Practical Source Coding with Side Information

Lorenzo Cappellari

Dept. of Information Engineering, University of Padova
Italy

1. Introduction

The problem of coding the realizations of a random source when some other one, correlated with the former, is available at the decoder but *not* at the encoder goes under the name of *source coding with side information*. The minimum achievable transmission rates in this scenario were already found about thirty years ago by means of a random coding analysis. Practical coding schemes have been instead investigated only recently for enabling improved compression performance in sensor networks and computationally light and robust source coding in video applications.

Differently from the traditional source coding scenario, these schemes take advantage of both a *code* that is good for *channel coding* and a code that is good for *source coding*. Practical approaches where these two codes are *nested* have been shown to be asymptotically optimal, but schemes that use *independent* codes have also appeared that are easier to implement, in particular in the dual context of channel coding with side information at the encoder.

In the first half of this chapter we will review the main theoretical results regarding both this problem and, in general, the problem of *distributed source coding* (Section 2). The most important coding schemes that have appeared in literature for achieving the promises of the theoretical investigations are also described (Section 3); in particular, we discuss an approach based on the principle of *superposition* coding that we also show to be optimal.

The second half of this chapter is more focused on practical coding schemes. In Section 4 we give an original *factor graph*-based interpretation of the decoding algorithms used in the schemes for *lossless* reconstruction based on *turbo codes*. We also present a performance comparison between several of them. In Section 5 we discuss a solution to the *lossy* source coding problem with side information based on *continuous-valued syndromes*. In particular, this scheme embodies the superposition approach and uses independent channel and source codes. In order to broad the range of applications of this coding scheme, model-aided statistical decoding of continuous-valued syndromes is also proposed for the case of coding Markov sources. We compare the performance of this coding scheme against other systems both for the case of coding purely Gaussian sources and for the case of coding natural video sequences, both in the *discrete cosine transform* and in the *discrete wavelet transform* domain. We will conclude with a short discussion on the drawbacks of the proposed coding solutions and on the future research (Section 6).

Throughout the chapter, we use the following notation. The *random variable* (r.v.) X takes realizations x on the set $\mathcal{X}$ and has *probability mass function* $p(x)$. X^n is an n-dimensional random process with independent and identically distributed components; the realizations x^n are elements of $\mathcal{X}^n$. Matrices and random vectors are shown in bold face (e.g. $\mathbf{X}$). Alphabets

are usually discrete. $P[\cdot]$ and $E[\cdot]$ denote the probability of an event and the expected value of a r.v., respectively; $\chi\{\cdot\}$ is the indicator function of an event. $H(\cdot)$, $H(\cdot|\cdot)$ and $I(\cdot;\cdot)$ denote entropy, relative entropy, and mutual information; $\mathcal{A}_{\varepsilon}^{(n)}$ is the (strongly) typical set over which n-dimensional processes distribute uniformly (Cover & Thomas, 2006). A variable $X \sim \mathcal{B}(p)$ is a Bernoulli r.v. that equals one with probability p; $H(p)$ is its entropy; addition of Bernoulli variables is defined over the group $GF(2)$. A variable $X \sim \mathcal{N}(\mu, \sigma^2)$ is a Gaussian r.v. with mean μ and variance σ^2. $a \circ b$ denotes function composition, $\cdot^T$ matrix transposition; the notation 2^{nR} usually means $\lfloor 2^{nR} \rfloor$. A *good* source/channel code is a code that achieves the rate-distortion/channel capacity function asymptotically with its length.

2. Problem Statement and Theoretical Results

Suppose that we want to map some environmental parameter, e.g. the temperature, over a certain space. Then, we place $m \geq 2$ temperature sensors across that space, and have each one communicate its measurement to a central unit; let X_i denote this measurement. Data compression is employed at each sensor node in order to save transmission power. It is likely that these measurements are somewhat correlated, i.e. that $H(X_1 X_2 \ldots X_m) < \sum_{i=1}^{m} H(X_i)$. According to classical information theory results (Shannon, 1948), at least $H(X_1 X_2 \ldots X_m)$ bits must be received by the central node for describing exactly all measurements. But this is achievable with traditional source coding means only if communication between the sensor nodes *is* possible. The problem of *distributed source coding* (DSC) refers to the scenario in which the sensor nodes *are not* allowed to communicate with each other.

