Distributed Source Coding Using Syndromes (DISCUS): Design and Construction †

S. Sandeep Pradhan and Kannan Ramchandran
ECE Dept., University of Illinois at Urbana-Champaign

Abstract

We address the problem of distributed source coding, i.e. compression of correlated sources that are not co-located and/or cannot communicate with each other to minimize their joint description cost. In this work we tackle the related problem of compressing a source that is correlated with another source *which is however available only at the decoder*. In contrast to prior information-theoretic approaches, we introduce a new *constructive* and practical framework for tackling the problem based on the judicious incorporation of *channel coding* principles into this source coding problem. We dub our approach as DIstributed Source Coding Using Syndromes (DISCUS). We focus in this paper on trellis-structured constructions of the framework to illustrate its utility. Simulation results confirm the power of DISCUS, opening up a new and exciting constructive playing-ground for the distributed source coding problem. For the distributed coding of correlated *i.i.d.* Gaussian sources that are noisy versions of each other with "correlation-SNR" in the range of 12 to 20 dB, the DISCUS method attains gains of 7-15 dB in SNR over the Shannon-bound using "naive" independent coding of the sources.

1 Introduction

Consider a distributed sensor array communication system consisting of individual sensors that image a common scene independently, perhaps using independent modalities. These sensors transmit their highly correlated information to a central processing unit that forms the best picture of the scene based on a fusion of the information collected by all the sensors. If the sensors could communicate with one another, they can avoid the transmission of any "redundant" information. However such co-operation not only requires an elaborate inter-sensor distributed network, but also comes at the expense of substantial (and possibly wasteful) communication bandwidth to facilitate this.

This raises the following interesting question: what is the best one can do if there is *no communication* among the sensors, that is, if the sensors were not able to "see" each another? If the joint distribution statistics quantifying the correlation structure are known, the surprising answer is that there is theoretically *no loss in performance* under certain conditions. The caveat however is that this is only in theory, as it is based on asymptotic, non-constructive arguments from information theory (under the name of the Slepian-Wolf coding theorem [1, 2] and its extentions). There has however been, to the best of our knowledge, no prior *constructive* and non-asymptotic approach to realizing this. If such a practical framework were feasible, it could have a profound impact on the evolution of some distributed networks, eliminating the need for, or at least significantly lightening, the required inter-node communication bandwidth. In fact, a key article in the 50^{th} year Commemorative Special Issue of the Transactions on Information Theory [7] notes that

†This work was supported in part by NSF grant MIP-97-03181 (CAREER), Army Research Office grant DAAH04-96-1-0342, Office of Naval Research grant N00014-97-1-0864, and Naval Air Warfare Center (China Lake, CA) grant N68936-98-M-2366.

"*despite the existence of potential applications, the conceptual importance of (Slepian-Wolf) distributed source coding has not been mirrored in practical data compression.*" In addition to the distributed sensing scenario mentioned earlier (e.g., related to ATR, weather, seismic activities, etc.), other potential applications include stereo and multi-camera vision systems, compression of hyper-spectral imagery, distributed data-base systems, surveillance systems, and enhancement of uncoded communication systems with the addition of a coded side-channel [8], with applications to enhancement of analog TV to digital TV.

This forms the main motivation for this paper: we introduce a novel *constructive* and non-asymptotic framework for tackling the distributed source coding problem. The problem is to design a practical framework in which to optimally compress correlated sources that cannot communicate with each other and therefore cannot "see" each other: the goal is to approach the optimal performance of the system where the sources *can* "see" each other. In this paper, we address the related problem of source coding in the presence of side information available only at the decoder.

Consider first the problem where X and Y are correlated discrete-alphabet *i.i.d.* sources, and we have to compress X losslessly, with Y being known at the decoder but *not* at the encoder. To elaborate, if Y were known at both ends (see Fig. 1(a)), then the problem of compressing X is well-understood: one can compress X at the theoretical rate of its conditional entropy [1] given Y, $H(X|Y)$. But what if Y were known only at the decoder for X and not at the encoder (see Fig. 1(b))? The surprising answer is that one can still compress X using only $H(X|Y)$ bits, the same as the case where the encoder *does* know Y. That is, by just knowing the joint distribution of X and Y, without explicitly knowing Y, the encoder of X can perform as well as an encoder which explicitly knows Y. This is known as the Slepian-Wolf coding theorem [2], one of the more surprising theorems of information-theory. As might be expected of information-theoretic results however, this is an asymptotic and non-constructive statement that requires the use of Shannon-style infinite codelength arguments. The Slepian-Wolf theorem has been extended to the lossy coding of continuous-

