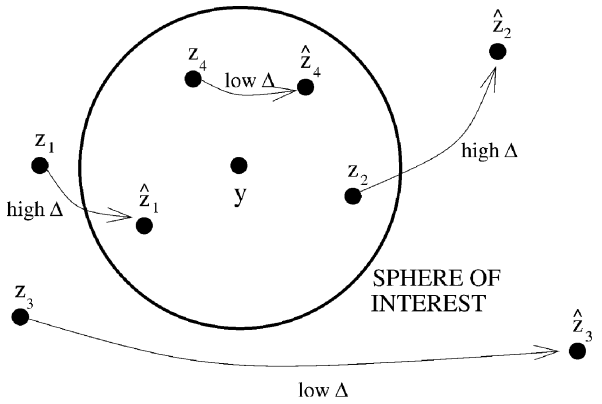


Fig. 4. Illustration of the behavior of query-dependent distortion measures.

knowledge in the form of the distribution of queries $P_Y(y)$ which may, in practice, be approximated from the query history. What makes compression of feature vectors a "special" case of the Wyner–Ziv problem is that the side information is a single random query instead of a sequence as in the original problem. Also, note that the query is *independent* of the sequence to be coded[3] since the user of the search engine is assumed to generate queries that are independent of the actual database entries. Despite this independence, the query distribution can be exploited to reduce the distortion, because the distortion measure is a function of the side information (see [11] for discussion of side-information-dependent distortion measures).

There is also a strong connection between the first layer of our system and the robust descriptions system proposed in [8]. The query-dependent reconstruction of the feature vector can be equivalently performed by several decoders, each corresponding to different query points y, where the distortion at each decoder is measured according to the ordinary measure $d_s(\cdot, \cdot, y)$. The difference in our exposition is that we consider the *average* distortion achieved by all such decoders (where the average is taken over the query alphabet), whereas in robust descriptions, the output quality of each decoder is evaluated separately.

We next derive the region of all achievable quadruples $\{R_s, R_r, D_s, D_r\}$. As with other scalable coding scenarios, an interesting question is whether the source is successively refinable without rate loss [6] at distortion levels D_s and D_r,

[2]Here and in the sequel, the subscripts s and r, respectively, refer to the *search* performance and the *reconstruction* performance.

[3]However, this does not preclude the possibility that queries and stored data be generated from the same probability distribution.

i.e., whether the quadruple $\{R_s(D_s), R_r(D_r), D_s, D_r\}$ is achievable.[4] In the classical scalable source coding setting, the meaning of this desirable property is that users with different bandwidths can be served simultaneously without compromising the quality of the system. In our scenario, on the other hand, it implies that the search over the database can be done optimally, i.e., with search accuracy D_s and time complexity $R_s(D_s)$, while keeping the total storage requirement at its theoretical minimum $R_r(D_r)$ subject to a prescribed reconstruction distortion D_r. Thus, successive refinement without rate loss trivializes the tradeoff of time versus storage complexity, as the theoretical minimum for both quantities can be reached simultaneously. The necessary and sufficient condition for successive refinability without rate loss follows as a corollary from the region of achievable $\{R_s, R_r, D_s, D_r\}$. The resultant condition involves Markovian properties similar to the well-known condition derived in [6] for the classical successive refinement problem.

Finally we consider examples where d_s and d_r assume only the values 0 and 1, and we set $D_s = D_r = 0$. In the classical scalable rate–distortion analysis, the case $D_s = D_r = 0$ is of little interest, since it only accounts for lossless coding, and hence eliminates the need for refinement. However, in our scenario, with proper choices of d_s and d_r, this corresponds to enforcing with high probability "good enough" search and reconstruction qualities for each *individual* data point and for *all* points in the query space, in contradistinction with enforcing distortion values D_s and D_r computed by averaging over a *block* of data points and over the query space. We show that in this setting, the region of achievable $\{R_s, R_r, 0, 0\}$ is obtained by adapting Rimoldi's characterization [12] to a new variant of the classical scalable coding problem.

II. Preliminaries and Problem Formulation

Let $\mathcal{X}$, $\mathcal{Y}$, and $\mathcal{Z}$ represent the alphabets for data objects, queries, and feature vectors, respectively. We denote by $X_t \in \mathcal{X}$, for $t = 1, 2, \ldots$ the random data sequence, by $Y \in \mathcal{Y}$ the random query vector, and by $Z_t \in \mathcal{Z}$ the feature vector deterministically extracted from X_t. Also, let $\hat{\mathcal{X}}$ and $\hat{\mathcal{Z}}$, respectively, denote data object reproduction and feature vector reproduction alphabets. In many cases of interest, $\mathcal{Y} = \mathcal{Z} = \hat{\mathcal{Z}}$ and $\mathcal{X} = \hat{\mathcal{X}}$. We restrict our attention to the case where $\mathcal{X}$, $\hat{\mathcal{X}}$, $\mathcal{Z}$, $\hat{\mathcal{Z}}$, and $\mathcal{Y}$ are all *finite* alphabets.

Although X_t, $\hat{X}_t$, Z_t, $\hat{Z}_t$, and Y may, in general, be vectors in some space, we will consider them as "letters" of the corresponding super-alphabets $\mathcal{X}$, $\hat{\mathcal{X}}$, $\mathcal{Z}$, $\hat{\mathcal{Z}}$, and $\mathcal{Y}$. We assume that the data objects are collected independently from the same distribution, i.e., X_t are independent and identically distributed (i.i.d.) $\sim P_X(x)$. For example, consider a web crawler creating a database by independently collecting random pictures from random websites, or a government database where biometric data of individuals are added in the same (random) order they apply for a driver license. Also, note that Y is independent of $\{X_t\}_{t=1}^{\infty}$.

[4]Here $R_s(D_s)$ denotes the minimum time complexity achieving search distortion D_s, whereas $R_r(D_r)$ denotes the ordinary rate–distortion function at distortion D_r.

We introduce a query-dependent distortion measure

$$\Delta : \mathcal{Z} \times \hat{\mathcal{Z}} \times \mathcal{Y} \longrightarrow [0, \infty)$$

in order to capture the dependence of quantization quality on the query point $y \in \mathcal{Y}$. Since feature extraction is a deterministic process, z is determined by x. Therefore, we can equivalently consider $d_s : \mathcal{X} \times \hat{\mathcal{Z}} \times \mathcal{Y} \longrightarrow [0, \infty)$. Before describing the compression scheme, we provide below examples demonstrating how the distortion measures $\Delta(z, \hat{z}, y)$ or $d_s(x, \hat{z}, y)$ might be chosen in practice to capture the notion of search accuracy.

A. Example A

Consider $\mathcal{X} = \hat{\mathcal{X}} = \{0, \ldots, K\}^M$ and let the feature $z(x)$ be the unnormalized *empirical distribution* vector (commonly referred to as *type* in information theory, or as *histogram* in multimedia searching terminology) of $x \in \mathcal{X}$. In other words

$$\mathcal{Z} = \left\{ (z_0, \ldots, z_K) \in \mathbb{N}^{(K+1)} : \sum_{i=0}^{K} z_i = M \right\}$$

and $\mathcal{Y} = \hat{\mathcal{Z}} = \mathcal{Z}$.

A well-known fact is that the number of distinct types are "polynomially many" in M [5]. More specifically

$$|\mathcal{Z}| \leq (M+1)^{K+1}.$$

Therefore, it suffices to use $(K+1)\log_2(M+1)$ bits to *losslessly* describe any feature vector $z \in \mathcal{Z}$. Following the common intuition borrowed from the method of types, the number of distinct types $|\mathcal{Z}|$ may be perceived to be negligibly small compared to the total number of distinct data objects $|\mathcal{X}| = (K+1)^M$. This, in turn, would trivialize the task of scalable coding at hand, because then the feature vectors can be losslessly encoded with zero rate for all practical purposes. However, that intuition is correct only for long data vectors defined on small alphabets, i.e., $K \ll M$, while there are many database examples where $K \approx M$ and even $K \gg M$. For instance, consider document search on the web, where $z \in \mathcal{Z}$ is the vector of word counts. In that context, K is the number of distinct words in English and M is the number of words in a typical document. Another example is image search based on similarity of color histograms, where $K = 2^{24}$ and $M = 2^{20}$ per megapixel.

Let

$$\rho(y, z) = \frac{1}{2} \sum_{i=0}^{K} |y_i - z_i|.$$

That is, the users are interested in finding the data vector whose empirical distribution is close to y (say, within a fixed ϵ neighborhood of it). The extra factor of $1/2$ is not necessary but is included since $\sum_i |y_i - z_i|$ is always an even integer. Note that for $K = 1$, the empirical distribution is equivalent to the Hamming weight, and $\rho(y, z) = |y_1 - z_1|$.

