An Implementable Lossy Version of the Lempel–Ziv Algorithm—Part I: Optimality for Memoryless Sources

Ioannis Kontoyiannis, *Member, IEEE*

***Abstract*— A new lossy variant of the Fixed-Database Lempel–Ziv coding algorithm for encoding at a fixed distortion level is proposed, and its asymptotic optimality and universality for memoryless sources (with respect to bounded single-letter distortion measures) is demonstrated: As the database size m increases to infinity, the expected compression ratio approaches the rate-distortion function. The complexity and redundancy characteristics of the algorithm are comparable to those of its lossless counterpart. A heuristic argument suggests that the redundancy is of order $(\log\log m)/\log m$, and this is also confirmed experimentally; simulation results are presented that agree well with this rate. Also, the complexity of the algorithm is seen to be comparable to that of the corresponding lossless scheme.**

We show that there is a tradeoff between compression performance and encoding complexity, and we discuss how the relevant parameters can be chosen to balance this tradeoff in practice. We also discuss the performance of the algorithm when applied to sources with memory, and extensions to the cases of unbounded distortion measures and infinite reproduction alphabets.

***Index Terms*—Fixed database, Lempel-Ziv, lossy data compression, universal source coding.**

I. Introduction

OVER the past 25 years, the practical requirement for efficient data compression has become apparent in almost every engineering application where large amounts of data are transmitted or stored.

In applications where the data needs to be perfectly reconstructed from its compressed form (*lossless* coding), the most prominent example of a successful practical scheme is probably the Lempel–Ziv data compression algorithm. Variations of the original scheme [47], [48] are implemented on most personal computers in use today. Although in terms of compression performance they have been shown to be asymptotically optimal and to achieve optimality universally over several general classes of data sources (i.e., without prior knowledge of the source) [47], [45], [48], [34], [28], [35], their practical success is mainly due to the fact that they provide low-complexity algorithms that offer themselves to easy on-line implementations. (A comprehensive introduction to several lossless Lempel–Ziv schemes and their implementations is given in the recent text [16]; see also [3] for numerous variants.)

On the other hand, there are several applications in which the requirement for perfect reconstruction of the data can be relaxed (*lossy* coding), for example, when images are transmitted over the World-Wide Web. In this case, the story has been somewhat less successful. From rate-distortion theory [4], we know that one can achieve a sometimes dramatic improvement in compression by allowing some amount of error in the reconstructed data. In fact, it has been demonstrated that there exist universal algorithms for lossy data compression that asymptotically achieve optimal performance and, moreover, there are explicit constructions of such universal codes; see [17], the references therein, and the more recent work of Zhang, Wei, and Yang [42], [43]. Typically, these constructions either involve exhaustive searches over the space of all possible codebooks or are of exponential complexity at the encoder and therefore cannot be realistically implemented in practice (cf. [46], [27], [25], and [37]). More practical algorithms have been recently proposed by Yang, Zhang, and Berger [39] (partly expanding on the ideas of Muramatsu and Kanaya [25]), where they suggest a new way for circumventing the exponential encoding complexity of earlier schemes.

Motivated by the success of the lossless Lempel–Ziv schemes, several attempts were made to extend them to the case of lossy coding, most notably by Morita and Kobayashi [24] and by Steinberg and Gutman [29]. Although fairly easy to implement, these schemes unfortunately turned out to have strictly suboptimal compression performance; see [23], [38], [12], and the discussion in Section III.

The purpose of this paper is to present and analyze a new universal lossy compression algorithm, generalizing the Fixed-Database Lempel–Ziv (FDLZ) lossless compression algorithm [34] to the lossy case. As we describe in the next section, it is a scheme for encoding memoryless sources at a fixed-distortion level. We show that its compression performance is asymptotically optimal with respect to bounded single-letter distortion measures, and in Section V we argue that its encoding complexity is comparable to that of its lossless counterpart, in that it is only polynomial in the length of the encoded message. We give simple (and rather crude) upper

Manuscript received July 27, 1998; revised June 23, 1999. This work was supported in part by NSF under Grant NCR-9628193, by JSEP under Grant DAAH04-94-G-0058, and by ARPA under Grant J-FBI-94-218-2. The material in this paper was presented in part at the IEEE International Symposium on Information Theory, MIT, Cambridge, MA, August 1998.

The author is with the Department of Statistics, Purdue University, 1399 Mathematical Sciences Building, W. Lafayette, IN 47907-1399 USA.

Communicated by N. Merhav, Associate Editor for Source Coding.

Publisher Item Identifier S 0018-9448(99)08517-X.

bounds on the complexity, and we also discuss some of the algorithm's practical limitations.

The gist of our approach is that, instead of using a database generated by the same distribution as the data, the encoder is allowed to have *multiple databases* simultaneously available, and to adaptively choose which one to use at each step in a "greedy" way. As the database length grows, the number of available databases also grows so that, in effect, codebooks are generated according to all possible reproduction distributions. By controlling the rate at which the number of databases grows, we can make sure that reasonable complexity is maintained at the encoder, while at the same time the set of possible codebook distributions is refined to cover an asymptotically dense set.

Although the notion of using multiple codebooks for source coding is well known in information theory [44], [26], multiple codebook algorithms typically involve either a training stage, or a large search over (essentially) all possible codebooks. For example, Chou, Effros, and Gray's [7] vector-quantization interpretation of universal lossy source coding is in terms of two-pass (or "two-stage") weighted universal codes. Another family of two-pass lossy compression algorithms is that of empirically designed vector quantizers, discussed by Linder, Lugosi, and Zeger [22] among many others. (More pointers to the large literature on vector quantization can be found in the recent review paper by Gray and Neuhoff [15].) Preliminary results from a work closer in spirit to our approach were recently reported by Zamir and Rose in [40] and [41].

We analyze the performance of the new algorithm (presented in Section II-B) by studying the asymptotic behavior of *waiting times* between stationary processes. The connection between waiting times and data compression seems to have been first made by Willems [31]. At about the same time, Wyner and Ziv [33] showed that the asymptotics of waiting (and recurrence) times are intimately connected to the performance of several variants of the Lempel–Ziv scheme, and, since then, a number of papers have appeared exploiting this connection (see, e.g., [34], [35], [29], [30], and [32]). The first step in our analysis, carried out in Section III, is to study the performance of an idealized version of the algorithm in terms of waiting times, whose asymptotic behavior is determined by the strategy that was introduced in [19] and [12], namely, the waiting times are first approximated by a sequence of large-deviation probabilities (Lemma 1), and then large-deviations techniques are used to identify the exponent of decay of these probabilities (Lemma 2). We should note that a related approach was adopted by Bucklew in [5] and [6], where he utilizes large deviations for distortion balls to prove direct coding theorems.

In Section IV, we relate this idealized scheme to the practical algorithm. This is done by realizing that there is a duality relationship between waiting times and match lengths. This relationship is not as straightforward as in the lossless case, and some new subtleties arise in the proofs. Nevertheless, the optimality of the practical algorithm can be deduced from carefully exploiting this duality, in combination with the waiting times results of the previous section.

The reason why this algorithm compresses optimally can be explained intuitively as follows: We know from rate-distortion theory that, unlike in the case of lossless coding, when distortion is allowed the optimal codebook distribution is typically different from the distribution of the source. The most straightforward way to fix this mismatch between a fixed database and the optimum one is to maintain multiple databases at the encoder and decoder, so that a good enough match can always be found. In this way, two objectives are simultaneously achieved.

i) Universality; the same algorithm with the same set of databases works for any memoryless source.

ii) Reasonable complexity; like FDLZ in the lossless case, what makes this algorithm potentially attractive is that it provides a sequence of suboptimal coding schemes, indexed by the database length and the number of available databases, that offer a handle on the complexity/redundancy tradeoff: Using a few short databases, we get efficient, easily implementable algorithms with high redundancy. On the other hand, increasing the length and the number of databases provides algorithms with compression performance that can be made arbitrarily close to being optimal, at the cost of increasing the encoding complexity. Quantitative bounds on the precise form of this tradeoff are given in Section V.

We also note that there is a wealth of approximate string matching algorithms (see [10], [1], [2], [9], and the references therein) allowing for efficient implementations.

The rest of the paper is organized as follows. In the next section we describe the algorithm in detail and present our main theoretical result, Theorem 1, stating its asymptotic optimality. In Section III we first give an informal explanation of this optimality, and we state and prove the theoretical results that are needed in order to establish it formally. This is done in Section IV. In Section V we discuss implementation issues, and present some details on the quantitative nature of the complexity/redundancy tradeoff. A heuristic argument suggests that the redundancy of the algorithm is of the same order of magnitude as that of the lossless FDLZ, and we present simulation results that seem to confirm this rate. In Section VI we describe extensions of the algorithm in several directions: more general classes of sources, unbounded distortion measures, fixed-rate coding. Section VII and Appendices I and II contain the proofs of the theoretical results in Sections III and IV.

II. Description of the Algorithm

After some preliminary definitions, in Section II-B we describe the compression algorithm in its simplest form and we state our first result, Theorem 1, which establishes its asymptotic optimality. The algorithm is a lossy source-coding scheme for encoding memoryless sources at a fixed distortion level, with respect to single-letter distortion measures. Extensions of the use of the algorithm to more general situations are discussed in Section VI.

A. Preliminaries

Let $\boldsymbol{X} = \{X_n; n \geq 1\}$ be a memoryless source with values in the *source alphabet* A, where A is a Polish space (namely,

a complete, separable metric space) equipped with its Borel σ-field $\mathcal{A}$. The distribution of $\boldsymbol{X}$ is determined by specifying that the random variables $\{X_n\}$ are independent and identically distributed (i.i.d.) according to some fixed measure P on $(A, \mathcal{A})$. Let $\hat{A}$ denote the *reproduction alphabet*, and assume that it is finite. Let $\boldsymbol{x} = \{x_n; n \geq 1\}$ denote an infinite realization (or "message") produced by the source $\boldsymbol{X}$, and, given integers $1 \leq i \leq j \leq \infty$, write x_i^j for the string corresponding to the part of the realization between i and j, $x_i^j = (x_i, x_{i+1}, \cdots, x_j)$. We also write X_i^j for the vector of random variables $(X_i, X_{i+1}, \cdots, X_j)$, and, similarly, for sequences $\boldsymbol{y} = \{y_n; n \geq 1\}$ and $\boldsymbol{Y} = \{Y_n; n \geq 1\}$ in the reproduction alphabet.