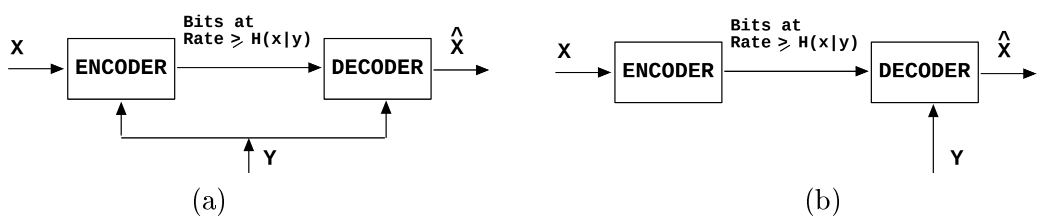


Figure 1: **Communication System: (a) Both encoder and decoder have access to the side information Y (which is correlated to X). X can be described with $H(X|Y)$ bits/sample. (b) Only decoder has access to the side information Y (which is correlated to X). The Slepian-Wolf theorem says that X can still be described with $H(X|Y)$ bits/sample.**

valued sources by Wyner and Ziv [3, 4, 5, 6], who showed that a similar result holds in the case where X and Y are correlated *i.i.d.* Gaussian random variables. If the decoder knows Y, then whether or not the encoder knows Y, the rate-distortion performance for coding X is identical[1]. As in the lossless case, the result is asymptotic and non-constructive.

While the general problem of distributed source coding has been studied constructively

[1]The only caveat is that Y has to be known *losslessly* at the decoder: the case where Y is known only partially or lossily is still an open problem with no known performance bounds!

in the literature, primarily by Flynn and Gray in [9], prior work on this topic has been restricted to a purely source quantization attack inspired essentially by extensions of the Generalized Lloyd Algorithm for quantizer design. In this work, we advocate a radical departure from this philosophy, and propose instead a framework resting heavily on channel coding principles. Of course, this being a source coding problem, there is no avoiding source quantization issues [2].

This paper is organized as follows. In Section 2, we give the intuition behind our approach using a lossless coding framework. In Section 3, we generalize this idea to lossy source coding in the presence of side information at the decoder. In Section 4, we provide an array of practical codes, based on trellises, using simulations to study their performance.

2 Intuition behind approach

As an attempt at demystifying the Slepian-Wolf coding theorem, consider a simple but interesting "riddle" that is related to the key intuition behind the result and our approach.

Suppose X and Y are equiprobable 3-bit binary words that are correlated in the following sense: the Hamming distance between X and Y is no more than one: i.e., X and Y differ in at most one of their 3 bits. If Y is available to both the encoder and decoder, clearly it is wasteful to describe X using 3 bits, as there are only 2 bits of uncertainty between X and Y (there are only 4 possibilities for the modulo-two binary sum of X and Y: $\{000, 001, 010, 100\}$, which can be indexed and sent). Now what if Y were revealed *only* to the decoder but not the encoder: could X still be described using only 2 bits of information?

A moment's thought reveals that the answer is indeed yes! The solution consists in realizing that since the decoder knows Y, it is wasteful for X to spend any bits in differentiating between $\{X = 000$ and $X = 111\}$, since the Hamming distance between these two words is 3, whereas Y is known to be within Hamming distance 1 of X. Thus, if the decoder knows that either $X = 000$ or $X = 111$, it can resolve this uncertainty by checking *which of them is closer in Hamming distance to* Y and declaring that as the value of X. Note that the set $\{000, 111\}$ is nothing but a 3-bit repetition code with a Hamming-distance of 3. Likewise, in addition to the set $\{000, 111\}$, consider the following 3 sets for X: $\{100, 011\}$, $\{010, 101\}$, and $\{001, 110\}$. Each of these sets is composed of pairs of words whose Hamming distance is 3. These are just simple variants *or cosets* of the 3-bit repetition code. While we typically use the set $\{000, 111\}$ as the 3-bit repetition code (0 is encoded as 000, and 1 as 111), it is clear that one could just as well have used any of the other three with the same performance. Further, these 4 sets cover the complete space of all possible binary 3-tuples that X can assume (the code is "perfect" in that it meets the sphere packing bound [10]).

[2]Due to the heavy influence of channel coding principles, a by-product of our framework is the novel and somewhat bizarre concept of *source coding with an "outage" (or failure) probability*. A moment's thought reveals that there is little reason for panic. First, for transmission scenarios, the question naturally arises as to why the source coder should be any more "perfect" than the channel coder which always comes with a failure rate: the system is after all limited by the weakest link in the chain. Secondly, the array of choices offered may be more attractive: the customer may well be willing to accept poor, if not outrageous, quality a very tiny percentage of the time (the outage period) for the luxury of having a much superior quality for normal operation: this is the "digital" way of life that has swept us! Finally, one can allay the fears of the source-coding purist by absorbing the failure probability into excess average source distortion.