Assuming that the search engine outputs data points for which $\rho(y, \hat{z}) \leq \epsilon$, the search accuracy criterion

$$\Delta(z, \hat{z}, y) = \begin{cases} 1, & \rho(y, z) \leq \epsilon \text{ and } \rho(y, \hat{z}) > \epsilon \\ 1, & \rho(y, z) > \epsilon \text{ and } \rho(y, \hat{z}) \leq \epsilon \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

is a natural choice since it exactly penalizes inclusion of irrelevant data in (false hits), or exclusion of relevant data from (false dismissals), the answer set. The more relaxed criterion

$$\Delta(z,\hat{z},y)=\begin{cases}1, & \rho(y,z)\le\epsilon-\alpha \text{ and } \rho(y,\hat{z})>\epsilon\\ 1, & \rho(y,z)>\epsilon+\beta \text{ and } \rho(y,\hat{z})\le\epsilon\\ 0, & \text{otherwise}\end{cases} \quad (2)$$

for some $\alpha\ge 0$, $\beta\ge 0$, penalizes only false hits that are farther than $\epsilon+\beta$ from, and false dismissals that are closer than $\epsilon-\alpha+1$ to, the given query point y. All data with distance to the query point between $[\epsilon-\alpha+1,\epsilon+\beta]$ are considered as "don't-care" points.

Depending on the application, one can generalize (2) by introducing unequal penalty terms for false hits and false dismissals, rather than penalize both by 1. Another (perhaps more important) direction of generalization is to average $\Delta(z,\hat{z},y)$ over p_ϵ, the distribution of ϵ, which reflects the frequency of range queries with "radius" ϵ

$$\Delta(z,\hat{z},y)=\sum_{\epsilon}p_\epsilon\Delta_\epsilon(z,\hat{z},y) \quad (3)$$

where $\Delta_\epsilon(z,\hat{z},y)$ denotes the distortion measure for a fixed ϵ, given by (2). Note that (3) requires only statistical rather than exact knowledge of ϵ during the design stage, and therefore gives the user freedom to choose ϵ together with the query point y.

B. Example B

Consider the same setup as in Example A. Although it may be desirable to use a distortion measure as in (3), suppose we do not know the distribution p_ϵ in advance. We observe that assuming $\alpha=\beta$, $\Delta(z,\hat{z},y)$ in (2) yields 0 if $|\rho(y,z)-\rho(y,\hat{z})|\le\alpha$, regardless of the value of ϵ. Moreover, if $|\rho(y,z)-\rho(y,\hat{z})|>\alpha$, then there exists an ϵ such that $\Delta(z,\hat{z},y)$ yields 1. Therefore, another possibility for the search accuracy criterion is given by

$$\Delta(z,\hat{z},y)=\begin{cases}1, & |\rho(y,z)-\rho(y,\hat{z})|>\alpha\\ 0, & \text{otherwise.}\end{cases} \quad (4)$$

C. Problem Formulation

In the design phase, the data objects are grouped into blocks of length n and encoded using the block encoder

$$f_1:\mathcal{X}^n\longrightarrow\mathcal{M}_1$$

which is a combination of feature extraction and compression. The compressed bit descriptions $f_1(X^n)$ are then written to disk sequentially. In the query processing phase, the whole set of feature vectors are reconstructed sequentially using the block decoder

$$g_1:\mathcal{M}_1\times\mathcal{Y}\longrightarrow\hat{\mathcal{Z}}^n.$$

Observe that for different queries the decoding of the same compressed description $f_1(X^n)$ may be performed differently because the distortion measure is query dependent. Therefore, despite its independence from the data, the query can be exploited to reduce the distortion. The resultant time complexity and expected search accuracy are given by $\frac{1}{n}\log|\mathcal{M}_1|$ and $E\{d_s(X^n,g_1(f_1(X^n),Y),Y)\}$, respectively, where

$$d_s(x^n,\hat{z}^n,y)=\frac{1}{n}\sum_{t=1}^{n}d_s(x_t,\hat{z}_t,y).$$

Definition 1: A pair (R_s,D_s) is **achievable** if for all $\delta>0$, there exists a block encoder $f_1:\mathcal{X}^n\longrightarrow\mathcal{M}_1$ and a decoding function $g_1:\mathcal{M}_1\times\mathcal{Y}\longrightarrow\hat{\mathcal{Z}}^n$ such that

$$\frac{1}{n}\log|\mathcal{M}_1|\le R_s+\delta \quad (5)$$

$$E\{d_s(X^n,g_1(f_1(X^n),Y),Y)\}\le D_s+\delta. \quad (6)$$

This is almost exactly the Wyner–Ziv problem with a side-information-dependent distortion measure [11], but with the exception that the side information Y is *not* a sequence, but rather, a single random variable. Nevertheless, as we show in Section III, the rate–distortion region is given by the Wyner–Ziv characterization [11], [18].

Consider next the more demanding requirement that the expected search accuracy conditioned on each query point $y\in\mathcal{Y}$ be less than or equal to D_s.

Definition 2: A pair (R_s,D_s) is **strongly achievable** if for all $\delta>0$, there exists an encoding function $f_1:\mathcal{X}^n\longrightarrow\mathcal{M}_1$ and a decoding function $g:\mathcal{M}_1\times\mathcal{Y}\longrightarrow\hat{\mathcal{Z}}^n$ such that

$$\frac{1}{n}\log|\mathcal{M}_1|\le R_s+\delta \quad (7)$$

$$\max_{y\in\mathcal{Y}}E\{d_s(X^n,g_1(f_1(X^n),y),y)\}\le D_s+\delta. \quad (8)$$

Remarks:

1) The strong achievability may be a desirable feature, as it imposes high search quality for all query objects, as opposed to high search quality averaged over the query alphabet $\mathcal{Y}$.
2) The region of strong achievability coincides exactly with that of a corresponding robust descriptions problem [8], where $g_1(\cdot,y)$ are $|\mathcal{Y}|$ distinct decoders, each equipped with distortion measures $d_s(\cdot,\cdot,y)$.
3) The strongly achievable region of $\{R_s,D_s\}$ is a subset of the achievable region described in Definition 1. Note further that the two regions coincide when $D_s=0$.

We next consider the scheme of scalable coding where descriptions of compressed feature vectors are embedded in the descriptions of compressed data. The data reproduction quality is evaluated by a "reconstruction" distortion measure $d_r:\mathcal{X}\times\hat{\mathcal{X}}\longrightarrow[0,\infty)$ which is generalized to blocks of length n as

$$d_r(x^n,\hat{x}^n)=\frac{1}{n}\sum_{t=1}^{n}d_r(x_t,\hat{x}_t).$$

The enhancement layer encoders and decoders are denoted by

$$f_2:\mathcal{X}^n\longrightarrow\mathcal{M}_2$$

and

$$g_2:\mathcal{M}_1\times\mathcal{M}_2\longrightarrow\hat{\mathcal{X}}^n$$

respectively. Note that the decoder g_2 does not rely on the knowledge of the query, as i) both the reconstruction distortion measure and the data objects are independent of the query, and therefore it cannot be utilized for *better* reconstruction, and ii) the data may not always be accessed via searching, i.e., the user might know the exact location of the specific data object to be reconstructed. The total storage complexity of the system and the reconstruction quality are respectively given by $\frac{1}{n}\log|\mathcal{M}_1||\mathcal{M}_2|$ and $E\{d_r(X^n, g_2(f_1(X^n), f_2(X^n)))\}$.

Definition 3: A quadruple $\{R_s, R_r, D_s, D_r\}$ is *successively achievable* if for all $\delta > 0$, there exist encoding functions $f_1 : \mathcal{X}^n \longrightarrow \mathcal{M}_1$, $f_2 : \mathcal{X}^n \longrightarrow \mathcal{M}_2$, and decoding functions $g_1 : \mathcal{M}_1 \times \mathcal{Y} \longrightarrow \hat{\mathcal{Z}}^n$, $g_2 : \mathcal{M}_1 \times \mathcal{M}_2 \longrightarrow \hat{\mathcal{X}}^n$, such that[5]

$$\frac{1}{n}\log|\mathcal{M}_1| \le R_s + \delta \tag{9}$$

$$\frac{1}{n}\log|\mathcal{M}_1||\mathcal{M}_2| \le R_r + \delta \tag{10}$$

$$E\{d_s(X^n, g_1(f_1(X^n), Y), Y)\} \le D_s + \delta \tag{11}$$

$$E\{d_r(X^n, g_2(f_1(X^n), f_2(X^n)))\} \le D_r + \delta. \tag{12}$$

The notion of strong successive achievability can be similarly defined as in Definition 2.

In the achievability proofs we provide in Section III, we use Csizsár and Körner's notation of types and strong typicality [5]. A vector $\boldsymbol{a} \in \mathcal{A}^n$ is said to be strongly δ-typical with respect to (w.r.t.) random variable $A \sim P_A$ if

$$\left|\frac{1}{n}N(a|\boldsymbol{a}) - P_A(a)\right| < \delta, \qquad \forall\, a \in \mathcal{A}$$

where $N(a|\boldsymbol{a})$ denotes the number of occurrences of symbol a in $\boldsymbol{a}$. The strongly δ-typical set of A, denoted by $T^n_{[A]}$, is the set of all $\boldsymbol{a}$ that are strongly δ-typical w.r.t. A. Similarly, given $\boldsymbol{a} \in \mathcal{A}^n$, the vector $\boldsymbol{b} \in \mathcal{B}^n$ is said to be conditionally strongly δ-typical w.r.t. $P_{B|A}$ if

$$\left|\frac{1}{n}N(a,b|\boldsymbol{a},\boldsymbol{b}) - \frac{1}{n}N(a|\boldsymbol{a})P_B(b|a)\right| < \delta, \quad \forall\, a \in \mathcal{A},\ \ b \in \mathcal{B}$$

and the corresponding typical set is denoted by $T^n_{[B|A]}(\boldsymbol{a})$. We refer the reader to [5] for a detailed discussion of types and typical sets.