Given an integer k, a probability mass function (pmf) Q on $\hat{A}$ is called a k-type if for every $y \in \hat{A}$, $Q(y)$ is of the form j/k for some nonnegative integer $j \leq k$.

Let ρ: $A \times \hat{A} \to [0, \infty)$ be a fixed, nonnegative (measurable) distortion measure, and define a sequence $\{\rho_n\}$ of single-letter distortion measures on $A^n \times \hat{A}^n \to [0, \infty)$ by

$$\rho_n(x_1^n, y_1^n) = \frac{1}{n} \sum_{i=1}^{n} \rho(x_i, y_i), \qquad x_1^n \in A^n, \ y_1^n \in \hat{A}^n.$$

Without loss of generality, throughout the paper we assume as usual that

$$\sup_{x \in A} \min_{y \in \hat{A}} \rho(x, y) = 0 \tag{1}$$

and also that the distortion measure ρ is bounded

$$M \triangleq \sup_{x \in A} \max_{y \in \hat{A}} \rho(x, y) < \infty.$$

Given $D \geq 0$ and a string $x_1^n \in A^n$, let $B(x_1^n, D)$ denote the distortion-ball of all n-strings in $\hat{A}^n$ that are within distortion D of x_1^n

$$B(x_1^n, D) = \{y_1^n \in \hat{A}^n \colon \rho_n(x_1^n, y_1^n) \leq D\}.$$

Given a source distribution P, we define

$$D_{\max} = \min_{y \in \hat{A}} E_P(\rho(X, y))$$

and assume that $D_{\max} > 0$. Given $D \geq 0$, let $R(D)$ denote the rate-distortion function of $\boldsymbol{X}$ with respect to $\{\rho_n\}$

$$R(D) = \inf I(X; Y) \tag{2}$$

where $I(X;Y)$ denotes the mutual information (in bits) between X and Y, and the infimum is taken over all jointly distributed random variables (X, Y) with values in $A \times \hat{A}$, such that $X \sim P$ and $E\rho(X, Y) \leq D$. Following the standard convention, we let the infimum of an empty set be equal to $+\infty$, so $R(D) = +\infty$ if there is no such pair (X, Y). It is easy to check that $R(D) = 0$ for $D \geq D_{\max}$, so we restrict our attention to the interesting range of allowable distortion values $D \in (0, D_{\max})$. Moreover, condition (1) and the finiteness of $\hat{A}$ immediately guarantee that $R(D) < \infty$ for all $D \geq 0$.

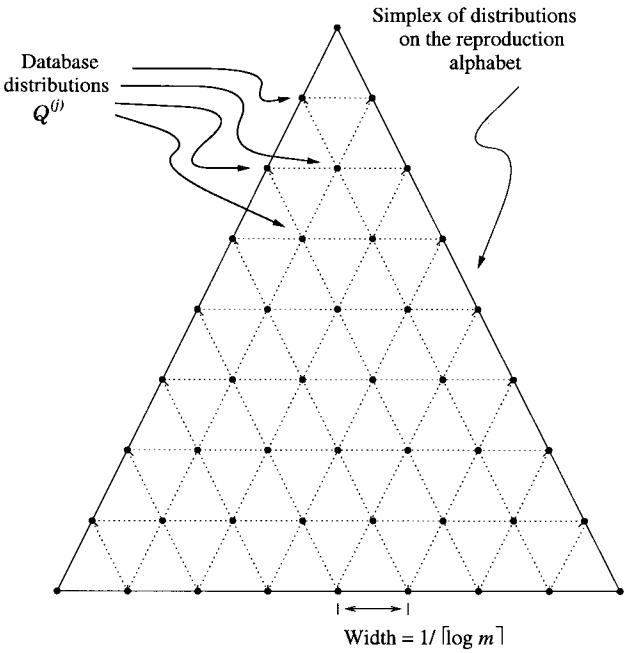

Fig. 1. The set of all $\lceil \log \ m \rceil$-types, correpsonding to the vertices of a uniform grid of width $1/\lceil \log \ m \rceil$ placed on the simplex of pmf's on $\hat{A}$.

B. The Algorithm

Let $X_1^N = (X_1, X_2, \cdots, X_N)$ be a message of length N generated by some memoryless source $\boldsymbol{X}$ of unknown distribution P on A, and let a distortion level D be fixed. Let $\{t(m)\}$ be a nondecreasing sequence of integers, write $T(m)$ for the number of $t(m)$-types on $\hat{A}$, and recall [11] that $T(m)$ is at most polynomial in $t(m)$

$$T(m) \leq [t(m) + 1]^{|\hat{A}|}. \tag{3}$$

For each m, we describe an encoding algorithm that uses $T(m)$ databases of length m.

Choose and fix an m for now. Assume that the encoder and decoder both have access to $T(m)$ memoryless databases

$$\begin{aligned}
&Y_1^{(1)}, Y_2^{(1)}, \cdots, Y_m^{(1)} \quad \text{i.i.d.} \ \sim Q^{(1)} \\
&Y_1^{(2)}, Y_2^{(2)}, \cdots, Y_m^{(2)} \quad \text{i.i.d.} \ \sim Q^{(2)} \\
&\vdots \\
&Y_1^{(T(m))}, Y_2^{(T(m))}, \cdots, Y_m^{(T(m))} \quad \text{i.i.d.} \ \sim Q^{(T(m))}
\end{aligned}$$

where each database has the same length m, they are all generated independently of the message X_1^N, and each one is i.i.d. according some $t(m)$-type $Q^{(j)}$ on $\hat{A}$, for $1 \leq j \leq T(m)$. Fig. 1 shows schematically the set of all $t(m)$-types for the specific choice of $t(m) = \lceil \log m \rceil$ (here and throughout the paper "log" denotes the logarithm taken to base two).

We can either assume that these databases are available to the encoder and decoder before the coding process begins, or that they are generated at the encoder and transmitted to the decoder using an overhead of

$$\lceil m T(m) \log \ |\hat{A}| \rceil \quad \text{bits.} \tag{4}$$

The encoding algorithm is as follows: First, the encoder calculates the length of the longest match of an initial portion

of the message, within distortion D, in any one of the databases. Let $L_{m,1} = L_{m,1}(D)$ denote the length of this longest match

$$\begin{aligned} L_{m,1} &= L_{m,1}(D) \\ &= \max\{k \geq 1 : (Y_i^{(j)}, Y_{i+1}^{(j)}, \cdots, Y_{i+k-1}^{(j)}) \in B(X_1^k, D) \\ &\quad \text{for some } 1 \leq i \leq m-k+1 \text{ and } 1 \leq j \leq T(m)\} \end{aligned}$$

and let $Z^{(1)}$ denote the initial phrase of length $L_{m,1}$ in X_1^N

$$Z^{(1)} \triangleq (X_1, X_2, \cdots, X_{L_{m,1}}).$$

Observe that $L_{m,1} \geq 1$ by assumption (1). Then the encoder describes to the decoder

a) the length $L_{m,1}$; this takes at most $C \log (L_{m,1}+1)$ bits, where C is a universal constant (cf. [14], [35]);
b) the index j of the database in which this longest match was found; this takes $\lceil \log T(m) \rceil$ bits;
c) the position i in database j where the match occurs; this takes $\lceil \log m \rceil$ bits.

Clearly, from a), b) and c) the decoder can easily recover the string

$$\hat{Z}^{(1)} = (Y_i^{(j)}, Y_{i+1}^{(j)}, \ldots, Y_{i+L_{m,1}-1}^{(j)})$$

which is within distortion D of $Z^{(1)}$. The description length of a), b), and c) is bounded above by

$$C \log (L_{m,1}+1) + \log T(m) + \log m + 2 \quad \text{bits.} \tag{5}$$

Alternatively, $\hat{Z}^{(1)}$ can be described by first describing its length $L_{m,1}$ as before, and then describing $\hat{Z}^{(1)}$ directly using

$$\lceil L_{m,1} \log |\hat{A}| \rceil \quad \text{bits.} \tag{6}$$

The encoder uses whichever one of the two descriptions is shorter, together with a one-bit flag to indicate which one was chosen. Therefore, from (5), (6), and (3), the length of the description of $Z^{(1)}$ is bounded above by

$$\begin{aligned} &\min\{C_1 \log(L_{m,1}+1) + C_2 \log(t(m)+1) \\ &\quad + \log m, C_3 L_{m,1}\} \quad \text{bits} \end{aligned} \tag{7}$$

for some fixed constants C_1, C_2, and C_3, independent of m, N, and of the source message X_1^N.

After $Z^{(1)}$ has been described within distortion D, the same process is repeated to encode the rest of the message: The encoder finds the length $L_{m,2} = L_{m,2}(D)$ of the longest string starting at position $(L_{m,1}+1)$ in X_1^N that matches within distortion D into any one of the databases, and describes

$$Z^{(2)} \triangleq (X_{L_{m,1}+1}, X_{L_{m,1}+2}, \cdots, X_{L_{m,1}+L_{m,2}})$$

to the decoder by repeating the above steps.

The algorithm is terminated, in the natural way, when the entire string X_1^N has been exhausted. At that point, X_1^N has been parsed into $\Pi_m = \Pi_m(X_1^N, D)$ distinct phrases $Z^{(k)}$, each of length $L_{m,k}$

$$X_1^N = Z^{(1)} Z^{(2)} \cdots Z^{(\Pi_m)}$$

with the possible exception of the last phrase, which may be shorter. Since each substring $Z^{(k)}$ is described within distortion D, also the concatenation of all the reproduction strings

$$\hat{Z}^{(1)} \hat{Z}^{(2)} \cdots \hat{Z}^{(\Pi_m)}$$

will be within distortion D of X_1^N.