Thus, instead of describing X by its 3-bit value, all we need to do is to encode *which coset X belongs to.* There are 4 equiprobable cosets, and this translates to a cost of 2 bits, just as in the case where Y is known to both encoder and decoder!

We can recast the solution of the above example in the language of channel codes. Recall that a channel code is specified by its three-tuple (n, k, d), where n is code length, k is the message length, and d is the minimum distance of the code. In the above example, we considered the cosets of the linear $(3, 1, 3)$ repetition code. In channel coding jargon, these cosets are associated with *syndromes* of the code: each coset has a unique syndrome associated with it. A word about the notation. Bold-faced lower and upper case letters represent vectors and matrices respectively. Recall that the syndrome $\mathbf{s}$ associated with a linear channel code is defined as $\mathbf{s} = \mathbf{Hx}$, where $\mathbf{H}$ is the parity-check matrix of the code, and $\mathbf{x}$ is any valid codeword. In words, the syndrome calculation involves verification of the parity checks of the valid codewords. The syndrome corresponding to all valid codewords is the zero-vector, since by definition all valid codewords are in the null-space of $\mathbf{H}$. A non-zero syndrome vector signals symptoms of an erroneous reception (hence the term syndrome). For example, if we consider the traditional 3-bit repetition code, there are only 2 valid codewords $\mathbf{x}$: $\{000, 111\}$. The parity-check matrix $\mathbf{H}$ for this code is given by:

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

The syndrome $\mathbf{s} = \mathbf{Hx}$ is seen to be $\{00\}$ for the two valid codewords for $\mathbf{x}$. Note also that the codevectors $\{100, 011\}$, $\{010, 101\}$, and $\{001, 110\}$ are associated with syndromes $\{11\}$, $\{10\}$, and $\{01\}$ respectively. It can be shown that the intuition behind what makes the Slepian-Wolf theorem work is similar to the intuition behind the solution to the above riddle. For every typical X, the collection of Y's that are jointly typical with that X will be, invoking the Asymptotic Equipartition Property (AEP) principle (which is essentially the Law of Large Numbers), at most a prescribed Hamming distance from X. Thus, there is a direct analogy to the above riddle[3].

The encoding of X is done as follows. The encoder observes X and sends the index of the coset in which it is resides. The decoder receives the coset index (say k) and decodes the value of X to be the code vector in the coset k, which is closest in Hamming distance to Y. The above method of encoding and decoding can be systematically implemented as follows. Let $\mathbf{H}$ be the parity check matrix of a binary linear (n, k, d) code in its systematic form, i.e., $\mathbf{H} = [\mathbf{A}|\mathbf{I}]$. Let $\mathbf{s} = \mathbf{Hx}$ be the syndrome of X when the outcome of the random variable X is $\mathbf{x}$. The encoder transmits this syndrome to the decoder. The decoder's task is to find the codevector closest to the outcome $\mathbf{y}$ of the random variable Y in the coset with syndrome $\mathbf{s}$. Instead, let $\mathbf{y}' = \mathbf{y} + \mathbf{a}$, where $\mathbf{a}$ is any codevector in the coset whose syndrome is $\mathbf{s}$ (and the addition is done modulo-2). We can first find a codevector closest

[3]Note that although connections between the Slepian-Wolf coding result and channel coding principles have been alluded to in the information-theory literature [2, 3, 8], they have been used only in an asymptotic sense to derive "existential" theoretical bounds rather than to inspire constructive approaches. Furthermore, we are unaware of any simple illustrations of the sort of the "riddle" in this section which *exactly* attain the Slepian-Wolf bound. Note also that the bound can be attained by other specific cases that involve the use of "perfect" codes: e.g., one could consider X and Y to be 7-bit words whose Hamming distance is at most one, and attain the Slepian-Wolf bound exactly by compressing X using only the 3-bit syndrome associated with the "perfect" (7,4,3) Hamming code.

to $\mathbf{y}'$ in the coset whose syndrome is the all-zero vector (referred to as principal coset). Let $\mathbf{x}'$ be such a codeword. Now it can be easily seen that $\mathbf{x} = \mathbf{x}' + \mathbf{a}$. Since any codevector in the coset with syndrome $\mathbf{s}$ will serve our purpose, we take the code vector $[\mathbf{0}|\mathbf{s}]$ to be $\mathbf{a}$ in all future discussions.