III. Main Results

In this section, we first derive the rate–distortion region for the first layer, i.e., the region of all achievable pairs of query processing time R_s and search accuracy D_s. Next, we derive the region of all achievable quadruples $\{R_s, R_r, D_s, D_r\}$. Finally, we analyze the conditions for the achievability of $\{R_s(D_s), R_r(D_r), D_s, D_r\}$, where $R_s(D_s)$ denotes the minimum rate needed to achieve search accuracy D_s, and $R_r(D_r)$ denotes the ordinary rate–distortion function at distortion D_r.

[5]Notice that according to this definition, $\{R_s, R_r, D_s, D_r\}$ can be achievable even when $R_s > R_r$. However, achievability of R_s and R_r implies only that the expended rates at the first and second layers are *at most* R_s and R_r, respectively. Thus, if $R_s > R_r$ is said to be achievable, it is to be understood as "there exists a scheme expending at most R_r bits at both layers."

A. Achievable First-Layer Rate–Distortion Region

We show in the next theorem that the region of all achievable $\{R_s, D_s\}$ is in fact given by the Wyner–Ziv characterization [11], [18].

Theorem 1: A pair (R_s, D_s) is achievable if and only if there exists a random variable U distributed on some alphabet $\mathcal{U}$, and a deterministic function $\phi : \mathcal{U} \times \mathcal{Y} \longrightarrow \hat{\mathcal{Z}}$, such that

$$P_{X,Y,U}(x,y,u) = P_X(x)P_Y(y)P_{U|X}(u|x)$$

and

$$I(X;U) \le R_s \tag{13}$$

$$E\{d_s(X, \phi(U,Y), Y)\} \le D_s. \tag{14}$$

Remark: The corresponding rate–distortion function $R_s(D_s)$ is given by

$$R_s(D_s) = \min_{\substack{P_{U|X}(u|x),\phi(u,y):\\ E\{d_s(X,\phi(U,Y),Y)\}\le D_s}} I(X;U). \tag{15}$$

Note that this is precisely the Wyner–Ziv characterization, since $I(X;U) = I(X;U|Y)$ when $Y - X - U$ form a Markov chain, and Y is independent of X. Therefore, the discussion in [18, Appendix A1] implies that $R_s(D_s)$ is convex and that it suffices to consider alphabets $\mathcal{U}$ of size $|\mathcal{X}| + 1$ to compute $R_s(D_s)$.

Proof of Theorem 1: We begin with the converse. Following the notation of Fig. 3(b), we let $J_1 = f(X^n)$ and $\hat{Z}^n = g_1(J_1, Y)$. If (R_s, D_s) is an achievable pair, then we know that for any $\delta > 0$, there exists large enough n such that

$$D_s + \delta \ge \frac{1}{n}\sum_{t=1}^{n} E\{d_s(X_t, \hat{Z}_t, Y)\} \tag{16}$$

and

$$\begin{aligned} n(R_s + \delta) &\ge H(J_1) \\ &\ge I(X^n; J_1) \\ &= H(X^n) - H(X^n|J_1) \\ &= \sum_{t=1}^{n} H(X_t) - H(X_t|X_1, \ldots, X_{t-1}, J_1) \\ &\ge \sum_{t=1}^{n} I(X_t; J_1). \end{aligned} \tag{17}$$

Since J_1 is independent of Y, we have

$$P_{X_t,Y,J_1}(x_t, y, j_1) = P_{X_t}(x_t)P_Y(y)P_{J_1|X_t}(j_1|x_t).$$

Also, $\hat{Z}_t = \phi_t(J_1, Y)$, where ϕ_t yields the tth component of $g_1(\cdot,\cdot)$. It then follows from the definition of $R_s(D_s)$ that

$$I(X_t; J_1) \ge R_s(E\{d_s(X_t, \hat{Z}_t, Y)\})$$

for all $t = 1, \ldots, n$ (J_1 playing the role of the auxiliary random variable U for all t). Using (16), (17), and convexity of $R_s(D_s)$, we obtain

$$\begin{aligned} R_s + \delta &\ge \frac{1}{n}\sum_{t=1}^{n} I(X_t; J_1) \\ &\ge \frac{1}{n}\sum_{t=1}^{n} R_s(E\{d_s(X_t, \hat{Z}_t, Y)\}) \end{aligned}$$

$$\geq R_s\left(\frac{1}{n}\sum_{t=1}^{n} E\{d_s(X_t,\hat{Z}_t,Y)\}\right)$$
$$\geq R_s(D_s+\delta)$$

for any $\delta>0$. From continuity of $R_s(D_s)$, we conclude $R_s \geq R_s(D_s)$.

For the achievability part, since Y is independent of X, we do not need to use the random binning argument introduced in [18]. Assume that there exist a random variable U and a deterministic function $\phi:\mathcal{U}\times\mathcal{Y}\to\hat{\mathcal{Z}}$, such that $P_{X,Y,U}(x,y,u)=P_X(x)P_Y(y)P_{U|X}(u|x)$, and $E\{d_s(X,\phi(U,Y),Y)\}\leq D_s$.

Code design: Choose M vectors $\{\boldsymbol{u}(k)\}_{k=1}^{M}$ independently and according to a uniform distribution over $T^n_{[U]}$. Reveal this set to the encoder and the decoder.

Encoding: For the source vector $\boldsymbol{x}\in\mathcal{X}^n$, find the lowest index k such that $\boldsymbol{u}(k)\in T^n_{[U|X]}(\boldsymbol{x})$, and send $j_1=f_1(\boldsymbol{x})=k$. If no such $\boldsymbol{u}$ is found, then let $j_1=1$. The rate needed for transmission of j_1 is obviously $\frac{1}{n}\log M$.

Decoding: Reconstruct $\boldsymbol{u}=\boldsymbol{u}(j_1)$ and evaluate $\hat{\boldsymbol{z}}=\Phi(\boldsymbol{u},y)=\{\phi(u_t,y)\}_{t=1}^{n}$. Note that this corresponds to the decoding operation $\hat{\boldsymbol{z}}=g_1(j_1,y)=g_1(f_1(\boldsymbol{x}),y)$.

Expected distortion: Letting $M=2^{n[I(X;U)+\delta]}$, we guarantee that the probability of the existence of $\boldsymbol{u}$ satisfying $\boldsymbol{u}\in T^n_{[U|X]}(\boldsymbol{x})$ approaches 1 as $n\to\infty$. This implies $(\boldsymbol{x},\boldsymbol{u})\in T^n_{[X,U]}$ with probability approaching 1, and therefore,

$$\begin{aligned}
d_s(\boldsymbol{x},\Phi(\boldsymbol{u},y),y)&=\frac{1}{n}\sum_{t=1}^{n} d_s(x_t,\phi(u_t,y),y)\\
&=\sum_{a\in\mathcal{X}}\sum_{b\in\mathcal{U}}\frac{1}{n}N(a,b|\boldsymbol{x},\boldsymbol{u})d_s(a,\phi(b,y),y)\\
&\leq\sum_{a\in\mathcal{X}}\sum_{b\in\mathcal{U}}[P_{X,U}(a,b)+\delta]d_s(a,\phi(b,y),y)\\
&\leq E\{d_s(X,\phi(U,Y),Y)|Y=y\}\\
&\quad+\delta|\mathcal{X}||\mathcal{U}|d_{1,\max}
\end{aligned}$$

where $d_{1,\max}=\max_{x,y,\hat{z}} d_s(x,\hat{z},y)$. Therefore, for large n we have

$$\begin{aligned}
&E\{d_s(X^n,g_1(f_1(X^n),Y),Y)|Y=y\}\\
&\quad\leq\Pr\{(X^n,U^n)\notin T^n_{[X,U]}\}d_{1,\max}\\
&\qquad+E\{d_s(X,\phi(U,Y),Y)|Y=y\}+\delta|\mathcal{X}||\mathcal{U}|d_{1,\max}\\
&\quad\leq E\{d_s(X,\phi(U,Y),Y)|Y=y\}+\delta c_1 \qquad (18)
\end{aligned}$$

where c_1 is a constant. Taking expectations w.r.t. $P_Y(y)$ on both sides establishes the desired achievability result. Note that the expectation on the left-hand side of (18) is w.r.t. the *random* codebook of messages U^n, as well as X^n. Hence, there must exist a deterministic codebook with the desired distortion level. □

The region of strongly achievable $\{R_s,D_s\}$ can similarly be characterized by a slight modification of the proof of Theorem 1. We present the result as a lemma, and omit the proof.