Let $\ell_m(X_1^N) = \ell_m(X_1^N, D)$ denote the overall description length for X_1^N. From (4) and (7), $\ell_m(X_1^N, D)$ is bounded above by

$$\begin{aligned} &\lceil m T(m) \log |\hat{A}| \rceil \\ &\quad + \sum_{k=1}^{\Pi_m} \min\{C_1 \log(L_{m,k}+1) + C_2 \log(t(m)+1) \\ &\qquad + \log m, C_3 L_{m,k}\} \quad \text{bits.} \end{aligned} \tag{8}$$

The following result establishes the asymptotic optimality of this algorithm by showing that, for long messages ($N \to \infty$), the expected compression ratio achieved does not exceed the rate-distortion function $R(D)$, as m tends to infinity. In fact, a somewhat stronger result is proved, namely, that for (almost) any message emitted by the source, the compression ratio achieved, averaged over all possible databases, is asymptotically no larger than $R(D)$. Theorem 1 is proved in Section VII-D.

Theorem 1. Algorithm Optimality: Let $0 < D < D_{\max}$. If the rate $t(m)$ at which the databases are refined is $t(m) = \lceil \log m \rceil$, then, with probability one in the source messages (or "almost surely," denoted "a.s.")

$$\limsup_{m\to\infty} \limsup_{N\to\infty} E\left\{ \frac{\ell_m(X_1^N, D)}{N} \middle| X_1^N \right\} \leq R(D) \quad \text{a.s.} \tag{9}$$

where the expectation is over all databases. Therefore,

$$\limsup_{m\to\infty} \limsup_{N\to\infty} E\left\{ \frac{\ell_m(X_1^N, D)}{N} \right\} \leq R(D) \tag{10}$$

with the expectation here being over both the message X_1^∞ and the databases. Moreover, (9) and (10) hold for any choice of $t(m) \to \infty$ with $(\log t(m))/\log m \to 0$, as $m \to \infty$.

Remark: The case of lossless compression can be regarded as a special case of the above algorithm, where the encoder looks for exact matches between the source and the database. In fact, implicit in the proof of Theorem 1 is a proof that the compression ratio achieved by the lossless FDLZ algorithm [34] applied to a memoryless source $\boldsymbol{X}$ converges to the entropy rate H of $\boldsymbol{X}$, for almost all source messages:

Corollary 1. Strong Optimality of Lossless FDLZ: Let $\boldsymbol{X}$ be a discrete memoryless source of entropy rate H, and let $\tilde{\ell}_m(X_1^N)$ denote the description length for X_1^N using the FDLZ algorithm. Then

$$\limsup_{m\to\infty} \limsup_{N\to\infty} E\left\{ \frac{\tilde{\ell}_m(X_1^N)}{N} \middle| X_1^N \right\} \leq H \quad \text{a.s.}$$

III. Waiting Times Results

A. Motivation

The first extensions of the Lempel–Ziv algorithm to the lossy case [24], [29] suggested using a database of the same distribution as the source, and doing approximate string matching with respect to that database. As it later turned out [23], [38], [12], this results in strictly suboptimal compression performance. In this section we illustrate how this performance can be understood by studying an idealized coding scenario in terms of waiting times (this reduction of a practical scheme to an idealized one was introduced by Wyner and Ziv [33] and it is described in detail in [18]), and in the next section we show how the idealized coding scheme can be modified to achieve optimal compression.

Let $\boldsymbol{X}$ be a memoryless source with values in A and distribution P, and suppose that a distortion level D is chosen and fixed. Assume that the encoder and decoder both have available to them an infinite database $\boldsymbol{Y} = \{Y_n; n \geq 1\}$ with values in $\hat{A}$, distributed i.i.d. according to the pmf Q, and independent of the source $\boldsymbol{X}$. The encoder's task is to describe the first n-string X_1^n produced by $\boldsymbol{X}$ to the decoder, with distortion no more than D. This is done as follows: The encoder looks for the first position in the database where X_1^n appears within distortion D and communicates it to the decoder. We call this position the *waiting time* for X_1^n and denote it by $W_n^{(Q)}(D)$

$$W_n^{(Q)}(D) = \inf\{k \geq 1 : Y_k^{k+n-1} \in B(X_1^n, D)\}.$$

Since [14], [35] it takes approximately

$$[\log W_n^{(Q)}(D) + O(\log \log W_n^{(Q)}(D))]$$

bits to describe $W_n^{(Q)}(D)$, the rate of this code is, to first order,

$$\approx \frac{\log W_n^{(Q)}(D)}{n} \quad \text{bits per symbol.}$$

Theorem [23], [38], [12]:

$$\lim_{n\to\infty} \frac{\log W_n^{(Q)}(D)}{n} = R(P, Q, D) \quad \text{a.s.}$$

where

$$R(P, Q, D) = \inf \int H(\gamma(\cdot|x)\|Q(\cdot))\, dP(x) \tag{11}$$

$$= \inf [I(X;Y) + H(Q'\|Q)] \tag{12}$$

and the infimum is taken over all random variables (X, Y) on $A \times \hat{A}$ with $X \sim P$, and $E\rho(X, Y) \leq D$, with

$$H(R\|R') = \sum_y R(y) \log (R(y)/R'(y))$$

denoting the relative entropy between two pmf's R and R', γ denoting the conditional distribution of Y given X, and Q' denoting the marginal of Y. As before, we let $R(P, Q, D) = +\infty$ if there is no such pair (X, Y).

If we compare (12) with (2) it becomes clear that the asymptotic rate $R(P, Q, D)$ of the code is generally strictly greater than the optimal rate $R(D)$ (the rate-distortion function of $\boldsymbol{X}$). In fact, $R(D)$ satisfies

$$R(D) = \inf_Q R(P, Q, D) \tag{13}$$

(with the infimum over all pmf's Q on $\hat{A}$), so the problem is that we do not know *a priori* which database distribution achieves the infimum in (13). The simple intuition behind our algorithm is to compensate for this by using multiple databases: We allow the encoder to generate one memoryless database for each n-type on $\hat{A}$, and then encode using the best one, i.e., the one for which X_1^n has the shortest waiting time. The additional coding cost incurred is that we must identify which database was used, but since there are only polynomially many n-types this extra cost is asymptotically negligible.

B. Results

Let $\{s(n)\}$ be a nondecreasing sequence of positive integers. For each n, let $S(n)$ be the number of $s(n)$-types on $\hat{A}$ and write $Q^{(j)}$, $1 \leq j \leq S(n)$, for each one of these $s(n)$-types. Assume that for each n we have $S(n)$ processes $\boldsymbol{Y}^{(j)}$, $1 \leq j \leq S(n)$, where $\boldsymbol{Y}^{(j)}$ is independent of $\boldsymbol{X}$ and distributed i.i.d. according to $Q^{(j)}$. For each j let $W_n^{(j)}(D)$ be the waiting time until X_1^n appears in $\boldsymbol{Y}^{(j)}$ within distortion D

$$W_n^{(j)}(D) = \inf\{i \geq 1 : (Y_i^{(j)}, Y_{i+1}^{(j)}, \cdots, Y_{i+n-1}^{(j)}) \in B(X_1^n, D)\}$$

and write $W_n^*(D)$ for the shortest one of these waiting times

$$W_n^*(D) = \min_{1\leq j\leq S(n)} W_n^{(j)}(D).$$

Theorem 2. Waiting Times: Let $0 < D < D_{\max}$. If $s(n) \to \infty$ then

$$\limsup_{n\to\infty} \frac{\log W_n^*(D)}{n} \leq R(D) \quad \text{a.s.}$$

Before we give the proof of the Theorem we need to introduce some notation and definitions. First, let $R_e(D)$ denote the rate-distortion function of $\boldsymbol{X}$ in nats rather than bits, and similarly write $R_e(P, Q, D)$ for the function defined as in (11) but with relative entropy in nats rather than in bits, i.e., with $H(\cdot\|\cdot)$ replaced by $H_e(R\|R') = \sum_y R(y) \ln (R(y)/R'(y))$. Equation (13) is equivalent to

$$R_e(D) = \inf_Q R_e(P, Q, D)$$

and we write Q^* for the pmf on $\hat{A}$ that achieves the infimum. (The fact that there does exist an achieving Q^* is easy to see: Let $\{q_n\}$ be a sequence of pmf's such that $R_e(P, q_n, D) \to R_e(D)$. Since the simplex of pmf's on the finite set $\hat{A}$ is a compact (Euclidean) subset of $\mathbb{R}^{|\hat{A}|}$, the sequence $\{q_n\}$ has a convergent subsequence $\{q'_n\}$ with $q'_n \to$ some Q^*. But $R_e(P, Q, D)$ is continuous in Q for pmf's Q is a neighborhood of Q^* (this follows easily from Lemma 4 of Section VII-B by an application of the dominated convergence theorem), and $\{q'_n\}$ is a subsequence of $\{q_n\}$ so we must have $R_e(D) = R_e(P, Q^*, D)$.)

For n large enough we can choose an $s(n)$-type Q_n on $\hat{A}$ such that $Q_n(y)>0$ for all $y \in \hat{A}$, and

$$|Q_n(y) - Q^*(y)| \le \frac{|\hat{A}|}{s(n)}, \qquad \text{for all } y \in \hat{A} \qquad (14)$$

(this is outlined in Appendix I). From now and until the end of this section we assume that n is large enough so that Q_n can be chosen as above. Write $W_n(D)$ for the waiting time until a D-close version of X_1^n appears in the $\boldsymbol{Y}$-process distributed according to Q_n, let $\boldsymbol{P}$ denote the product measure P^∞ on the product space $(A^\infty, \mathcal{A}^\infty)$ of infinite sequences $\boldsymbol{x}$ drawn from A, and, similarly, write $\boldsymbol{Q}_n$ for the product measure $(Q_n)^\infty$ on $(\hat{A}^\infty, \mathcal{F})$, where $\mathcal{F}$ is the σ-field on $\hat{A}$ generated by finite-dimensional cylinders.