Suppose X and Y are *i.i.d.* binary random variables (equally likely to be 0 or 1) that are correlated as follows: $Y = X + e$ (modulo-2 addition), where e is an *i.i.d.* binary random variable that is independent of X. Then, using the above method for encoding X, employing an (n,k,2v+1) linear code would result in a probability of decoding success given by:

$$P = \sum_{i=0}^{v} \binom{n}{i} (1-p)^{n-i} p^i.$$

The above algorithm is not restricted to block codes. We can use convolutional codes in the same manner: as in block codes, we can define a syndrome "sequence" for convolutional codes. In the next section we consider trellis-structured codes due to their combination of superior performance and ease of implementation using soft decoding (which is key to our performance) due to the availability of the soft Viterbi algorithm.

3 Encoding with a fidelity criterion

Here we remove the constraint that X, Y belong to a binary or even discrete alphabet, and consider the *continuous-valued* case (defined on the the real line $\mathcal{R}$). Specifically, X and Y are correlated memoryless processes characterized by the discrete *i.i.d.* sequences $\{X_i\}_{i=1}^{\infty}$ and $\{Y_i\}_{i=1}^{\infty}$ respectively. We consider the specific case where the correlation between X and Y is captured as follows: Y is a noisy version of X: i.e., $Y_i = X_i + N_i$, where $\{N_i\}_{i=1}^{\infty}$ is also continuous-valued (defined on the real line $\mathcal{R}$), *i.i.d.*, and independent of the X_i's. As before, the setup is that *the decoder alone has access to the Y process*, and the task is to optimally compress the X process. For the rest of this paper, we will confine ourselves to the case where the X_i's and N_i's are zero-mean Gaussian random variables with known variances: our approach can be easily generalized to arbitrary distributions for X and N - we choose the Gaussian distribution so as to benchmark our performance against the theoretical performance bounds that are known only for the Gaussian case.

The goal is to form the best approximation $\hat{X}$ to X given an encoding bit budget of R bits per sample. We will assume block encoding with a block length n. We want to minimize the distortion measure $\rho(.)$ over the n-sequence – we assume an additive distortion measure:

$$\rho(\mathbf{x}_n, \hat{\mathbf{x}}_n) = \frac{1}{n} \sum_{i=1}^{n} \rho(x_i, \hat{x_i})$$

The encoder maps the input space ($\mathcal{R}^n$) to the index set $\{1, 2, ...2^{nR}\}$. The decoder maps the product space characterized by the encoded index set *and* the available correlated n-sequence Y to the n-sequence reconstruction $\hat{X}$: $\{1, 2, \ldots, 2^{nR}\} \times \mathcal{R}^n \longrightarrow \mathcal{R}^n$.

The encoding objective is to design a source codebook for X within a target distortion bound and *to partition the source codeword space into a bank of channel code cosets.* The minimum distance between any two codewords in a given coset should be kept as large as possible while maintaining symmetry among the cosets. In contrast to "standard" source coding where the encoder transmits the index of the source codeword that is "closest" (in the desired metric) to the source signal X, here the encoder sends, not the source codeword index, but rather the index of the *coset* containing the closest codeword, *relying on* Y *known only at the decoder to disambiguate which member in the coset* is the correct one. We need to minimize the probability of decoding failure, corresponding to the decoder picking the *wrong* coset member. For this, we need channel codes to maximize the minimum distance between coset members.

Throughout this paper, we confine ourselves to simple scalar quantization (SQ). However, However, we will explore the advantages of vector-based (sequence-based) methods for building the cosets in order to maximize the minimum distance between coset entries, in order to minimize decoding error. We first describe the case of "memoryless" coset formation.

3.1 Scalar quantization and memoryless coset-construction

Consider first a simple fixed-length (length-V) Lloyd-Max scalar quantizer designed for the p.d.f. of X. Let V=8 for ease of discussion. Let $\nabla = \{r_0, r_1, \ldots, r_{V-1}\}$ be the set of reconstruction levels as shown in Fig. 2. Note that ∇ partitions the real line into V