Lemma 1: A pair (R_s,D_s) is strongly achievable if and only if there exist a random variable U distributed on some alphabet $\mathcal{U}$, and a deterministic function $\phi:\mathcal{U}\times\mathcal{Y}\longrightarrow\hat{\mathcal{Z}}$, such that

$$P_{X,Y,U}(x,y,u)=P_X(x)P_Y(y)P_{U|X}(u|x)$$

and

$$I(X;U)\leq R_s \qquad (19)$$
$$\max_{y\in\mathcal{Y}} E\{d_s(X,\phi(U,y),y)\}\leq D_s. \qquad (20)$$

B. The Successive Achievability Region

The next theorem gives a single-letter characterization for the achievable quadruples $\{R_s,R_r,D_s,D_r\}$.

Theorem 2: A quadruple $\{R_s,R_r,D_s,D_r\}$ is successively achievable if and only if there exist random variables $U\in\mathcal{U}$ and $\hat{X}\in\hat{\mathcal{X}}$, and a deterministic function $\phi:\mathcal{U}\times\mathcal{Y}\longrightarrow\hat{\mathcal{Z}}$, such that

$$P_{X,Y,U,\hat{X}}(x,y,u,\hat{x})=P_X(x)P_Y(y)P_{U,\hat{X}|X}(u,\hat{x}|x)$$

and

$$I(X;U)\leq R_s \qquad (21)$$
$$I(X;U,\hat{X})\leq R_r \qquad (22)$$
$$E\{d_s(X,\phi(U,Y),Y)\}\leq D_s \qquad (23)$$
$$E\{d_r(X,\hat{X})\}\leq D_r. \qquad (24)$$

Remarks:

1) For a successively achievable $\{R_s,R_r,D_s,D_r\}$, inequalities (22) and (24) imply $R_r\geq R_r(D_r)$ where $R_r(D_r)$ denotes the ordinary rate–distortion function [2], evaluated at D_r

$$R_r(D_r)=\min_{E\{d_r(X,\hat{X})\}\leq D_r} I(X;\hat{X}). \qquad (25)$$

This is shown by contradiction. Suppose that $R_r<R_r(D_r)$. Then by (22)

$$R_r(D_r)>I(X;U,\hat{X})\geq I(X;\hat{X})$$

which contradicts (24) and (25). Similarly, (21) and (23) imply $R_s\geq R_s(D_s)$.

This result formalizes the intuitively obvious notion that by adopting a two-stage coding scheme, we may be penalized with increased time complexity, or increased storage complexity, or both. In the corollary to the theorem, we will provide conditions for achieving $R_s=R_s(D_s)$ and $R_r=R(D_r)$ simultaneously.

2) Using arguments similar to [18, Theorem A2], it is easy to prove that the successive achievability region defined by (21)–(24) is convex, and that it suffices to consider alphabets $\mathcal{U}$ with size $|\mathcal{X}||\hat{\mathcal{X}}|+3$.

3) If we replace condition (23) with

$$\max_{y\in\mathcal{Y}} E\{d_s(X,\phi(U,y),y)\}\leq D_s$$

we obtain the characterization of the region of strongly and successively achievable $\{R_s,R_r,D_s,D_r\}$. The proof is implied by the proof of the theorem.

Proof of Theorem 2: We begin with the converse. Once again following the notation of Fig. 3(b), we let $J_1=f_1(X^n)$,

$J_2 = f_2(X^n)$, $\hat{Z}^n = g_1(J_1, Y)$, and $\hat{X}^n = g_2(J_1, J_2)$. If (R_s, R_r, D_s, D_r) is an achievable quadruple, then for any $\delta > 0$, there exists large enough n such that

$$D_s + \delta \geq \frac{1}{n}\sum_{t=1}^{n} E\{d_s(X_t, \hat{Z}_t, Y)\} \tag{26}$$

$$D_r + \delta \geq \frac{1}{n}\sum_{t=1}^{n} E\{d_r(X_t, \hat{X}_t)\}. \tag{27}$$

Also, from (17)

$$R_s + \delta \geq \frac{1}{n}\sum_{t=1}^{n} I(X_t; J_1). \tag{28}$$

Finally,

$$\begin{aligned} n(R_r + \delta) &\geq H(J_1, J_2) \\ &\geq I(X^n; J_1, J_2) \\ &\geq I(X^n; J_1, \hat{X}^n) = H(X^n) - H(X^n|J_1, \hat{X}^n) \\ &= \sum_{t=1}^{n} H(X_t) - H(X_t|J_1, X_1, \ldots, X_{t-1}, \hat{X}^n) \\ &\geq \sum_{t=1}^{n} I(X_t; J_1, \hat{X}_t). \end{aligned} \tag{29}$$

Note that

$$P_{X_t,Y,J_1,\hat{X}_t}(x_t, y, j_1, \hat{x}_t) = P_{X_t}(x_t)P_Y(y)P_{J_1,\hat{X}_t|X_t}(j_1, \hat{x}_t|x_t)$$

and that $\hat{Z}_t = \phi_t(J_1, Y)$. The converse, therefore, follows from the convexity of the region defined by (21)–(24), which can be easily proved, and from (26)–(29), by letting J_1 play the role of U in (21)–(24) and taking similar steps as in the converse part of the proof of Theorem 1.

For the achievability part, assume that there exist random variables U, $\hat{X}$, and a deterministic function $\phi : \mathcal{U} \times \mathcal{Y} \to \hat{\mathcal{X}}$, such that

$$P_{X,Y,U,\hat{X}}(x, y, u, \hat{x}) = P_X(x)P_Y(y)P_{U,\hat{X}|X}(u, \hat{x}|x)$$

and

$$\begin{aligned} E\{d_s(X, \phi(U, Y), Y)\} &\leq D_s \\ E\{d_r(X, \hat{X})\} &\leq D_r. \end{aligned}$$

Code design: Choose M_1 vectors $\{\boldsymbol{u}(k)\}_{k=1}^{M_1}$ independently and according to a uniform distribution over $T^n_{[U]}$. For each $k \in \{1, \ldots, M_1\}$, choose M_2 vectors $\{\hat{\boldsymbol{x}}(l|k)\}_{l=1}^{M_2}$ independently and uniformly from $T^n_{[\hat{X}|U]}(\boldsymbol{u}(k))$. Reveal this tree structured codebook to the encoder and the decoder.

Encoding: For the source vector $\boldsymbol{x}$, find the lowest index k such that $\boldsymbol{u}(k) \in T^n_{[U|X]}(\boldsymbol{x})$, and send $j_1 = f_1(\boldsymbol{x}) = k$ as the first-layer message. If no such $\boldsymbol{u}$ is found, then let $j_1 = 1$. If in the first-layer encoding a $\boldsymbol{u}(k) \in T^n_{[U|X]}(\boldsymbol{x})$ was found, find the lowest index l such that $\hat{\boldsymbol{x}}(l|k) \in T^n_{[\hat{X}|U,X]}(\boldsymbol{u}(k), \boldsymbol{x})$, and send $j_2 = f_2(\boldsymbol{x}) = l$ as the second layer message. If no such $\hat{\boldsymbol{x}}$ is found, or if in the first layer no $\boldsymbol{u}$ was found, then let $j_2 = 1$. The resultant first- and second-layer rates are $\frac{1}{n}\log M_1$ and $\frac{1}{n}\log M_1 M_2$, respectively.

Decoding: First-layer decoder reconstructs $\boldsymbol{u} = \boldsymbol{u}(j_1)$ and evaluates

$$\hat{\boldsymbol{z}} = g_1(j_1, y) = \Phi(\boldsymbol{u}, y) = \{\phi(u_t, y)\}_{t=1}^{n}$$

as the first-layer output. The second-layer decoder reconstructs

$$\hat{\boldsymbol{x}} = g_2(j_1, j_2) = \hat{\boldsymbol{x}}(j_2|j_1)$$

as the second-layer output.

Expected distortion: Letting $M_1 = 2^{n[I(X;U)+\delta]}$ and $M_2 = 2^{n[I(X;\hat{X}|U)+\delta]}$, we guarantee that the probability that $(\boldsymbol{x}, \boldsymbol{u}, \hat{\boldsymbol{x}}) \in T^n_{[X,U,\hat{X}]}$ approaches 1 as $n \to \infty$. Having $(\boldsymbol{x}, \boldsymbol{u}, \hat{\boldsymbol{x}}) \in T^n_{[X,U,\hat{X}]}$ implies

$$d_s(\boldsymbol{x}, \Phi(\boldsymbol{u}, y), y) \leq E\{d_s(X, \phi(U, Y), Y)|Y = y\} + \delta c_1$$

as shown in the proof of Theorem 1, and

$$\begin{aligned} d_r(\boldsymbol{x}, \hat{\boldsymbol{x}}) &= \frac{1}{n}\sum_{t=1}^{n} d_r(x_t, \hat{x}_t) \\ &= \sum_{a\in\mathcal{X}}\sum_{b\in\hat{\mathcal{X}}} \frac{1}{n}N(a, b|\boldsymbol{x}, \hat{\boldsymbol{x}})d_r(a, b) \\ &\leq \sum_{a\in\mathcal{X}}\sum_{b\in\hat{\mathcal{X}}} [P_{X,\hat{X}}(a, b) + \delta]d_r(a, b) \\ &\leq E\{d_r(X, \hat{X})\} + \delta|\mathcal{X}||\hat{\mathcal{X}}|d_{2,\max} \end{aligned}$$