Proof of Theorem 2: Theorem 2 follows by combining Lemmas 1 and 2, below, together with the trivial observation that $W_n^*(D) \le W_n(D)$ with probability one.

Lemma 1 shows that asymptotically, on an exponential scale, the waiting time $W_n(D)$ for a D-close match of X_1^n into $\boldsymbol{Y}$ cannot be significantly larger than the reciprocal of the probability $\boldsymbol{Q}_n(B(X_1^n, D))$ of the event that such a match occurs. Its proof parallels those of the corresponding strong approximation theorems in [19] and [21].

Lemma 1. Strong Approximation:

$$\limsup_{n\to\infty} \frac{1}{n} \log [W_n(D) \boldsymbol{Q}_n(B(X_1^n, D))] \le 0 \quad \text{a.s.}$$

Lemma 2 is a large deviations result; it will follow by an application of the Gärtner–Ellis theorem [13, Theorem 2.3.6].

Lemma 2. Large Deviations:

$$\liminf_{n\to\infty} \frac{1}{n} \log \boldsymbol{Q}_n(B(X_1^n, D)) \ge -R(D) \quad \text{a.s.}$$

Lemmas 1 and 2 are proved in Sections VII-A and VII-B, respectively. □

IV. Algorithm Optimality

We will use the waiting times results of the previous section to prove Theorem 1, establishing the optimality of the algorithm presented in Section II-B.

First observe that there is a duality relationship between the waiting times $W_n^*(D)$ and the match lengths $L_{m,1}(D)$ in that

$$W_n^*(D) \le m - n + 1 \Rightarrow L_{m,1}(D) \ge n. \qquad (15)$$

Strictly speaking, since the definitions of $W_n^*(D)$ and $L_{m,1}(D)$ depend on the choices of the underlying sequences $\{s(i)\}$ and $\{t(j)\}$, respectively, we should say that: If $W_n^*(D)$ defined with respect to a fixed sequence $\{s(i)\}$ satisfies

$$W_n^*(D) \le m - n + 1$$

then $L_{m,1}(D)$ defined with respect to some sequence $\{t(j)\}$ such that $s(n) = t(m)$ satisfies $L_{m,1}(D) \ge n$.

Unlike in the case of no distortion, the implication in (15) is generally *not* an equivalence. Nevertheless, (15) is sufficient to translate the asymptotic upper bound for $W_n^*(D)$ of Theorem 2 to an asymptotic lower bound for $L_{m,1}(D)$.

Corollary 2. Match Lengths: Let $0<D<D_{\max}$. If $t(m)\to\infty$ then

$$\liminf_{m\to\infty} \frac{L_{m,1}(D)}{\log\ m} \ge \frac{1}{R(D)} \quad \text{a.s.}$$

The proof of Corollary 2 is a straightforward but tedious calculation, very similar to the ones done in the lossless case, and therefore omitted here. The optimality of the algorithm (proof of Theorem 1 below) essentially follows from the fact that the match lengths grow like $(\log m)/R(D)$, similarly, at least in spirit, to the lossless case, where the optimality of FDLZ follows from the fact that the lengths L_m of the longest exact matches grow like $(\log m)/H$. Unfortunately, the elegant combinatorial argument used by Wyner and Ziv in [33] and [35] no longer works when distortion is allowed, and for that reason in the proof of Theorem 1 we need a stronger bound on the (conditional) lower tails of $L_{m,1}(D)$; its proof is given in Section VII-C.

Corollary 3. Tails of Match Lengths: Let $0 < D < D_{\max}$. If $t(m) \to \infty$ then for any $\epsilon>0$

$$(\log\ m) \Pr\left\{ L_{m,1}(D) \le \frac{\log\ m}{R(D)+\epsilon} \middle| X_1^\infty \right\} \to 0 \quad \text{a.s.}$$

V. Complexity, Redundancy, and Implementation

A useful feature of the algorithm is that it provides a handle in balancing the tradeoff of encoding complexity versus compression redundancy, depending on the requirements of particular applications. This tradeoff is discussed in some more detail below. First an upper bound is given for the complexity of the algorithm, and then a heuristic argument is presented, suggesting that if the rate at which the databases are being refined is chosen appropriately the redundancy of the algorithm is of the same order as that of the lossless FDLZ (where "redundancy" refers to the difference between the expected compression ratio achieved by the algorithm and the rate-distortion function). This heuristic rate is also confirmed by brief simulation results presented in Section V-C.

A. Complexity

The worst case complexity of the algorithm can be roughly upper-bounded as follows. In the worst conceivable case, for each position of the message string (N such positions) the algorithm might have to look for a match starting in every position of each of the databases ($mT(m)$ of them), and make m comparisons between the source string and the corresponding database string, resulting in at most

$$m^2 T(m) N \quad \text{operations.}$$

Of course this is an extremely crude upper bound, but it illustrates the facts that 1) the complexity of the algorithm is certainly not exponential in the message length N; and 2) that increasing the number or the lengths of the databases improves the compression performance and also increases the complexity. Much more accurate bounds based on efficient implementations of approximate string-matching algorithms

can be derived from those reported in [2] (see also [10], [1], [9], and the references therein).

B. Redundancy Versus Complexity

There are three "terms" contributing to the redundancy of the algorithm, due to three different reasons.

i) *Finite-length databases.* Since the databases used by the algorithm are finite, the compression will not be optimal even if we encode with respect to a database with the optimal distribution. As with FDLZ in the lossless case, we expect that the penalty incurred by using a database of finite length m will be of the order of

$$O\left(\frac{\log\log m}{\log m}\right).$$

The main ingredient in deriving this rate for FDLZ [36] is the fact that the expectations of the exact match lengths L_m grow like $(\log m)/H + O(1)$, where L_m denotes the longest exact match between the initial portion of the message and a single database of length m with the same distribution as the source. We expect that, to some extent, the same behavior persists in the case when distortion is allowed, and that when only one database of distribution Q is used we will have

$$EL_{m,1}(D) = \frac{\log m}{R(P,Q,D)} + O(\log\log m), \quad \text{as } m \to \infty$$

(in the notation of Section III, and under some regularity conditions on the distortion measure ρ). This should not come as a surprise, particularly in view of [12] where it is demonstrated that, in addition to their first-order behavior, various second-order properties of the match-lengths $L_{m,1}(D)$ (when only one database of distribution Q is used) are analogous to those obtained in the lossless case (compare [12, Theorem 4] with [19, eq. (1.6), Corollary 3]).

ii) *Several databases.* If the rate $t(m)$ at which the databases are refined is polynomial in $(\log m)$, then the coding cost of identifying which database was used is also of the order of

$$O\left(\frac{\log\log m}{\log m}\right).$$

This can be verified easily by reading through the proof of Theorem 1 in Section VII-D, and it is also intuitively clear since we use $O(\log\log m)$ bits to identify one of the databases each time we describe a string of length $O(\log m)$. In general, if $t(m)$ grows at a different rate, the contribution to the redundancy is of the order of $(\log t(m))/\log m$.

iii) *Wrong database.* Finally, there is an error associated with the fact that for finite m the optimal database is (typically) not included among the databases currently available to the algorithm, so that the data is encoded with respect to a $(\log m)$-type approximation to the optimal database. In the idealized scenario of Section III, this corresponds to comparing the rate in the exponent of $\boldsymbol{Q}_n(B(X_1^n, D))$ with that of $(Q^*)^n(B(X_1^n, D))$, and (14) indicates that this difference is $O(1/n)$ for the choice $s(n) = n$. Therefore, for the algorithm in Section II-B we would expect an additional redundancy term of order

$$O\left(\frac{1}{\log m}\right)$$

corresponding to taking $t(m) = \lceil \log m \rceil$.

Combining i)–iii) suggests that the leading term in the redundancy of the algorithm is of the order of $(\log\log m)/\log m$, just like in the lossless case [36].

In particular, now it should be clear why the choice $t(m) = \lceil \log m \rceil$ was singled out in Theorem 1; because it makes the contributions of the terms in ii) and iii) comparable to the contribution of i).

C. Implementation Issues and Simulation Results

As stated in Theorem 1, the algorithm converges to optimality if the rate $t(m)$ at which the databases are refined tends to infinity while $(\log t(m))/\log m$ tends to zero. More generally, from the proof of Theorem 1 it is clear that any asymptotically dense set of database distributions will work, as long as the number $T(m)$ of available databases of length m does not grow too fast, namely, as long as $(\log T(m))/\log m \to 0$ as $m \to \infty$. Therefore, in practice, we have the freedom to choose any set of database distributions that fit the specific application better instead of uniformly covering all possible distributions (as shown in Fig. 1). In particular, prior knowledge about the distribution of the source can easily be incorporated into the structure of the algorithm.

In terms of its complexity, the algorithm is comparable to the lossless FDLZ in the following sense: For a fixed database size m, the FDLZ encoder typically searches though m possible starting positions to find an exact match of length $O(\log m)$. Similarly, when $t(m) = \lceil \log m \rceil$ in the lossy case, the encoder has to search through $T(m)$ databases (where $T(m)$ is at most polynomial in $(\log m)$—see (3)) to find an approximate match of length $O(\log m)$. And as in the lossless case, there is a very extensive literature devoted to efficient algorithms for *approximate* string matching. Implementation details and algorithmic issues are discussed at length in the text [10], and, in the context of data compression, in [1], [2], and [9].

Although the above arguments indicate that the encoding complexity of the lossy algorithm in certainly polynomial in the message length, the upper bound for the number of databases provided by (3), $T(m) \le [t(m)+1]^{|\hat{A}s|}$, reveals one of the algorithms' major practical limitations: that for finite database lengths m, the number of databases becomes unreasonably large when the reproduction alphabet $\hat{A}$ becomes large. For example, even in the (rather modest) case when $|\hat{A}| = 256$ (corresponding, say, to 256 gray levels of an image) and $m = 512$, the above upper bound for $T(m)$ becomes 10^{256}, obviously an impossibility.