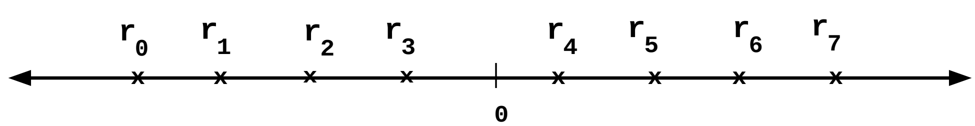


Figure 2: **Reconstruction levels of Scalar Quantizer with V=8.**

intervals each associated with one of the reconstruction levels. Let $\mathcal{T} = \{, _i\}_{i=0}^{V-1}$ be the partition of $\mathcal{R}$ as described below. $, _i$ is the open interval $\left(\frac{r_{i-1}+r_i}{2}, \frac{r_i+r_{i+1}}{2}\right)$ and we take $r_{-1} = -\infty$ and $r_V = \infty$. Now we need to partition the set ∇ into $M(\leq V)$ cosets. For illustration, let M=4. We group r_0 and r_4 into one coset. Similarly $\{r_1, r_5\}$, $\{r_2, r_6\}$ and $\{r_3, r_7\}$ are grouped into different cosets. The encoding can be described as follows:

1 Find the codeword from the set ∇ which is closest (in terms of minimizing the desired distortion metric) to the source sample X. Call this the active codeword.

2 Send the index $U \in \{0, 1, \ldots, M-1\}$ of the coset containing the active codeword.

The decoder deciphers the active codeword by finding the codeword which is closest[4] to Y in the coset whose index is sent by the encoder. After finding this codeword (say r_k), the decoder estimates X using *all* available information. We wish to minimize the expected value of the distortion $\rho(x, \hat{x})$, where $\hat{x}$ is the estimate of x. The optimal estimate $\hat{x}$ is given as follows:

$$\hat{x} = \underset{a \in \mathcal{R}}{argmin} \; E\left[\rho(X, a) \middle| \begin{array}{c} X \in , _k, \\ Y = y \end{array} \right]$$

where E(.) is the expectation operator.

[4]in the sense of $argmaxP(r_k|Y)$ which is the same as $argmin||r_k - Y||^2$ for the Gaussian case.

3.2 Scalar quantization and trellis-based coset-construction

We now describe the case of scalar quantization but using a coset construction *having memory.* In memoryless coset-construction, the cosets are built on the space ∇. Here, instead of coset-encoding X sample by sample, we do it as an n-sequence. *Note that we still use fixed-length scalar quantizers for $\{X_i\}_{i=1}^n$, but the cosets are built on the space ∇^n.* Consider the space ∇^n, and let V=8. In this space there are totally 2^{3n} distinct sequences. The task is to partition this sequence space into cosets in such a way that the minimum distance between any two sequences in a coset is made as large as possible, while maintaining symmetry among the cosets. In the following, a trellis-based partitioning is considered based on convolutional codes and set-partitioning rules as in Trellis-Coded-Modulation (TCM) [11]. Note that this is not to be confused with the Trellis-Coded-Quantization (TCQ) [12] framework: we still use simple scalar quantization but use trellises to build coset sequences.

A bitstream with R bits/unit time is used to partition 2^{R+1} codevectors taking values in $\mathcal{R}$. The set ∇ is partitioned into 4 subsets (for the sake of clarity) as before. We use Ungerboeck's 4-state trellis with the above set partitioning rules. The trellis on this set

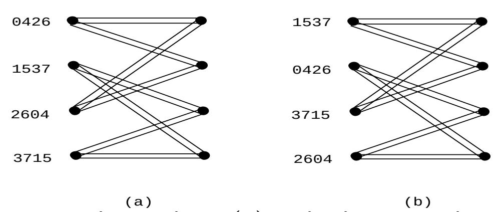


Figure 3: **Convolutional code trellis section: (a) Principal Trellis, (b) Complementary Trellis.**

is shown in Fig. 3(a), (which we call the *principal trellis*). Let $Q : \{0,1\}^3 \longrightarrow \nabla$ be the one-to-one mapping from 3-tuple binary data onto ∇ according to the following rules. $Q(\zeta) = r_\eta$ where $\zeta \in \{0,1\}^3$ is the binary representation of η. The following notation will be used in representing the function Q of sequences or vectors. $\mathbf{Q}(\mathbf{x}) = \mathbf{Q}(x_1, x_2, \ldots, x_n) = [Q(x_1), Q(x_2), \ldots, Q(x_n)]$.

We thus have a convolutional code which inserts redundancy in the input bitstream, and a carefully designed mapping from the coded stream into the set ∇. Note that we can partition the space ∇^n into 2^n cosets, each containing 2^{2n} sequences. Let $\mathbf{H}(\mathbf{t})$ be the parity check matrix polynomial of the convolutional code used in the structure. Let $\mathbf{\Theta}$ be any sequence in ∇^n, thus $\mathbf{Q}^{-1}(\mathbf{\Theta}) \in \{0,1\}^{3n}$. Let $\mathbf{S} = \mathbf{Q}^{-1}(\mathbf{\Theta})$. Thus the function $\mathbf{H}(\mathbf{t})\mathbf{S}(\mathbf{t})$ maps any $\mathbf{\Theta}$ belonging to ∇^n into $\{0,1\}^n$. We are computing the syndrome of the given codevector $\mathbf{\Theta}$: this is precisely what the encoder needs to send to the decoder.