where $d_{2,\max} = \max_{x,\hat{x}} d_r(x, \hat{x})$. Therefore, for large n, it follows that

$$\begin{aligned} &E\{d_s(X^n, \hat{Z}^n, Y)|Y = y\} \\ &\qquad \leq E\{d_s(X, \phi(U, Y), Y)|Y = y\} + \delta c_1 \end{aligned} \tag{30}$$

and

$$E\{d_r(X^n, \hat{X}^n)\} \leq D_r + \delta c_2.$$

Taking expectations of (30) w.r.t. $P_Y(y)$ establishes the desired result. □

Corollary 1 (Successive Refinability Without Rate Loss): The quadruple $\{R_s(D_s), R_r(D_r), D_s, D_r\}$ is successively achievable if and only if there exist random variables $U \in \mathcal{U}$ and $\hat{X} \in \hat{\mathcal{X}}$, and a deterministic function $\phi : \mathcal{U} \times \mathcal{Y} \longrightarrow \hat{\mathcal{Z}}$, such that

$$P_{X,Y,U,\hat{X}}(x, y, u, \hat{x}) = P_X(x)P_Y(y)P_{\hat{X}|X}(\hat{x}|x)P_{U|\hat{X}}(u|\hat{x})$$

and

$$I(X; U) = R_s(D_s) \tag{31}$$

$$I(X; \hat{X}) = R_r(D_r) \tag{32}$$

$$E\{d_s(X, \phi(U, Y), Y)\} \leq D_s \tag{33}$$

$$E\{d_r(X, \hat{X})\} \leq D_r. \tag{34}$$

Remark: Note that this is the familiar Markov chain condition that appeared in [6]. That is, the optimal solutions $P^*_{U|X}(u|x)$ and $P^*_{\hat{X}|X}(\hat{x}|x)$ achieving $R_s(D_s)$ and $R_r(D_r)$, respectively, are "compatible" in the sense that there exists a distribution $P_{U|\hat{X}}(u|\hat{x})$ satisfying

$$P^*_{U|X}(u|x) = \sum_{\hat{x}\in\hat{\mathcal{X}}} P^*_{\hat{X}|X}(\hat{x}|x)P_{U|\hat{X}}(u|\hat{x})$$

i.e., $X - \hat{X} - U$ forms a Markov chain. An insightful interpretation of this condition, which is also valid in our scenario, was provided in [12, Sec. III and Fig. 1]. More explicitly, for $\hat{X}$

and U achieving $R_s(D_s)$ and $R_r(D_r)$, if $X - \hat{X} - U$ forms a Markov chain, then optimal "search" balls

$$\left\{ \boldsymbol{x} : \sum_{y \in \mathcal{Y}} P_Y(y) d_s(\boldsymbol{x}, \Phi(\boldsymbol{u}, y), y) \leq D_s \right\}$$

about $\boldsymbol{u} \in T^n_{[U]}$ are *almost* unions of optimal "reconstruction" balls

$$\{ \boldsymbol{x} : d_r(\boldsymbol{x}, \hat{\boldsymbol{x}}) \leq D_r \}$$

about $\hat{\boldsymbol{x}} \in T^n_{[\hat{X}]}$. Therefore, there exists a tree-structured partition of the space such that the first- and second-level regions almost exactly give the optimal search and reconstruction balls, respectively.

Proof of Corollary 1: For the "if" part, having $X - \hat{X} - U$ implies

$$\begin{aligned} R_r(D_r) &= I(X; \hat{X}) \\ &= I(X; \hat{X}) + I(X; U|\hat{X}) = I(X; U, \hat{X}). \end{aligned}$$

Therefore, Theorem 2 implies successive achievability of $\{R_s(D_s), R_r(D_r), D_s, D_r\}$.

For the "only if" part, assume the successive achievability of $\{R_s(D_s), R_r(D_r), D_s, D_r\}$. Then, according to Theorem 2, there must exist a

$$P_{X,Y,U,\hat{X}}(x, y, u, \hat{x}) = P_X(x) P_Y(y) P_{U,\hat{X}|X}(u, \hat{x}|x)$$

and a deterministic function $\phi(u, y)$ such that (21)–(24) holds for $R_s = R_s(D_s)$ and $R_r = R_r(D_r)$. From the very definition of $R_s(D_s)$, (23) implies that (21) holds with equality. Moreover, since

$$I(X; U, \hat{X}) = I(X; \hat{X}) + I(X; U|\hat{X}) \leq R_r(D_r)$$

(24) implies

$$\begin{aligned} I(X; \hat{X}) &= R_r(D_r) \\ I(X; U|\hat{X}) &= 0. \end{aligned}$$

Since the last equality implies that $X - \hat{X} - U$ forms a Markov chain, we have

$$P_{U,\hat{X}|X}(u, \hat{x}|x) = P_{\hat{X}|X}(\hat{x}|x) P_{U|\hat{X}}(u|\hat{x})$$

which completes the proof. □

IV. Computation of the Achievability Region

The successive achievability region in its full generality is difficult to compute, because of i) the abundance of possible functions $\phi(u, y)$ one can choose, and ii) the fact that $E\{d_s(X, \phi(U, Y), Y)\}$ is not differentiable w.r.t. ϕ. However, as we will soon show, the special case of $D_s = D_r = 0$ is easy to compute, as the characterization of successively achievable $\{R_s, R_r, 0, 0\}$ reduces to that of the classical scalable coding scenario derived in [12].

In ordinary scalable rate–distortion analysis, this specialization is perhaps the least interesting, because the distortion measures usually dictate $d(x, \hat{x}) = 0$ if and only if $x = \hat{x}$, and therefore this zero-distortion case is already covered by the analysis of *lossless* coding. Moreover, even for more general $d(x, \hat{x})$, since the distortion is usually measured by the same function in both layers, i.e., $d_s = d_r = d$, there is no need for *refinement* to achieve $D_s = D_r$.

In our scenario, however, fixing $D_s = D_r = 0$ is not necessarily trivial. Recall that the distortion measures quantify different criteria at the first and the second layers, i.e., the inaccuracy of the search, and the inaccuracy of the reconstruction, respectively. Thus, $D_s = D_r$ or even $D_s < D_r$ are in general nontrivial cases. Moreover, for all finite n

$$E\{d_s(X^n, \hat{Z}^n, Y)\} = 0 \tag{35}$$

$$E\{d_r(X^n, \hat{X}^n)\} = 0 \tag{36}$$

imply that

$$\begin{aligned} d_s(x_t, \hat{z}_t, y) &= 0 \\ d_r(x_t, \hat{x}_t) &= 0 \end{aligned}$$

for $t = 1, \ldots, n$, and for all $y \in \mathcal{Y}$, $x_t \in \mathcal{X}$. In other words, (35) and (36) ensure that for *each* object in the database, and for *any* point on the query space, the search distortion and the reconstruction distortion are always "acceptable," i.e., within the allowable limits. Those limits are, of course, determined by the triplets $(x, \hat{z}, y)$ for which $d_s(x, \hat{z}, y) = 0$, and pairs $(x, \hat{x})$ for which $d_r(x, \hat{x}) = 0$. Although achievability of $\{R_s, R_r, 0, 0\}$ does not imply (35) and (36), it certainly implies for arbitrarily small $\delta > 0$ the existence of large enough n such that

$$E\{d_s(X^n, \hat{Z}^n, Y)\} \leq \delta \tag{37}$$

$$E\{d_r(X^n, \hat{X}^n)\} \leq \delta. \tag{38}$$

This in turn means that an arbitrarily large subset of data objects are searched and reconstructed with acceptable accuracy, which is a very desirable feature.

A. An Alternative Characterization for $D_s = D_r = 0$

Substituting $D_s = D_r = 0$ in (23) and (24) of Theorem 2 yields

$$E\{d_s(X, \phi(U, Y), Y)\} = 0 \tag{39}$$

$$E\{d_r(X, \hat{X})\} = 0. \tag{40}$$

Expanding (39), we obtain

$$\sum_{x \in \mathcal{X}} P_X(x) \sum_{y \in \mathcal{Y}} P_Y(y) \sum_{u \in \mathcal{U}} P_{U|X}(u|x) d_s(x, \phi(u, y), y) = 0. \tag{41}$$

Assuming without loss of generality that $P_X(x) > 0$ and $P_Y(y) > 0$ for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, (41) holds if and only if for every $x \in \mathcal{X}$ and $u \in \mathcal{U}$,

$$P_{U|X}(u|x) > 0 \Longrightarrow d_s(x, \phi(u, y), y) = 0, \qquad \forall\, y \in \mathcal{Y}.$$

Using this observation, we prove the following simplified characterization of all successively achievable $\{R_s, R_r, 0, 0\}$.