Closely related to DSC is the problem of *source coding with side information (at the decoder)* (SCSI). Suppose that the $m - 1$ measurements $X_2, X_3, \ldots, X_m$ are reconstructed at the central node upon receiving data from the respective $m - 1$ sensor nodes. Again, the remaining sensor node is not allowed to communicate with any of them. Then, the decoder has knowledge about the *side information* (SI) $Y \triangleq (X_2, X_3, \ldots, X_m)$, which is correlated with measurement X_1, but unavailable at the respective sensor node.

Answers to questions regarding the minimum transmission rates needed in the DSC/SCSI problem for *lossless* or *lossy* (i.e. within a certain distortion) reconstruction have been mostly given and are summarized in the following. Coding schemes achieving these rates have been also inherently suggested while answering these questions, but they are in practice not useful. Luckily, more structured coding schemes have been investigated in literature that achieve the same performance, as described in Section 3.

2.1 Distributed Source Coding

Let $\mathbf{X}$ denote the source vector $(X_1, X_2, \ldots, X_m)$. A *code of length n and rate* $(R_1, R_2, \ldots, R_m)$ for the DSC problem consists of the following functions:

$$f_i \quad : \quad \mathcal{X}_i^n \to \left[1, 2^{nR_i}\right] , \quad i = 1, 2, \ldots, m , \tag{1}$$

$$g \quad : \quad \prod_{i=1}^{m} \left[1, 2^{nR_i}\right] \to \prod_{i=1}^{m} \mathcal{X}_i^n . \tag{2}$$

Its *probability of error*, once $p(\mathbf{x})$ is known, is defined as

$$P_e^{(n)} \triangleq P\left[g \circ (f_1, f_2, \ldots, f_m)(\mathbf{X}^n) \neq \mathbf{X}^n\right] , \tag{3}$$
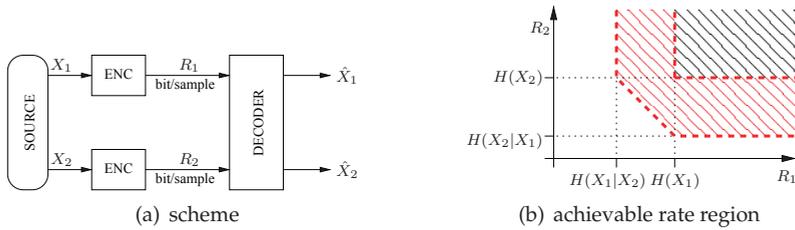
Fig. 1. Slepian-Wolf coding.

and the rate $(R_1, R_2, \ldots, R_m)$ is said to be *achievable* if there exist a sequence of codes at rate $(R_1, R_2, \ldots, R_m)$ such that $P_e^{(n)} \to 0$ as $n \to \infty$. The *achievable rate region* is the closure of the set of achievable rates.

Consider the DSC problem with two encoders ($m = 2$) shown in Fig. 1(a). The achievable rate region for this problem (Slepian & Wolf, 1973) is given by

$$R_1 \geq H(X_1|X_2), \tag{4}$$

$$R_2 \geq H(X_2|X_1), \tag{5}$$

$$R_1 + R_2 \geq H(X_1 X_2), \tag{6}$$

and is shown in Fig. 1(b). Each internal point (R_1, R_2) of this region is shown to be achievable. In particular it is shown that, among the *random partitionings* of the elements of $\mathcal{X}_i^n$ into 2^{nR_i} bins each, there exist at least one such that if

- $f_i(x_i^n)$ reveal the bin to which x_i^n belongs, and

- $g(j_1, j_2)$ returns the tuple[1] $(x_1^n, x_2^n) \in \mathcal{A}_\varepsilon^{(n)}$ with x_i^n belonging to the j_i-th bin of $\mathcal{X}_i^n$,

the code defined by these functions has asymptotically a negligible probability of error.