Decoder structure: The decoder has access to the process Y in addition to the syndrome sequence sent by the encoder. In the present example, it receives n bits of syndrome and n samples of the process Y. Once the decoder gets the syndrome sequence, it recognizes the coset (containing 2^{2n} sequences) containing the the active codeword sequence. We need a computationally efficient algorithm for searching through this list. The search is for that codeword sequence which is closest to the sequence $(y_1, y_2, ..y_n)$ in terms of the given distortion measure. If the syndrome were the all-zero sequence, then we can use the Viterbi algorithm for this search in the principal coset. Here we need to modify the Viterbi decoding algorithm which is suitable for *any* syndrome sequence. Consider the k^{th} stage of the 4 state trellis as shown in Fig. 3(a). This is the trellis for the coset with an all-zero syndrome (referred to as the principal coset). Here each edge connecting one of the 4 nodes at the $(k-1)^{th}$ stage to one of the nodes at the k^{th} stage has a label associated with it. At each of the 4 nodes at the $(k-1)^{th}$ stage, the minimum-metric path (which is the distance between partially received sequences) is maintained. At the k^{th} stage, for each node, we need to compute the metrics of all the paths leading to that node and choose that path with the least metric. If the k^{th} bit of the syndrome sequence is 1 rather than 0, we need to modify the labels on each edge at the k^{th} stage. As discussed earlier, for the convolutional code under consideration, the sequence $\mathbf{Q}\left[[\mathbf{0}|\mathbf{0}|\mathbf{s}(\mathbf{t})]^T\right]$ is one of the codeword sequences in the coset whose syndrome is $\mathbf{s}(\mathbf{t})$. Thus at the k^{th} stage of decoding, if the k^{th} bit of $\mathbf{s}(\mathbf{t})$ is 1 rather than 0, we need to switch from the principal coset to the complementary coset (there are only 2 trellises in the given example: see Fig. 3). This can be thus be done at every stage in a very computationally elegant way.

Let $\mathbf{a}(\mathbf{t}) = [\mathbf{0}|\mathbf{0}|\mathbf{s}(\mathbf{t})]^T$ and a_k be the 3-tuple bit value of the sequence $\mathbf{a}(\mathbf{t})$ at time t=k. Let r_τ be the label of an edge (at the k^{th} stage) for some $0 \le \tau \le V-1$. The new label for that edge would be given as $Q\left[Q^{-1}(r_\tau) + a_k\right]$. Thus starting from $k=1$ to $k=n$, at every stage we need to keep relabeling the edges in the principal trellis used in the Viterbi decoder. Let $\mathbf{x}'(\mathbf{t})$ be the sequence which is closest to $\mathbf{y}(\mathbf{t})$ as obtained using the variant of the Viterbi algorithm discussed above, i.e., using relabeled edges. Then the active codeword sequence is $\mathbf{x}'(\mathbf{t})$, which belongs to ∇^n. Construction of more efficient and sophisticated codes is considered in [13].

Estimation of X: After obtaining the active codeword (with some probability of success in decoding), we need to estimate the sequence $\mathbf{x}(\mathbf{t})$. The optimum estimate is given as follows.

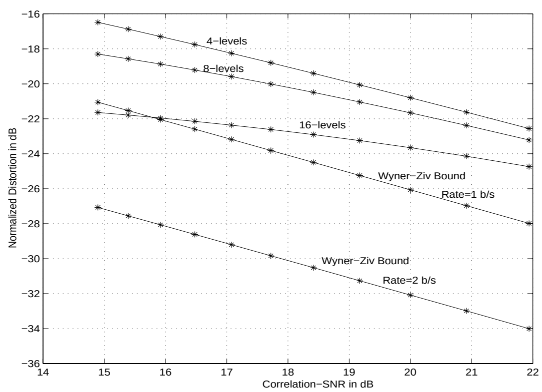
$$\hat{\mathbf{x}} = \begin{matrix} argmin \\ \mathbf{a} \in \mathcal{R}^n \end{matrix} \left[E\left\{ \rho(\mathbf{X}, \mathbf{a}) \middle| \begin{matrix} \mathbf{Y} = \mathbf{y} \\ \mathbf{X} \in \mathbf{\Gamma}(\mathbf{t}) \end{matrix} \right\} \right]$$

where $\mathbf{\Gamma}(\mathbf{t})$ is the sequence of intervals associated with each element of $\mathbf{x}'(\mathbf{t})$.