Theorem 3: $\{R_s, R_r, 0, 0\}$ is successively achievable if and only if there exist random variables $W \in \mathcal{W}$ and $\hat{X} \in \hat{\mathcal{X}}$, jointly distributed with X, such that

$$I(X; W) \leq R_s \tag{42}$$

$$I(X; W, \hat{X}) \leq R_r \tag{43}$$

$$E\{d_1'(X, W)\} = 0 \tag{44}$$

$$E\{d_r(X, \hat{X})\} = 0 \tag{45}$$

where

$$d_1'(x,w)=\begin{cases}0, & x\in w\\ 1, & x\notin w\end{cases} \tag{46}$$

and

$$\begin{aligned}\mathcal{V}\triangleq\{&v\subseteq\mathcal{X}:\exists\psi(v,y)\text{ s.t. }d_s(x,\psi(v,y),y)=0\\ &\forall\, x\in v,\ \forall\, y\in\mathcal{Y}\}\\ \mathcal{W}\triangleq\{&w\in\mathcal{V}:\nexists v\supset w\text{ s.t. }v\in\mathcal{V}\}.\end{aligned}$$

Remarks:

1) The new alphabet $\mathcal{V}$ consists of subsets of the data space satisfying the following: For any query y, it is possible to reconstruct a single feature vector $\hat{z}=\psi(v,y)$ yielding acceptable search quality for the whole collection v of data objects. The alphabet $\mathcal{W}$, on the other hand, consists only of those sets in $\mathcal{V}$ which are maximal, i.e., which are not properly contained in any other set in $\mathcal{V}$. This result indicates that the problem of encoding with acceptable search quality for all data objects and queries reduces to a regular "covering" problem, where optimal encoding of blocks of objects drawn from $\mathcal{X}^n$ is performed by covering the high probability set $T_{[X]}^n$ using Cartesian products of the sets in $\mathcal{W}$.

2) Although the theorem would remain valid by setting $\mathcal{W}=\mathcal{V}$, as we will see via examples in Section IV-C, the size of the set $\mathcal{V}$ could be prohibitively large for computation purposes. Using $\mathcal{W}$ instead of $\mathcal{V}$ significantly reduces the complexity of the task of computing the successive achievability region.

Proof of Theorem 3: The sufficiency part follows by setting $U=W$ (and, therefore, $\mathcal{U}=\mathcal{W}$), and $\phi(u,y)=\psi(u,y)$, because (44) and (46) imply that

$$P_{U|X}(u|x)>0\Longrightarrow x\in u.$$

Hence, by the definition of $\mathcal{V}$, and the fact that $U=W\in\mathcal{W}\subseteq\mathcal{V}$, we obtain for every $x\in\mathcal{X}$ and $u\in\mathcal{U}$ that

$$P_{U|X}(u|x)>0\Longrightarrow d_s(x,\phi(u,y),y)=0,\qquad\forall\, y\in\mathcal{Y}$$

which implies that (39) holds. Therefore, $\{R_s,R_r,0,0\}$ is successively achievable, as (21), (22), and (40) also hold.

For the necessity part, assume that (21), (22), (39), and (40) are satisfied. Then, consider subsets of $\mathcal{X}$ defined by the mapping

$$v(u)\triangleq\{x:P_{U|X}(u|x)>0\}. \tag{47}$$

Now, observe that

$$x\in v(u)\Longrightarrow d_s(x,\phi(u,y),y)=0,\qquad\forall\, y\in\mathcal{Y}$$

which follows from (39) and (47). Therefore, by setting $\psi(v,y)=\phi(u,y)$ for some arbitrary u such that $v(u)=v$, we have

$$x\in v\Longrightarrow d_s(x,\psi(v,y),y)=0,\qquad\forall\, y\in\mathcal{Y}$$

and, hence, $v(u)\in\mathcal{V}$. Next, consider an arbitrary mapping $\sigma:\mathcal{V}\to\mathcal{W}$ such that $v\subseteq\sigma(v)$ for all $v\in\mathcal{V}$, and let

$$P_{W,\hat{X}|X}(w,\hat{x}|x)=\sum_{u:\sigma(v(u))=w}P_{U,\hat{X}|X}(u,\hat{x}|x).$$

Now, (21) and (22) automatically imply (42) and (43), respectively, and (45) is the same as (40). Finally, (44) also holds, since

$$\begin{aligned}P_{W|X}(w|x)>0&\Longrightarrow\exists u\text{ s.t. }\sigma(v(u))=w\\ &\qquad\text{and }P_{U|X}(u|x)>0\\ &\Longrightarrow\exists u\text{ s.t. }x\in v(u)\subseteq w\\ &\Longrightarrow x\in w\\ &\Longrightarrow d_1'(x,w)=0.\end{aligned}$$

□

B. Computation of Achievable Rates

The alternative characterization of Theorem 3 is *precisely* the Rimoldi characterization for successive refinability [12], specialized to $D_s=D_r=0$. Therefore, any method devised for the computation of Rimoldi's region can be utilized to compute the region of successively achievable $\{R_s,R_r,0,0\}$ in our scenario.

In [15], we developed an iterative algorithm which is guaranteed to converge to a point on the rate–distortion surface. The limit point on the surface is determined by the positive Lagrangian multipliers for the rate and the distortion terms. To achieve a point with $D_s=D_r=0$, it suffices to let the multipliers for the distortion terms tend to $+\infty$. Assuming without loss of generality that the other two multipliers add up to 1, the algorithm in [15] simplifies to the algorithm provided as follows.

1) Initialize with arbitrary $Q(w,\hat{x})>0$ for all $w\in\mathcal{W}$, $\hat{x}\in\hat{\mathcal{X}}$, such that
$$\sum_{w,\hat{x}}Q(w,\hat{x})=1.$$
Also set the Lagrangian multiplier $0\le\lambda\le 1$ for the first-layer rate.

2) For fixed $Q(w,\hat{x})$, compute $P_{W,\hat{X}|X}(w,\hat{x}|x)$ as
$$P_{W,\hat{X}|X}(w,\hat{x}|x)=\frac{Q(w,\hat{x})\mu(x,w)^{-\lambda}}{\nu(x)} \tag{48}$$
if $d_1'(x,w)=0$ and $d_r(x,\hat{x})=0$, and
$$P_{W,\hat{X}|X}(w,\hat{x}|x)=0$$
otherwise. Here
$$\begin{aligned}\mu(x,w)&=\sum_{\hat{x}:d_r(x,\hat{x})=0}\frac{Q(w,\hat{x})}{\sum_{\hat{x}'}Q(w,\hat{x}')}\\ &=\sum_{\hat{x}:d_r(x,\hat{x})=0}Q(\hat{x}|w)\end{aligned}$$
and
$$\nu(x)=\sum_{w,\hat{x}:d_1'(x,w)=0,d_r(x,\hat{x})=0}Q(w,\hat{x})\mu(x,w)^{-\lambda}.$$

3) For fixed $P_{W,\hat{X}|X}(w,\hat{x}|x)$, compute $Q(w,\hat{x})$ as
$$Q(w,\hat{x})=\sum_x P_X(x)P_{W,\hat{X}|X}(w,\hat{x}|x).$$

4) Iterate steps 2) and 3) until convergence.

5) Compute $R_s=I(X;W)$ and $R_r=I(X;W,\hat{X})$ using the resultant $P_{W,\hat{X}|X}(w,\hat{x}|x)$.

$w_H(x)$	x	$\hat{z}=0$	1	2	3	$\hat{z}=0$	1	2	3	$\hat{z}=0$	1	2	3	$\hat{z}=0$	1	2	3
0	000	0	0	1	1	0	0	0	0	0	1	1	1	0	0	1	1
1	001	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1
1	010	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1
2	011	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0
1	100	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1
2	101	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0
2	110	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0
3	111	1	1	0	0	1	1	1	0	0	0	0	0	1	1	0	0
		$y=0$				$y=1$				$y=2$				$y=3$			

Fig. 5. Tabular demonstration of $d_s(x,\hat{z},y)$ in Example A with $M=3$, $\alpha=1$, $\beta=0$, and $\epsilon=1$.

The complexity considerations of [15] imply that analytical computation of points on the rate–distortion surface is not feasible in general. Therefore, an iterative algorithm such as the above is the only practical tool for the computation of the region of successively achievable $\{R_s, R_r, 0, 0\}$.

C. Examples

We revisit here the examples given in Section II, and compute the region of successively achievable $\{R_s, R_r, 0, 0\}$. Although the examples are intended for large K and/or M, we focus here on very simple cases and analyze the behavior of the achievable rate region.

1) Example A: Let $K=1$, i.e., $\mathcal{X}=\hat{\mathcal{X}}=\{0,1\}^M$, $\mathcal{Z}=\hat{\mathcal{Z}}=\mathcal{Y}=\{0,\ldots,M\}$, and $\rho(y,z)=|y-z|$. Consider the search accuracy measure $d_s(x,\hat{z},y)$ induced by (2), and reconstruction quality measure given by

$$d_r(x,\hat{x})=\begin{cases}1, & d_H(x,\hat{x})>T\\ 0, & \text{otherwise}\end{cases} \tag{49}$$

which thresholds at T the value of the Hamming distance between x and $\hat{x}$, given by

$$d_H(x,\hat{x})=\frac{1}{M}\sum_{i=1}^{M}|x_i-\hat{x}_i|.$$

Assume a uniform distribution over $\mathcal{X}$, i.e., $P_X(x)=\frac{1}{2^M}$ for all $x\in\mathcal{X}$. We consider three simple cases, each of which demonstrates a different phenomenon of the achievability region $\{R_s, R_r\}$.

a) Let $M=3$, $\alpha=1$, $\beta=0$, $\epsilon=1$, and $T=1/3$. With the specified parameters, $d_s(x,\hat{z},y)$ becomes

$$d_s(x,\hat{z},y)=\begin{cases}1, & |y-w_H(x)|=0 \text{ and } |y-\hat{z}|>1\\ 1, & |y-w_H(x)|>1 \text{ and } |y-\hat{z}|\le 1\\ 0, & \text{otherwise}\end{cases} \tag{50}$$

where $w_H(x)$ denotes the Hamming weight of $x\in\{0,1\}^M$. This accuracy measure is displayed as a table in Fig. 5.