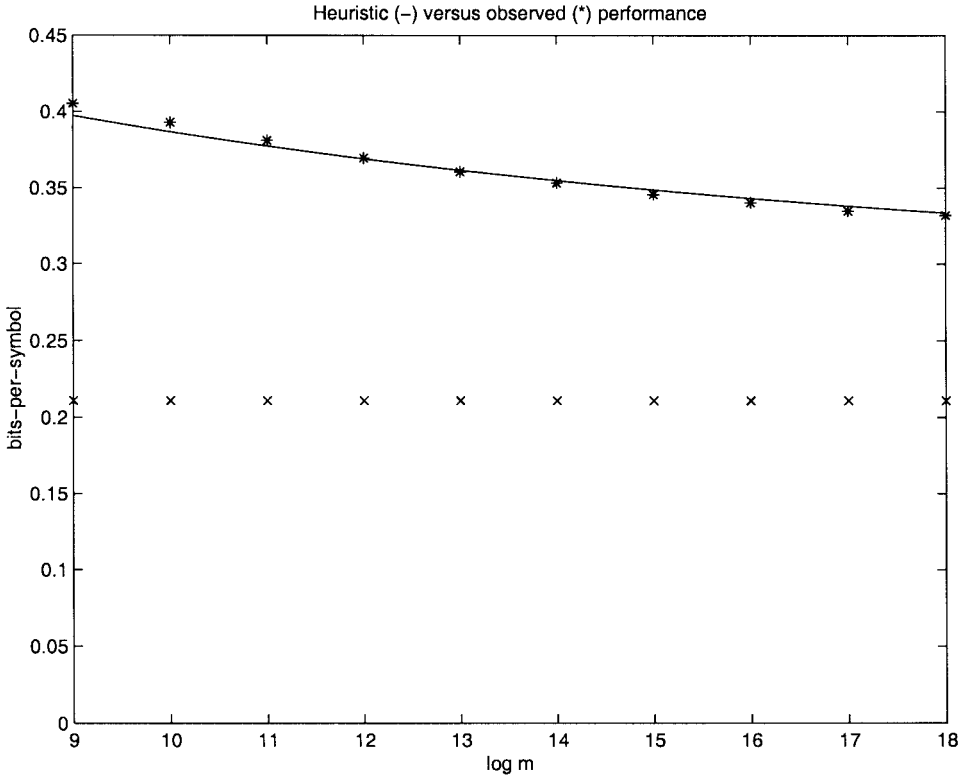

Fig. 2. Compression performance on a memoryless Bernoulli(0.4) source, with respect to Hamming distortion and $D = 0.22$. The compression ratios achieved by the algortihm for different database sizes m are denoted by $(*)$; the ideal compression ratio (rate-distortion function) is shown as $(\times)$; the performance suggested by the heuristic argument in Section V-B, namely, $R(D) + C(\log\log m)/\log m$, is shown as a solid line, with the constant $C \approx 0.53$ empirically fitted to the data.

To illustrate the algorithm's performance, we chose the simple example of lossy compression of a binary memoryless source with respect to Hamming distortion. We pseudorandomly generated binary Bernoulli(0.4) data, and implemented the algorithm as described in Section III, with some minor practical modifications.

Fig. 2 shows its compression performance on a sequence of 524 288 bits (64 kbytes), with the distortion level D set to 0.22, and for a total of 15 databases of lengths $m = 2^9, 2^{10}, \cdots, 2^{18}$ bits each. For reference, we note that typical values of m in current implementations of lossless versions of Lempel–Ziv are around $m = 2^{15}$ bits (for example, m corresponding to the window size used by LZ77 as implemented in the Unix command `gzip`; see [16]).

As in several current implementations of lossless versions of Lempel–Ziv coding, we set a maximum possible match length of 128 bits. This restriction allowed us to describe each $L_{m,1}$ using a fixed 7 bits rather than the $C\log(L_{m,1}+1)$ bits suggested in Section III.

VI. Extensions

A. A Fixed-Rate Version

We informally outline how the algorithm can be modified to provide fixed-rate lossy compression for memoryless sources. The main difference is that instead of looking for the *longest* match with distortion smaller than a fixed D, here we look for the *most accurate* match with length greater than some fixed length M.

Let R be the target rate, and recall from (7) that a string of length L in the message that matches somewhere in one of the databases, can be encoded using no more than

$$\psi_m(L) \stackrel{\Delta}{=} \min\{C_1 \log(L+1) + C_2 \log(t(m)+1) + \log m, C_3 L\} \quad \text{bits.} \tag{16}$$

To guarantee an encoding rate below R bits per symbol, we consider initial strings X_1^L of the message X_1^N of lengths L large enough so that $\psi_m(L)/L \le R$, i.e., $L \ge M_m(R)$, where

$$M_m(R) \stackrel{\Delta}{=} \min\left\{1 \le L \le m\colon \frac{\psi_m(L)}{L} \le R\right\}$$

(since the function $\psi_m(L)/L$ is nonincreasing in L, $L \ge M_m(R)$ implies $\psi_m(L)/L \le R$). Of all such strings X_1^L, choose the one that matches somewhere into one of the databases with minimal distortion; let

$$D_{m,1}(R) = \min\{\rho_L(X_1^L, (Y_i^{(j)}, Y_{i+1}^{(j)}, \cdots, Y_{i+L-1}^{(j)}))\colon M_m(R) \le L \le m, 1 \le i \le m-L+1, 1 \le j \le T(m)\}$$

and write $\Lambda_{m,1}(R)$ for the achieving L in the above definition. Then the initial string in X_1^N of length $\Lambda_{m,1}(R) \ge M_m(R)$ can be encoded, within distortion $D_{m,1}(R)$, using

$$\frac{\psi_m(\Lambda_{m,1}(R))}{\Lambda_{m,1}(R)} \le R \quad \text{bits per symbol.} \tag{17}$$

The same process can be repeated iteratively until the entire message has been encoded, yielding a total of Π substrings of X_1^N, of lengths $a_i \stackrel{\Delta}{=} \Lambda_{m,i}(R)$, and corresponding description-lengths $b_i \stackrel{\Delta}{=} \psi_m(\Lambda_{m,i}(R))$. By (17) and the log-sum inequality [8, Theorem 2.7.1] it follows that

$$\log\left[\frac{\sum_{i=1}^{\Pi} a_i}{\sum_{i=1}^{\Pi} b_i}\right] \le \left(\sum_{i=1}^{\Pi} a_i\right)^{-1} \sum_{i=1}^{\Pi}\left(a_i \log\frac{a_i}{b_i}\right) \le \log R$$

so the overall encoding rate of X_1^N is

$$\frac{\sum_{i=1}^{\Pi} a_i}{\sum_{i=1}^{\Pi} b_i} \le R \quad \text{bits per symbol.}$$

Now let us look at the distortion achieved. From the definition of ψ_m it is clear that the dominant term in the right-hand side of (16) is the $(\log m)$-term, which means that, for large m, $\psi_m(L)/L \approx (\log m)/L$ and $M_m(R) \approx (\log m)/R$. Therefore, $D_{m,1}(R)$ is the minimal distortion that can be achieved between the source and any one of the databases by strings of lengths longer than $(\log m)/R$. But from Corollary 2 we know that there exist D-close matches of length at least $(\log m)/R(D)$, which suggests that

$$\limsup_{m\to\infty} D_{m,1}(R) \le D(R) \quad \text{a.s.} \tag{18}$$

with $D(R)$ denoting the distortion-rate function of the source. So, in the same way that Corollary 2 is the essential technical ingredient in proving Theorem 1, it is plausible that the optimality of the above scheme (i.e., that the overall description of the message X_1^N is asymptotically within distortion $D(R)$) will similarly follow from (18).

B. Sources with Memory

A simple inspection of the proofs immediately reveals that all the results from Sections II–IV remain true in the case when the assumption that $\boldsymbol{X}$ is memoryless is replaced with the assumption that it is a stationary ergodic process. In particular, the asymptotic compression ratio achieved by the algorithm is equal to the first-order approximation to the rate-distortion function of $\boldsymbol{X}$, which is, in general, larger than the rate-distortion function itself. In the companion paper [21] we present a different modification of FDLZ that achieves the rate-distortion function for a wide class of processes with memory.

C. Unbounded Distortion Measures

The assumption that ρ is bounded is a technical assumption that can be significantly relaxed at the price of more complex proofs. We expect that the algorithm optimality, as well as the waiting times results of Section III, remain valid for a much more general class of distortion measures, satisfying only certain moment conditions.

D. General Reproduction Alphabets

As already mentioned in Section V, the algorithm optimality does not depend on the exact form of the database distributions chosen, as long as 1) they are asymptotically dense, and 2) their number $T(m)$ satisfies $(\log T(m))/\log m \to 0$ as $m \to \infty$. In the case of general reproduction alphabets, the algorithm can be extended in a straightforward way, by including several databases uniformly covering the space of all possible reproduction distributions. Such asymptotically dense finite covers should be possible to construct in a systematic manner, at least as long as the space of database distributions is "compact," in a natural sense.