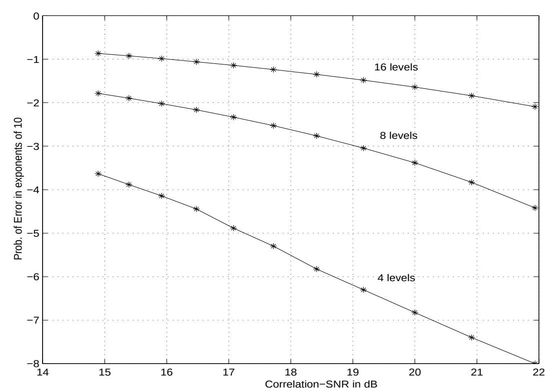
4 Design and construction: Results and Discussion

In this section, we give the design and construction of codes for the DISCUS approach and study their performance using simulations. As was noted earlier, source distortion depends only on the *root quantizer*, which is a scalar quantizer in all of our discussions in this paper. We first present an array of schemes for transmission of 1 bit per source sample.

Let us first consider memoryless encoding (uncoded system). The first choice is a fixed-length 4-level Lloyd-Max quantizer. We partition the space of 4 Voronoi regions into 2 cosets as considered in Section. 2. Similarly we can use 8 and 16 level scalar quantizers, each partitioned into 2 cosets, with each coset containing 4 and 8 Voronoi regions respectively. In our simulations, we assume the following correlation-structure for X and Y: $Y = X + N$, where X is *i.i.d* Gaussian with zero mean and unit variance. N is *i.i.d* Gaussian with zero mean and independent of X. We term the ratio of the variances of X and N as *Correlation-SNR (C-SNR)*, and the distortion as *normalized distortion* since the source is of unit variance (the method can be generalized to non-Gaussian distributions). The normalized distortion is plotted versus C-SNR for these 3 schemes in Fig. 4(a). This plot also shows the Wyner-Ziv bound [5] for 1 bit and 2 bits per source sample. Fig. 4(b) shows the probability of decoding failure for the same system. These are the results of Monte Carlo simulations, where the number of samples used for each case is 10^9. As can be noted from Fig. 4, there is a trade-off between the distortion and probability of decoding failure. The



(a)



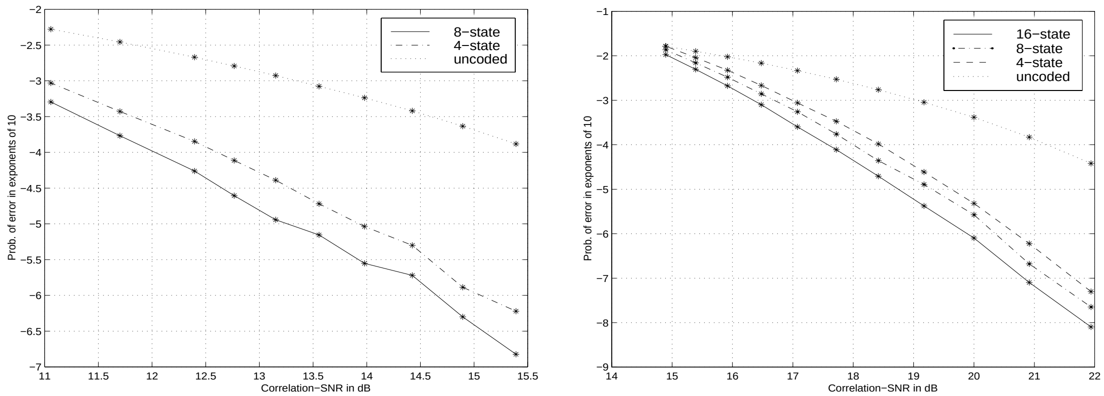
(b)

Figure 4: **(a)Normalized distortions (when decoding succeeds) for different root quantizers with Wyner-Ziv bounds are plotted as a function of Correlation-SNR (b) Uncoded system: Probability of decoding failure versus Correlation-SNR for 4,8 and 16 level quantizers for transmission of 1 bit per source sample.**

4-level quantizer can tolerate more noise ("looser" correlation structure), while its distortion performance is poor. But the 16-level quantizer has very good distortion performance (even crossing the bound)[5], at the same time operating with an "unacceptable" probability of failure. In practice, one may deem failure probabilities smaller than 10^{-4} as acceptable for many applications. 16-level quantizers have meaning only at very high C-SNR's (high correlation). Thus the choice of the scalar quantizer depends heavily on the C-SNR, or the correlation structure between X and Y.