We now form the set $\mathcal{V}$, and then the set $\mathcal{W}$, to utilize the alternative characterization in Theorem 3. According to the definition, in order for a subset $v\subset\{0,1\}^M$ to be an element of $\mathcal{V}$, there must exist a $\hat{z}$ for each y such that all entries in the column $(\hat{z},y)$ and the rows $x\in v$ are 0. For example, $\{000,001,010,011\}\notin\mathcal{V}$, because for $y=0$ or $y=2$, there is no choice of $\hat{z}$ such that the first four rows are 0. On the other hand, $\{000,001,010\}\in\mathcal{V}$, since $\hat{z}=0$ satisfies the requirement for each y. By close inspection, we conclude that

$$\mathcal{V}=\{v\subset\mathcal{X}:|w_H(x)-w_H(x')|\le 1\ \forall\, x,x'\in v\}.$$

There are 79 elements in the set $\mathcal{V}$. However, the cardinality of $\mathcal{W}$ is only 3, since it consists of only the maximal elements in $\mathcal{V}$, which are

$$\begin{aligned}&\{x\in\mathcal{X}:w_H(x)=0 \text{ or } w_H(x)=1\}\\ &\quad=\{000,001,010,100\}\\ &\{x\in\mathcal{X}:w_H(x)=1 \text{ or } w_H(x)=2\}\\ &\quad=\{001,010,011,100,101,110\}\\ &\{x\in\mathcal{X}:w_H(x)=2 \text{ or } w_H(x)=3\}\\ &\quad=\{011,101,110,111\}.\end{aligned}$$

Running the algorithm provided in Section III for various $0\le\lambda\le 1$, we obtain the achievability curve shown in Fig. 6(a). The two corner points corresponding to $\lambda=1$ and $\lambda=0$ are given by $\{0.8113, 1.3078\}$ and $\{1,1\}$, respectively. It is evident that we cannot simultaneously achieve the nonscalable bounds $R_s(0)=0.8113$ and $R_r(0)=1$. In other words, this specific setup of $(P_X,M,\alpha,\beta,\epsilon,T)$ is not successively refinable without rate loss.

b) Let $M=3$, $\alpha=1$, $\beta=1$, $\epsilon=1$, and $T=1/3$. Then $d_s(x,\hat{z},y)$ becomes

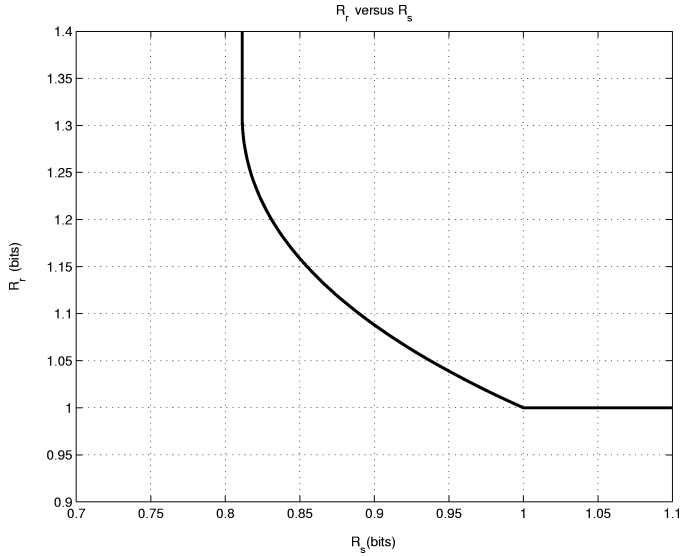
$$d_s(x,\hat{z},y)=\begin{cases}1, & |y-w_H(x)|=0 \text{ and } |y-\hat{z}|>1\\ 1, & |y-w_H(x)|=3 \text{ and } |y-\hat{z}|\le 1\\ 0, & \text{otherwise.}\end{cases}$$

A similar analysis as above shows that $\mathcal{W}$ consists of only two elements

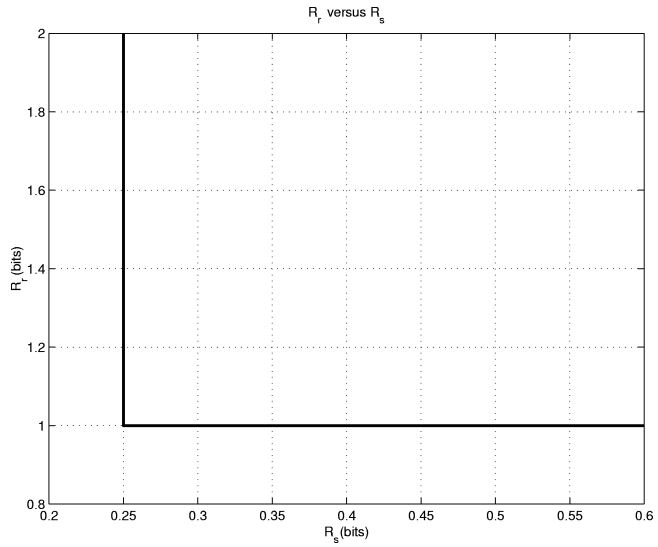
$$\begin{aligned}&\{x\in\mathcal{X}:w_H(x)\neq 3\}\\ &\quad=\{000,001,010,011,100,101,110\}\\ &\{x\in\mathcal{X}:w_H(x)\neq 0\}\\ &\quad=\{001,010,011,100,101,110,111\}.\end{aligned}$$

In this case, without running the iterative algorithm, we can analytically prove that successive refinability is achieved without rate loss, i.e., the nonscalable bounds $R_s(0)=0.25$ and $R_r(0)=1$ are simultaneously achievable. Therefore, the achievability region is as shown in Fig. 6(b).

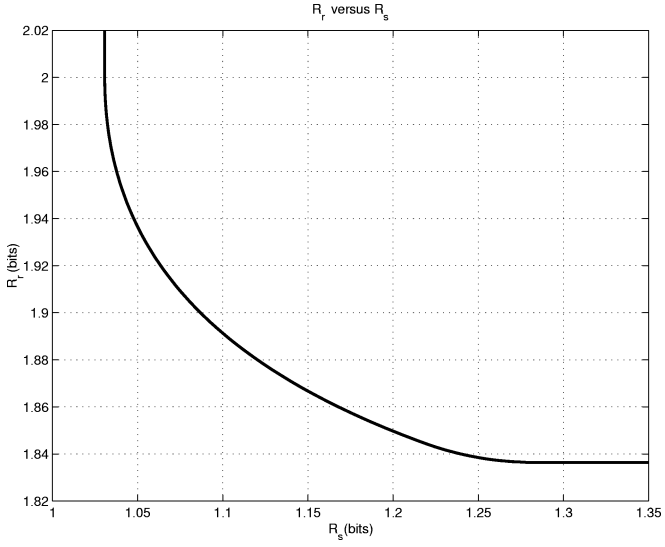
c) Let $M=4$, $\alpha=1$, $\beta=0$, $\epsilon=1$, and $T=1/4$. Therefore, $d_s(x,\hat{z},y)$ is the same as in (50), and it is not



(a)



(b)



(c)

Fig. 6. Regions of successively achievable $\{R_s, R_r\}$ for Example A, cases a), b), and c), respectively.

difficult to show that $\mathcal{W}$ consists of four elements of the form

$$\{x \in \mathcal{X} : i \le w_H(x) \le i+1\}, \qquad i = 0, 1, 2, 3.$$

In Fig. 6(c), we show the achievability region computed by the iterative algorithm for various $0 \le \lambda \le 1$. For this example, we observe a somewhat surprising outcome: $R_r = R_r(0) = 1.6781$ is *not achievable* by any scalable coding scheme. That is, not only is the setup not successively refinable without rate loss, but there is also an inherent rate loss at the second layer, i.e., $R_r \ge 1.8364 > R_r(0)$. In other words, as soon as one decides to facilitate search in the database, one has to pay an extra price in storage. This is perhaps because of the fact that the requirements of this particular type of search (i.e., the measure d_1') are "incompatible" with the requirements of reconstruction quality of the data objects (the measure d_r).