VII. Proofs

A. Proof of Lemma 1

Fix $D \in (0, D_{\max})$, write $\hat{P}_n$ for the empirical measure induced by X_1^n on A, i.e., the measure which assigns mass $(1/n)$ to each one of the values X_i, $i = 1, 2, \cdots, n$. Recall that the $s(n)$-types Q_n were chosen such that $Q_n(y) > 0$ for all $y \in \hat{A}$. Since by (1) we have

$$\sup_{x\in A} \min_{y\in \hat{A}s} \rho(x,y) = 0$$

it follows that for any $x_1^n \in A^n$ the ball $B(x_1^n, D)$ is not empty, and hence

$$\boldsymbol{Q}_n(B(X_1^n, D)) > 0, \qquad \boldsymbol{P}-\text{a.s.} \tag{19}$$

Now choose and fix any realization $\boldsymbol{x}$, let $\epsilon > 0$ arbitrary, and assume $K \ge 1$ is some fixed constant. For any n large enough so that $e^{n\epsilon} \ge 2(n+1)$, we have

$$\begin{aligned}
&\boldsymbol{P}\times\boldsymbol{Q}_n\{W_n(D) > K \,|\, X_1^n = x_1^n\} \\
&\quad \le \boldsymbol{Q}_n\left\{Y_{in+1}^{(i+1)n} \notin B(x_1^n, D), \right. \\
&\qquad\qquad \left. \text{for all } i = 0, 1, \cdots, \left\lfloor \frac{K-1}{n} \right\rfloor \right\} \\
&\quad = [1 - \boldsymbol{Q}_n(B(x_1^n, D))]^{\lfloor K - 1/n \rfloor}
\end{aligned}$$

(since $W_n(D) \ge 1$ by definition we need not consider values of $K < 1$). Letting $K = 2^{n\epsilon}/\boldsymbol{Q}_n(B(x_1^n, D))$ above, and noting that $(1-z)^R \le 1/(Rz)$ for all $z \in (0,1)$ and $R > 0$ yields

$$\begin{aligned}
&\boldsymbol{P}\times\boldsymbol{Q}_n\left\{\frac{1}{n}\,\log[W_n(D)\boldsymbol{Q}_n(B(x_1^n, D))] > \epsilon \middle| X_1^n = x_1^n\right\} \\
&\quad \le \left[\boldsymbol{Q}_n(B(x_1^n, D))\left\lfloor \frac{\frac{2^{n\epsilon}}{\boldsymbol{Q}_n(B(x_1^n, D))} - 1}{n} \right\rfloor\right]^{-1} \\
&\quad \le \left[\frac{2^{n\epsilon} - \boldsymbol{Q}_n(B(x_1^n, D))}{n} - \boldsymbol{Q}_n(B(x_1^n, D))\right]^{-1} \\
&\quad \le 2n2^{-n\epsilon}.
\end{aligned} \tag{20}$$

Averaging over all strings $x_1^n \in A^n$ and applying the Borel–Cantelli lemma completes the proof. □

B. Proof of Lemma 2

To avoid cumbersome notation, we prove Lemma 2 in terms of natural logarithms (denoted by ln) instead of logarithms taken to base 2, i.e., we will show that

$$\liminf_{n\to\infty} \frac{1}{n} \ln \boldsymbol{Q}_n(B(X_1^n, D)) \geq -R_e(D) \quad \text{a.s.} \tag{21}$$

(recall that $R_e(D)$ is the rate-distortion function of $\boldsymbol{X}$ in nats). In the proof of (21) we will use two more technical Lemmas from [20] and [12]. Lemma 3 is simply a restatement of [12, Lemma 1] (or [20, Lemma 2.1], where it is proved). Lemma 4 ii) corresponds to [12, Proposition 1] or [20, Proposition 2.2], and Lemma 4 i) is an immediate consequence of the corresponding proof.

Lemma 3 [20], [12]: Let μ and ν be arbitrary probability measures on A and $\hat{A}$, respectively. Let

$$D_{\min}^{\mu,\nu} = E_\mu \left[\min_{y:\, \nu(y) > 0} \rho(X, y) \right]$$
$$D_{\max}^{\mu,\nu} = E_{\mu\times\nu} \left[\rho(X, Y) \right]$$

and for $\lambda, x \in \mathbb{R}$ define

$$\Lambda_{\mu,\nu}(\lambda) = E_\mu[\ln E_\nu(e^{\lambda\rho(X,Y)})]$$

and its Fenchel–Legendre transform

$$\Lambda_{\mu,\nu}^*(x) = \sup_{\lambda \leq 0} [\lambda x - \Lambda_{\mu,\nu}(\lambda)].$$

Clearly,

$$0 \leq D_{\min}^{\mu,\nu} \leq D_{\max}^{\mu,\nu} \leq M < \infty.$$

i) $\Lambda_{\mu,\nu}$ is infinitely differentiable on $(-\infty, 0)$, $\Lambda'_{\mu,\nu}(0) = D_{\max}^{\mu,\nu}$, and $\Lambda'_{\mu,\nu}(\lambda) \to D_{\min}^{\mu,\nu}$ as $\lambda \to -\infty$.

ii) $\Lambda''_{\mu,\nu}(\lambda) \geq 0$ for all $\lambda \leq 0$; if, moreover, $D_{\min}^{\mu,\nu} < D_{\max}^{\mu,\nu}$, then $\Lambda''_{\mu,\nu}(\lambda) < 0$ for all $\lambda \leq 0$.

iii) If $D_{\min}^{\mu,\nu} < D_{\max}^{\mu,\nu}$ and $D \in (D_{\min}^{\mu,\nu}, D_{\max}^{\mu,\nu})$, then there exists a unique $\lambda < 0$ such that $\Lambda'_{\mu,\nu}(\lambda) = D$ and $\Lambda_{\mu,\nu}^*(D) = \lambda D - \Lambda_{\mu,\nu}(\lambda)$. Therefore, $\Lambda_{\mu,\nu}^*(D)$ is finite, continuous, and decreasing for $D \in (D_{\min}^{\mu,\nu}, D_{\max}^{\mu,\nu})$.

Lemma 4 [20], [12]: In the notation of Lemma 3, with μ and ν as arbitrary probability measures on A and $\hat{A}$, respectively, we have:

i) For all $D \geq 0$, $R_e(\mu, \nu, D) \geq \Lambda_{\mu,\nu}^*(D)$.

ii) For all $D \in (D_{\min}^{\mu,\nu}, D_{\max}^{\mu,\nu})$, $R_e(\mu, \nu, D) = \Lambda_{\mu,\nu}^*(D)$.

(Recall that $R_e(\mu, \nu, D)$ is defined as in (11), but with relative entropy in nats instead of bits.)

Proof of Lemma 2: For all $x_1^n \in A^n$ and $\lambda \in \mathbb{R}$ define

$$\Lambda_{x_1^n}(\lambda) = \ln E_{(Q_n)^n}(e^{\lambda\rho_n(x_1^n, Y_1^n)})$$

so that, by expanding ρ_n as a sum and using independence

$$\frac{1}{n} \Lambda_{x_1^n}(\lambda n) = \frac{1}{n} \sum_{i=1}^n f_n(x_i)$$

where

$$f_n(x) = \ln E_{Q_n}(e^{\lambda\rho(x,Y)}), \qquad x \in A.$$

If we define $f(\cdot)$ on A like $f_n(\cdot)$, but with Q_n replaced by Q^*, then

$$\begin{aligned} \left| \frac{1}{n} \Lambda_{X_1^n}(\lambda n) - \frac{1}{n} \sum_{i=1}^n f(X_i) \right| &\leq \frac{1}{n} \sum_{i=1}^n |f_n(X_i) - f(X_i)| \\ &\leq \sup_{x\in A} |f_n(x) - f(x)| \\ &\leq \sup_{x\in A} |\log(1 + \epsilon_n(x))| \end{aligned}$$

where

$$\epsilon_n(x) = \frac{\sum_{y\in\hat{A}} [Q_n(y) - Q^*(y)] e^{\lambda\rho(x,y)}}{\sum_{y\in\hat{A}} Q^*(y) e^{\lambda\rho(x,y)}}.$$

But from (14)

$$|\epsilon_n(x)| \leq \frac{|\hat{A}|}{s(n)} e^{2|\lambda|M} \to 0, \qquad \text{for all } x \in A$$

which implies that

$$\left| \frac{1}{n} \Lambda_{X_1^n}(\lambda n) - \frac{1}{n} \sum_{i=1}^n f(X_i) \right| \to 0 \quad \text{a.s.} \tag{22}$$

Also, since ρ is bounded (by assumption), so is f, and by the ergodic theorem

$$\frac{1}{n} \sum_{i=1}^n f(X_i) \to E(f(X_1)) = \Lambda_{P,Q^*}(\lambda) \quad \text{a.s.} \tag{23}$$

From (22) and (23) we get that

$$\frac{1}{n} \Lambda_{X_1^n}(\lambda n) \to \Lambda_{P,Q^*}(\lambda) \quad \text{a.s.}$$

This, together with Lemma 3, allows us to apply the Gärtner–Ellis theorem [13, Theorem 2.3.6] along (almost) every realization of $\boldsymbol{X}$, to obtain that, with $\boldsymbol{P}$-probability one

$$\begin{aligned} &\liminf_{n\to\infty} \frac{1}{n} \ln \boldsymbol{Q}_n(B(X_1^n, D)) \\ &\quad = \liminf_{n\to\infty} \\ &\qquad \frac{1}{n} \ln \boldsymbol{P} \times \boldsymbol{Q}_n \left\{ \frac{1}{n} \sum_{i=1}^n \rho(X_i, Y_i^{(n)}) \leq D \middle| X_1^n \right\} \\ &\quad \geq - \inf_{z\in(0,D)} \Lambda_{P,Q^*}^*(z) \\ &\qquad \text{(by the Gärtner–Ellis theorem)} \\ &\quad = -\Lambda_{P,Q^*}^*(D) \quad \text{(by Lemma 3)} \\ &\quad = -R_e(P, Q^*, D) \quad \text{(by Lemma 4)} \\ &\quad = -R_e(D), \quad \text{(by the definition of } Q^*) \end{aligned}$$

and this proves (21) and the lemma. But notice that, in applying Lemmas 3 and 4, we are implicitly assuming that $D \in (D_{\min}^{P,Q^*}, D_{\max}^{P,Q^*})$. The outline of an argument verifying this is provided in Appendix II. □

C. Proof of Corollary 3

We follow the notation in the proofs of Theorem 2 and Lemma 1.