Now consider the trellis-coded systems as discussed in Section 3. We use 4 and 8-state trellises built on a 4-level root scalar quantizer. These trellises are designed based on set-partitioning principles as given in [11]. Since these trellises are built on a 4-level alphabet, and the rate of transmission is 1 bit per source sample, we can use rate-1/2 convolutional codes. Fig. 5(a) gives the probability of decoding failure (P_{df}) versus C-SNR. Note that by going to coded systems, we get gains of around 3-4 dB in C-SNR ranges. Thus without increasing the rate, at $P_{df} \leq 10^{-4}$, we can operate at C-SNR's (correlation) no smaller than 12 dB. Similarly we can build the coded system on an 8-level quantizer. We use 4, 8 and 16 state trellises as given in [11] built on 8-level quantizer for 1 bit per source sample. Here we use systematic (3,2) convolutional codes. Fig. 5(b) gives the P_{df} versus C-SNR. Here again, we get 3-4 dB gain as we increase the number of states from 4 to 16, at $P_{df} = 10^{-4}$. Thus we have an array of solutions to the problem with transmission of 1 bit per source sample. The right choice of the solution depends on the C-SNR, or the correlation structure of X and Y. Note that for the same unit-variance Gaussian source, the traditional Shannon

[5]The Wyner-Ziv bound is crossed because there is an outage or failure probability associated with these simulations (see Fig. 4(b)).



(a) (b)

Figure 5: **Probability of decoding failure for the coded system with 4 and 8-state trellis, as compared to the uncoded system built on the same root quantizer: Gains of 3-4 dB are obtained in coded system. (a) 4-level quantizer (b) 8-level quantizer.**

rate-distortion bound for 1 bit/sample is at -6.02 dB. With 4-level root scalar quantizer, and using 8-state trellis-based DISCUS approach, the distortion at 1 bit/sample is in the range from -13 to -21 dB at C-SNR of 12 to 20 dB resp. with P_{error} less than 10^{-4}. Thus, significant gains can be attained using the DISCUS approach.

Solutions for arbitrary bitrates are considered in [13]. These are the issues of consideration in our ongoing and future research [13, 14], where we extend the DISCUS methodology to more powerful frameworks such as those based on lattice-quantization, trellis-coded-quantization and vector quantization, as well as the use of state-of-the-art channel codes, and the consideration of more general correlation structures.

References

[1] T. M. Cover and J. A. Thomas, *Elements of information theory.* New York: Wiley, 1991.

[2] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inform. Theory*, vol. IT-19, pp. 471–480, July 1973.

[3] A. D. Wyner, "Recent results in the Shannon theory," *IEEE Trans. Inform. Theory*, vol. IT-20, pp. 2–10, January 1974.

[4] A. D. Wyner, "On source coding with side information at the decoder," *IEEE Trans. Inform. Theory*, vol. IT-21, pp. 294–300, May 1975.

[5] 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. 1–10, January 1976.

[6] A. D. Wyner, "The rate-distortion function for source coding with side information at the decoder-II: General sources," *Inform. Contr.*, vol. 38, pp. 60–80, 1978.

[7] S. Verdu, "Fifty years of Shannon theory," *IEEE Trans. Inform. Theory*, vol. 44, pp. 2057–2078, October 1998.

[8] S. Shamai, S. Verdu, and R. Zamir, "Systematic lossy source/channel coding," *IEEE Trans. Inform. Theory*, vol. 44, pp. 564–579, March 1998.

[9] T. J. Flynn and R. M. Gray, "Encoding of correlated observations," *IEEE Trans. Inform. Theory*, vol. IT-33, pp. 773–787, November 1987.

[10] F. J. MacWilliams and N. J. A. Sloane, *The theory of error-correcting codes.* Elsevier-North-Holland, 1977.

[11] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Trans. on Inform. Theory*, vol. IT-28, pp. 55–67, Jan. 1982.

[12] M. W. Marcellin and T. R. Fischer, "Trellis coded quantization of memoryless and Gauss-Markov sources," *IEEE Trans. on Comm.*, vol. 38, pp. 82–93, Jan. 1990.

[13] S. S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS)," *to be submitted to IEEE Trans. Inform. Theory*, February 1999.

[14] S. S. Pradhan and K. Ramchandran, "Further results on distributed source coding using syndromes (DISCUS)," *to be submitted to IEEE Information theory workshop (South Africa)*, June 1999.