2) Example B: Let $K = 1$ and consider the search accuracy measure $d_s(x, \hat{z}, y)$ induced by (4), and the reconstruction quality measure $d_r(x, \hat{x})$ given in (49). We again consider an example case characterized by the parameters $M = 10$, $\alpha = 2$, and $T = 1/10$, and the distribution $P_X(x) = \frac{1}{2^M}$. We observe that

$$|\rho(y,z) - \rho(y,\hat{z})| = ||y-z| - |y-\hat{z}|| \le |z-\hat{z}|$$

with equality when $y = 0$. Thus, $y = 0$ is the most restrictive query point in determining the set $\mathcal{V}$, because if $d_s(x, \hat{z}, 0) = 0$, then $d_s(x, \hat{z}, y) = 0$ for all $y \in \mathcal{Y}$. In other words, the set $\mathcal{V}$ becomes

$$\mathcal{V} = \{v \subset \mathcal{X} : \exists \psi \text{ s.t. } d_s(x, \psi(v, 0), 0) = 0 \; \forall \, x \in v\}$$

with

$$d_s(x, \hat{z}, 0) = \begin{cases} 1, & |w_H(x) - \hat{z}| > \alpha \\ 0, & \text{otherwise.} \end{cases}$$

It easily follows that the maximal sets in $\mathcal{V}$ are characterized by $\alpha \le \hat{z} \le M - \alpha$, and therefore, $\mathcal{W} = \{w_0, w_1, \ldots, w_6\}$, where

$$w_i = \{x \in \mathcal{X} : i \le w_H(x) \le i + 4\}$$

for $0 \le i \le 6$. Fig. 7 shows the successive achievability region obtained using the iterative algorithm. Again, we observe from Fig. 7 that the chosen parameters, namely, $M = 10$, $\alpha = 2$, and $T = 1$, do not yield successive refinability without rate loss, as nonscalable bounds $R_s(0) = 0.4055$ and $R_r(0) = 6.5406$ are not achieved simultaneously.

V. Summary and Concluding Remarks

This work investigates the relationship between rate–distortion theory and efficient content-based data retrieval from high-dimensional databases. It is motivated by the observation that the optimal performance tradeoff of a similarity search engine is closely related to the fundamental rate–distortion tradeoff.

We showed that the minimum asymptotic rate $R_s(D_s)$ required to achieve search quality D_s is given by the rate–distortion function for a special case of the well-known Wyner–Ziv problem, where the query point is viewed as decoder side information. Since the distortion measure we consider is side-information dependent, knowledge of the query distribution is exploited to reduce the search distortion, although queries are assumed to be independent of database entries.

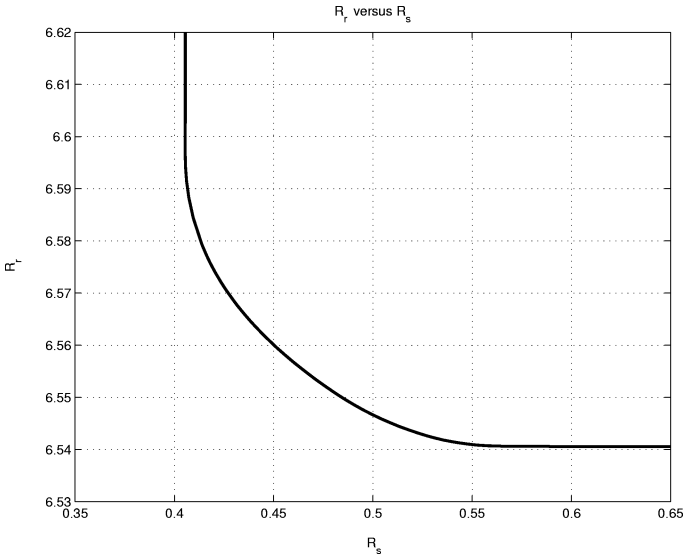


Fig. 7. Successively achievable R_r versus R_s for Example B with parameters $M = 10$, $\alpha = 2$, and $T = 1/10$.

If the database itself has to be in compressed form, the optimal strategy is to implement a scalable coder. For searching purposes, only the first layer is decoded, while reconstruction uses the entire bitstream. We analyzed the tradeoff between the first-layer rate R_s, and the total rate R_r, employed to achieve search distortion D_s and reconstruction distortion D_r. We derived the necessary and sufficient condition for successive refinability without rate loss, i.e., the achievability of $\{R_s(D_s), R_r(D_r), D_s, D_r\}$. As in the classical successive refinement problem, the derived condition involves Markovian properties of the distributions that achieve $R_s(D_s)$ and $R_r(D_r)$.

Finally, we considered examples where d_s and d_r are distortion measures assuming only values 0 and 1, and we require $D_s = D_r = 0$. This special case is both nontrivial and of practical interest, as it enforces with high probability acceptable search and reconstruction qualities for each data point and for the entire query space. We showed that, in this setting, the achievability region corresponds to that of a classical scalable coding problem. The analysis of this special case also shed some light on the design of practical database systems. Although the design of a practical system is out of the scope of this paper, it is worth mentioning here some observations. It is apparent that the optimal encoding regions of the first layer quantizer (determined by elements of $\mathcal{W}$) are *not* classical nearest neighbor regions. Instead, they are regions for which a good feature vector reconstruction (according to the distortion measure $d_s(\cdot, \cdot, \cdot)$) is possible for each query. This makes the design of a quantizer a tedious task, and may practically necessitate recourse to suboptimal solutions. For instance, one way to tackle the design difficulty can be to partition the feature vector space in a nearest neighbor manner first, and then combine regions with similar characteristics.

ACKNOWLEDGMENT

The authors wish to thank the anonymous reviewers and the Associate Editor for their constructive comments. They are also indebted to Hakan Ferhatosmanoglu for introducing them to open problems in database management systems.

REFERENCES

[1] S. Berchtold, D. Keim, and H. Kriegel, "The X-tree: An index structure for high-dimensional data," in *Proc. Int. Conf. Very Large Data Bases*, Bombay, India, 1996, pp. 28–39.

[2] T. Berger, *Rate Distortion Theory. A Mathematical Basis for Data Compression*. Englewood Cliffs, NJ: Prentice-Hall, 1971.

[3] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is "nearest neighbor" meaningful?," in *Proc. Int. Conf. Database Theory*, Jerusalem, Israel, Jan. 1999, pp. 217–225.

[4] K. Chakrabarti and S. Mehrotra, "Local dimensionality reduction: A new approach to indexing high dimensional databases," in *Proc. 26th Int. Conf. Very Large Databases*, Cairo, Egypt, Sept. 2000, pp. 89–100.

[5] I. Csiszár and J. Körner, *Information Theory: Coding Theorems for Discrete Memoryless Systems*. New York: Academic, 1982.

[6] W. H. R. Equitz and T. M. Cover, "Successive refinement of information," *IEEE Trans. Inform. Theory*, vol. 37, pp. 269–275, Mar. 1991.

[7] H. Ferhatosmanoglu, E. Tuncel, D. Agrawal, and A. El Abbadi, "Approximate nearest neighbor searching in multimedia databases," in *Proc. 17th IEEE Int. Conf. Data Engineering*, Heidelberg, Germany, Apr. 2001, pp. 503–511.

[8] A. El Gamal and T. M. Cover, "Achievable rates for multiple descriptions," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 851–857, Nov. 1982.

[9] A. Guttman, "R-trees: A dynamic index structure for spatial searching," in *Proc. ACM SIGMOD Int. Conf. Management of Data*, 1984, pp. 47–57.

[10] K. V. R. Kanth, D. Agrawal, and A. K. Singh, "Dimensionality reduction for similarity searching dynamic databases," in *Proc. ACM SIGMOD Int. Conf. Management of Data*, Seattle, WA, June 1998, pp. 166–176.

[11] T. Linder, R. Zamir, and K. Zeger, "On source coding with side-information-dependent distortion measures," *IEEE Trans. Inform. Theory*, vol. 46, pp. 2697–2704, Nov. 2000.

[12] B. Rimoldi, "Successive refinement of information: Characterization of the achievable rates," *IEEE Trans. Inform. Theory*, vol. 40, pp. 253–259, Jan. 1994.

[13] J. T. Robinson, "The kdb-tree: A search structure for large multi-dimensional dynamic indexes," in *Proc. ACM SIGMOD Int. Conf. Management of Data*, 1981, pp. 10–18.

[14] E. Tuncel, H. Ferhatosmanoglu, and K. Rose, "VQ-index: An index structure for similarity searching in multimedia databases," in *Proc. ACM Multimedia Conf.*, Juan Les Pins, France, Dec. 2002, pp. 543–552.

[15] E. Tuncel and K. Rose, "Computation and analysis of the N-layer scalable rate distortion function," *IEEE Trans. Inform. Theory*, vol. 49, pp. 1218–1230, May 2003.

[16] R. Weber and K. Bohm, "Trading quality for time with nearest-neighbor search," in *Proc. Int. Conf. Extending Database Technology*, Konstanz, Germany, Mar. 2000, pp. 21–35.

[17] R. Weber, H. J. Schek, and S. Blott, "A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces," in *Proc. Int. Conf. Very Large Data Bases*, New York, Aug. 1998, pp. 194–205.

[18] A. D. Wyner and J. Ziv, "The rate distortion function for source coding with side information at the decoder," *IEEE Trans. Inform. Theory*, vol. IT-22, pp. 1–11, Jan. 1976.