Let $\epsilon > 0$ be given, and pick one of the (almost all) realizations $\boldsymbol{x} = x_1^\infty$ of $\boldsymbol{X}$ such that the result of Lemma 2 holds. By Lemma 2, we can choose N_0 (depending on $\boldsymbol{x}$) large enough so that

$$\frac{1}{n}\log \boldsymbol{Q}_n(B(x_1^n, D)) > -R(D) - \frac{\epsilon}{4}, \qquad \text{for all } n \geq N_0. \tag{24}$$

Then, by the duality relationship (15) and the fact that $W_n^*(D) \leq W_n(D)$

$$\begin{aligned}\Pr\left\{L_{m,1}(D) \leq \frac{\log m}{R(D)+\epsilon}\middle| X_1^\infty = x_1^\infty\right\} &\leq \boldsymbol{P}\times\boldsymbol{Q}_n\{W_n(D) \geq m-n+1 | X_1^\infty = x_1^\infty\} \\ &= \boldsymbol{P}\times\boldsymbol{Q}_n\left\{\frac{\log W_n(D)}{n} \geq \frac{\log(m-n+1)}{n}\middle| X_1^\infty = x_1^\infty\right\}\end{aligned}$$

where $n = \lceil (\log m)/(R(D)+\epsilon)\rceil$. If we take m large enough, say $m \geq M_0$, so that $n \geq N_0$ and

$$[\log(m-n+1)]/n \geq R(D) + \epsilon/2,$$

then this is bounded above by

$$\begin{aligned}&\boldsymbol{P}\times\boldsymbol{Q}_n\left\{\frac{\log W_n(D)}{n} \geq R(D) + \frac{\epsilon}{2}\middle| X_1^\infty = x_1^\infty\right\} \\ &\quad\leq \boldsymbol{P}\times\boldsymbol{Q}_n\left\{\frac{\log[W_n(D)\boldsymbol{Q}_n(B(x_1^n,D))]}{n} \geq R(D) + \frac{1}{n}\log \boldsymbol{Q}_n(B(x_1^n,D)) + \frac{\epsilon}{2}\middle| X_1^\infty = x_1^\infty\right\} \\ &\quad\leq \boldsymbol{P}\times\boldsymbol{Q}_n\left\{\frac{\log[W_n(D)\boldsymbol{Q}_n(B(x_1^n,D))]}{n} \geq \frac{\epsilon}{4}\middle| X_1^\infty = x_1^\infty\right\}\end{aligned}$$

where the last inequality follows by (24). Finally, take $m \geq M_0$ sufficiently large to make the corresponding n large enough so that the bound (20) from the proof of Lemma 1 applies. Combining (20) with the above bounds yields

$$\Pr\left\{L_{m,1}(D) \leq \frac{\log m}{R(D)+\epsilon}\middle| X_1^\infty = x_1^\infty\right\} \leq 2n2^{-\epsilon n/4} \leq \alpha m^{-\beta}\log m$$

for some fixed constants $\alpha, \beta > 0$; since this argument holds for $\boldsymbol{P}$-almost any $\boldsymbol{x}$, the result of Corollary 3 follows. □

D. Proof of Theorem 1

Let $\epsilon > 0$ be given, and choose and fix one of the (almost all) realizations $\boldsymbol{x} = x_1^\infty$ of $\boldsymbol{X}$ such that Corollary 3 holds. Recall that the encoding algorithm parses up X_1^N into Π_m distinct words $Z^{(k)}$, each of length $L_{m,k}$. Let $n = (\log m)/(R(D)+\epsilon)$. Following [34] we assume, without loss of generality, that n is an integer and that the last phrase in the parsing of X_1^N is complete, i.e.,

$$Z^{(\Pi_m)} \text{ has length } L_{m,\Pi_m}(D).$$

We call a phrase $Z^{(k)}$ *short* if its length satisfies $L_{m,k}(D) \leq n$; otherwise, $Z^{(k)}$ is called *long*.

Next, the upper bound (8) for the description length $\ell_m(X_1^N)$ is broken into three parts

$$\begin{aligned}\ell_m(X_1^N) \leq \lceil mT(m)\log|\hat{A}|\rceil &+ C_3 \sum_{k:\, Z^{(k)} \text{ is short}} L_{m,k} \\ &+ \sum_{k:\, Z^{(k)} \text{ is long}} [C_1\log(L_{m,k}+1) \\ &+ C_2\log(t(m)+1) + \log m].\end{aligned} \tag{25}$$

The first term is nonrandom and independent of N, so that dividing by N and letting $N \to \infty$ it tends to zero. The second term, after taking its conditional expectation, can be bounded above as

$$\begin{aligned}&E\left\{C_3 \sum_{k:\, Z^{(k)} \text{ is short}} L_{m,k}\middle| X_1^N\right\} \\ &\quad\leq C_3\,\frac{\log m}{R(D)+\epsilon}\,E\left\{\sum_{k:\, Z^{(k)} \text{ is short}} \mathbb{I}_{\{L_{m,k}\leq n\}}\middle| X_1^N\right\} \\ &\quad\leq C_4(\log m)\,N\,\Pr\left\{L_{m,1}(D) \leq \frac{\log m}{R(D)+\epsilon}\middle| X_1^N\right\}\end{aligned}$$

where the first inequality follows from the definition of being "short," $C_4 = C_3/(R(D)+\epsilon)$, $\mathbb{I}_F$ denotes the indicator function of an event F, and the second inequality follows by considering not just all k's but all the possible positions on X_1^N where a short match can occur. We can now divide by N, let $N \to \infty$, and apply Corollary 3 to see that the conditional expectation of the second term in (25) also converges to zero, $\boldsymbol{P}$-almost surely.

Finally, we turn to the third—and dominant—term in (25). By the assumptions of Theorem 1, for all m large enough (independently of N and X_1^N) we have

$$C_2\,\frac{\log t(m)}{\log m} < \epsilon. \tag{26}$$

From now and until the end of the proof we assume that m is large enough for (26) to hold. Also, let Π'_m be the number of long phrases $Z^{(k)}$. Since each long $Z^{(k)}$ has length $L_{m,k} \geq n$, we must have

$$\Pi'_m n \leq N. \tag{27}$$

Now, as in the lossless case [34], we can bound above the third term in (25) by

$$\begin{aligned}C_1\Pi'_m \sum_{k:\, Z^{(k)} \text{ is long}} &\left[\frac{1}{\Pi'_m}\,\log(L_{m,k}+1)\right] \\ &+\Pi'_m\,\log m\left(1 + C_2\frac{\log t(m)}{\log m}\right)\end{aligned}$$

which, applying Jensen's inequality and (26), is bounded above by

$$\begin{aligned}&C_1\Pi'_m \log\left(\frac{1}{\Pi'_m}\sum_{k:\, Z^{(k)} \text{ is long}} (L_{m,k}+1)\right) \\ &\quad+\Pi'_m\,(\log m)\,(1+\epsilon) \\ &\quad\overset{\text{a)}}{\leq} C_1\Pi'_m\,\log\left(1+\frac{N}{\Pi'_m}\right) + \Pi'_m\,(\log m)\,(1+\epsilon)\end{aligned}$$

$$\stackrel{\text{b)}}{\le} C_1 N \frac{\Pi'_m}{N} \log\left(1+\frac{N}{\Pi'_m}\right) + \frac{N}{n}(\log m)(1+\epsilon)$$
$$\stackrel{\text{c)}}{\le} C_1 N \frac{1}{n}\log(1+n) + N(1+\epsilon)(R(D)+\epsilon)$$
$$\stackrel{\text{d)}}{=} N\left[(R(D)+\epsilon)(1+\epsilon) + C_5\frac{\log\log m}{\log m}\right]$$

where a) follows by the fact that the sum of the lengths of long phrases cannot exceed N; b) follows from (27); c) follows from (27) together with the fact that the function $x\log(1+1/x)$ is increasing for all $x>0$; and d) follows from the definition of n in terms of m, with $C_5 = 2C_1(R(D)+\epsilon)$.

Combining this with the fact that the first two terms in (25) vanish, immediately yields

$$\limsup_{m\to\infty}\limsup_{N\to\infty} E\left\{\frac{\ell_m(X_1^N,D)}{N}\middle| X_1^N\right\} \le (R(D)+\epsilon)(1+\epsilon) \quad \text{a.s.}$$

and since $\epsilon>0$ was arbitrary we get (9). Finally, (10) follows from (9) and Fatou's lemma. □

Appendix I

Choice of $s(n)$-types: Since $s(n)\to\infty$ and it is nondecreasing, for all n large enough we have

$$s(n) > |\hat{A}|\max\{1/Q^*(y): y\in\hat{A} \text{ with } Q^*(y)>0\}.$$

Pick $y_o\in\hat{A}$ with $Q^*(y_o)>0$, and define

$$Q_n(y) = \begin{cases} \frac{[s(n)Q^*(y)]}{s(n)}, & \text{if } y\ne y_o \text{ and } Q^*(y)>0 \\ \frac{1}{s(n)}, & \text{if } Q^*(y)=0 \\ 1-\sum_{y\in\hat{A},y\ne y_o} Q_n(y), & \text{if } y=y_o. \end{cases}$$

It is now trivial to check that Q_n has the required properties. □

Appendix II

Here we give an argument verifying that, if $D\in(0,D_{\max})$, then, in fact, $D\in(D_{\min}^{P,Q^*}, D_{\max}^{P,Q^*})$. First note that

$$D_{\max} = \min_{y\in\hat{A}} E_P(\rho(X,y)) \le E_{P\times Q^*}\rho(X,Y) = D_{\max}^{P,Q^*}$$

so since $D<D_{\max}$ by assumption, we also have $D<D_{\max}^{P,Q^*}$.

On the other hand, if $D<D_{\min}^{P,Q^*}$, then it is clear from the definition (11) of $R(P,Q^*,D)$ that $R(D)=R(P,Q^*,D)=\infty$, but as noted in Section II-A, this is impossible. Finally, it remains to rule out the case $D=D_{\min}^{P,Q^*}$, so we assume that, in fact, $D=D_{\min}^{P,Q^*}$, and next we outline how this leads to a contradiction: Since $R(D)$ is a convex function of D [4, p. 270] and $R(D)<\infty$ for all $D>0$, it must have a one-sided derivative at $D>0$

$$\lim_{D'\downarrow D}\frac{R(D')-R(D)}{D'-D} > -\infty$$

therefore, there exist constants $\delta>0$ and $0<C<\infty$ such that, for all $D'\in(D,D+\delta)$

$$-C \le \frac{R(D')-R(D)}{D'-D} \le \frac{R(P,Q^*,D')-\Lambda^*_{P,Q^*}(D)}{D'-D} \quad (28)$$

where the last inequality follows from the observation (13) and Lemma 4. Now from Lemmas 3 and 4 it easily follows that for each $D'>D$ there exists a $\lambda'\le 0$ such that

$$R(P,Q^*,D') = \lambda' D' - \Lambda_{P,Q^*}(\lambda') \quad (29)$$

and also it can be seen that, since $D=D_{\min}^{P,Q^*}$, the quantity

$$\lambda D - \Lambda_{P,Q^*}(\lambda)$$

increases to $\Lambda^*_{P,Q^*}(D)$ as $\lambda\to-\infty$. Combining this with (28) and (29) yields

$$-C \le \lim_{\lambda\to-\infty}\left\{\frac{\lambda' D' - \Lambda_{P,Q^*}(\lambda') - [\lambda D - \Lambda_{P,Q^*}(\lambda)]}{D'-D}\right\}$$

where the quantity in the curly brackets $\{\cdots\}$ on the right-hand side above decreases as $\lambda\to-\infty$. Therefore, taking $\lambda=\lambda'$ we get

$$-C \le \lambda'. \quad (30)$$

But from Lemma 3 we see that, by taking D' close enough to D, λ' can be made arbitrarily small toward $-\infty$, and this contradicts (30). □

Acknowledgment

The author gratefully acknowledges several interesting conversations on the subject with A. Dembo and T. Cover, and also wishes to thank W. Szpankowski and the referees for their useful comments on an early version of this paper, and for pointing out the similarities with the work of Bucklew [5], [6].

References

[1] D. Arnaud and W. Szpankowski, "Pattern matching image compression with prediction loop: Preliminary experimental results," in *Proc. Data Compression Conf.—DCC 97*. Los Alamitos, CA: IEEE Comput. Soc. Press, 1997.
[2] M. Atallah, Y. Génin, and W. Szpankowski, "Pattern matching image compression: Algorithmic and empirical results," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, pp. 614–627, July 1999.
[3] J. G. Bell, T. C. Cleary, and I. H. Witten, *Text Compression*. Englewood Cliffs, NJ: Prentice-Hall, 1990.
[4] T. Berger, *Rate Distortion Theory: A Mathematical Basis for Data Compression*. Englewood Cliffs, NJ: Prentice-Hall, 1971.
[5] J. A. Bucklew, "The source coding theorem via Sanov's theorem," *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 907–909, Nov. 1987.
[6] ——, "A large deviation theory proof of the abstract alphabet source coding theorem," *IEEE Trans. Inform. Theory*, vol. 34, pp. 1081–1083, Sept. 1988.
[7] P. A. Chou, M. Effros, and R. M. Gray, "A vector quantization approach to universal noiseless coding and quantizations," *IEEE Trans. Inform. Theory*, vol. 42, pp. 1109–1138, July 1996.
[8] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.
[9] M. Crochemore and T. Lecroq, "Pattern-matching and text-compression algorithms," *ACM Comput. Surv.*, vol. 28, no. 1, pp. 39–41, 1996.
[10] M. Crochemore and W. Rytter, *Text Algorithms*. New York: Oxford Univ. Press, 1994.
[11] I. Csiszár and J. Körner, *Information Theory: Coding Theorems for Discrete Memoryless Systems*. New York: Academic, 1981.

[12] A. Dembo and I. Kontoyiannis, "The asymptotics of waiting times between stationary processes, allowing distortion," *Ann. Appl. Probab.*, vol. 9, pp. 413–429, 1999.
[13] A. Dembo and O. Zeitouni, *Large Deviations Techniques And Applications*, 2nd ed. New York: Springer-Verlag, 1998.
[14] P. Elias, "Universal codeword sets and representations of the integers," *IEEE Trans. Inform. Theory*, vol. IT-21, pp. 194–203, 1975.
[15] R. M. Gray and D. L. Neuhoff, "Quantization," *IEEE Trans. Inform. Theory*, vol. 44, pp. 2325–2383, Oct. 1998.
[16] D. Hankerson, G. A. Harris, and P. D. Johnson, Jr., *Introduction to Information Theory and Data Compression*. Boca Raton, FL: CRC, 1998.
[17] J. C. Kieffer, "A survey of the theory of source coding with a fidelity criterion," *IEEE Trans. Inform. Theory*, vol. 39, pp. 1473–1490, Sept. 1993.
[18] I. Kontoyiannis, "Second-order analysis of lossless and lossy versions of Lempel-Ziv codes," in *31st Asilomar Conf. Signals, Systems and Computers*, (Monterey, CA, Nov. 1997).
[19] ———, "Asymptotic recurrence and waiting times for stationary processes," *J. Theoret. Probab.*, vol. 11, pp. 795–811, 1998.
[20] ———, "Recurrence and waiting times in stationary processes, and their applications in data compression," Ph.D. dissertation, Dept. Elec. Eng., Stanford Univ.,Stanford, CA, May 1998.
[21] ———, "An implementable lossy version of the Lempel-Ziv algorithm—Optimality for sources with memory," in preparation, 1999.
[22] T. Linder, G. Lugosi, and K. Zeger, "Rates of convergence in the source coding theorem, in empirical quantizer design, and in universal lossy source coding," *IEEE Trans. Inform. Theory*, vol. 40, pp. 1728–1740, Nov. 1994.
[23] T. Łuczak and W. Szpankowski, "A suboptimal lossy data compression algorithm based on approximate pattern matching," *IEEE Trans. Inform. Theory*, vol. 43, pp. 1439–1451, Sept. 1997.
[24] H. Morita and K. Kobayashi, "An extension of LZW coding algorithm to source coding subject to a fidelity criterion," in *4th Joint Sweedinsh–Soviet Int. Worksh. Information Theory* (Gotland, Sweden, 1989), pp. 105–109.
[25] J. Muramatsu and F. Kanaya, "Distortion-complexity and rate-distortion function," *IEICE Trans. Fundamentals*, vol. E77-A, pp. 1224–1229, 1994.
[26] R. M. Neuhoff, D. L. Gray, and L. D. Davisson, "Fixed rate universal block source coding with a fidelity criterion," *IEEE Trans. Inform. Theory*, vol. IT-21, pp. 511–523, Sept. 1975.
[27] D. Ornstein and P. C. Shields, "Universal almost sure data compression," *Ann. Probab.*, vol. 18, pp. 441–452, 1990.
[28] D. Ornstein and B. Weiss, "Entropy and data compression schemes," *IEEE Trans. Inform. Theory*, vol. 39, pp. 78–83, Jan. 1993.
[29] Y. Steinberg and M. Gutman, "An algorithm for source coding subject to a fidelity criterion, based upon string matching," *IEEE Trans. Inform. Theory*, vol. 39, pp. 877–886, May 1993.
[30] W. Szpankowski, "Asymptotic properties of data compression and suffix trees," *IEEE Trans. Inform. Theory*, vol. 39, pp. 1647–1659, Sept. 1993.
[31] F. M. J. Willems, "Universal data compression and repetition times," *IEEE Trans. Inform. Theory*, vol. 35, pp. 54–58, Jan. 1989.
[32] A. D. Wyner and A. J. Wyner, "Improved redundancy of a version of the Lempel–Ziv algorithm," *IEEE Trans. Inform. Theory*, vol. 35, pp. 723–731, May 1995.
[33] A. D. Wyner and J. Ziv, "Some asymptotic properties of the entropy of a stationary ergodic data source with applications to data compression," *IEEE Trans. Inform. Theory*, vol. 35, pp. 1250–1258, Nov. 1989.
[34] ———, "Fixed data base version of the Lempel-Ziv data compression algorithm," *IEEE Trans. Inform. Theory*, vol. 37, pp. 878–880, May 1991.
[35] ———, "The sliding-window Lempel-Ziv algorithm is asymptotically optimal," *Proc. IEEE*, vol. 82, pp. 872–877, June 1994.
[36] A. J. Wyner, "The redundancy and distribution of the phrase lengths of the Fixed-Database Lempel-Ziv algorithm," *IEEE Trans. Inform. Theory*, vol. 43, pp. 1452–1464, Sept. 1997.
[37] E.-H. Yang and J. C. Kieffer, "Simple universal lossy data data compression schemes derived from the Lempel-Ziv algorithm," *IEEE Trans. Inform. Theory*, vol. 42, pp. 239–245, Jan. 1996.
[38] ———, "On the performance of data compression algorithms based upon string matching," *IEEE Trans. Inform. Theory*, vol. 44, pp. 47–65, Jan. 1998.
[39] E.-H. Yang, Z. Zhang, and T. Berger, "Fixed-slope universal lossy data compression," *IEEE Trans. Inform. Theory*, vol. 43, pp. 1465–1476, Sept. 1997.
[40] R. Zamir and K. Rose, "Toward lossy Lempel-Ziv: Natural type selection," in *Proc. Information Theory Worksh.* (Haifa, Israel, June 1996), p. 58.
[41] ———, "A type generator model for adaptive lossy compression," in *Proc. IEEE Internation Symp. Information Theory* (Ulm, Germany, June/July 1997), p. 186.
[42] Z. Zhang and V. K. Wei, "An on-line universal lossy data compression algorithm by continuous codebook refinement—Part I: Basic reults," *IEEE Trans. Inform. Theory*, vol. 42, pp. 803–821, May 1996.
[43] Z. Zhang and E.-H. Yang, "An on-line universal lossy data compression algorithm by continuous codebook refinement—Part II: Optimality for phi-mixing models," *IEEE Trans. Inform. Theory*, vol. 42, pp. 822–836, May 1996.
[44] J. Ziv, "Coding of sources with unknown statistics—Part II: Distortion relative to a fidelity criterion," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 389–394, May 1972.
[45] ———, "Coding theorems for individual sequences," *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 405–412, July 1978.
[46] ———, "Distortion-rate theory for individual sequences," *IEEE Trans. Inform. Theory*, vol. IY-26, pp. 137–143, Mar. 1980.
[47] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Trans. Inform. Theory*, vol. IT-23, pp. 337–343, May 1977.
[48] ———, "Compression of individual sequences by variable rate coding," *IEEE Trans. Inform. Theory*, vol. IT-24, pp. 530–536, Sept. 1978.