If $R_1 > H(X_1|X_2)$ and $R_2 > H(X_2|X_1)$, a coding scheme is indeed given for achieving a *sum rate* arbitrarily close to $H(X_1 X_2)$, i.e. to the achievable rate when the two encoders *can* communicate with each other. However, this scheme is not practical because (i) no constructive procedures are given to find the needed code, and (ii) there is no structure to be exploited in order to evaluate the functions f_i and g without resorting to huge lookup tables. Similar results can be shown for jointly ergodic sources and for the case with $m > 2$ (Cover & Thomas, 2006).

2.2 Source Coding with Side Information

Assume that in Fig. 1(a) we have $R_2 > H(X_2)$. Then, by classical results, the decoder can already reconstruct X_2 without receiving any data from the first encoder. We may then wonder which is the minimum rate needed for reconstructing $X \triangleq X_1$ when the SI $Y \triangleq X_2$ is available at the decoder, as in Fig. 2(a). From the analysis of the DSC problem, it can be inferred that lossless reconstruction of X is possible at rates over $H(X|Y)$.

---

[1] More precisely, if there are no tuples or there is more than one that satisfies this property, g assigns a random tuple to (j_1, j_2). Similar strategies are also taken by the "theoretical" algorithms discussed in the following.

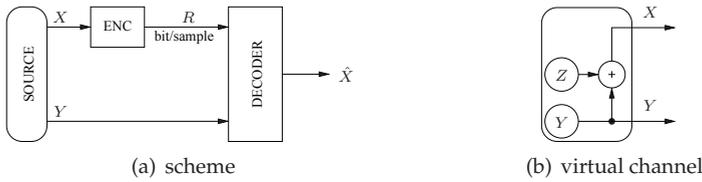(a) scheme   (b) virtual channel

Fig. 2. Wyner-Ziv coding.

From a broader point of view, it is also interesting to investigate the rates needed for lossy reconstruction with SI. If $\hat{\mathcal{X}}$ is the set over which the reconstruction $\hat{X}$ of the source at the decoder takes values, a *(single-letter) distortion function* $d : \mathcal{X} \times \hat{\mathcal{X}} \to \mathbb{R}^+$ is usually defined that is extended to n-dimensional realizations by assuming (with an abuse of notation) $d(x^n, \hat{x}^n) \triangleq \frac{1}{n} \sum_{i=1}^n d(x_i, \hat{x}_i)$.

In the SCSI scenario, a *code* of length n and rate R consists of the following functions

$$f \quad : \quad \mathcal{X}^n \to \left[1, 2^{nR}\right] , \tag{7}$$

$$g \quad : \quad \left[1, 2^{nR}\right] \times \mathcal{Y}^n \to \hat{\mathcal{X}}^n . \tag{8}$$

Its *distortion*, once $p(x, y)$ is known, is defined as

$$D^{(n)} \triangleq E\left[d\left(X^n, g\left(f(X^n), Y^n\right)\right)\right] , \tag{9}$$

and the pair (R, D) is said to be *achievable* if there exist a sequence of codes at rate R such that $\lim_{n \to \infty} D^{(n)} \leq D$. The *rate-distortion region* is the closure of the set of these achievable pairs; the *rate-distortion function* $R^*(D)$ is the infimum of rates R such that the pair (R, D) is achievable.

The rate-distortion function for SCSI equals (Wyner & Ziv, 1976)

$$R^*(D) = \min_{p(u|x), p(\hat{x}|u,y)} I(X; U|Y) , \tag{10}$$

where the minimum[2] is taken over all variables U and $\hat{X}$ such that $Y \to X \to U$ and $X \to (U, Y) \to \hat{X}$ form a Markov chain, and $E[d(X, \hat{X})] \leq D$. It is interesting to note that the rate-distortion function when *both* the encoder and the decoder have access to the SI, i.e. $R_{X|Y}(D) = \min I(X; \hat{X}|Y)$ where the minimum is over all $p(\hat{x}|x, y)$ such that $E[d(X, \hat{X})] \leq D$ (Shannon, 1959), can be similarly expressed as

$$R_{X|Y}(D) = \min_{p(u|x), p(\hat{x}|u,y)} I(X; \hat{X}|Y) , \tag{11}$$

where the minimum is taken under the same conditions of equation (10) (Pradhan et al., 2003). Since $I(X; U|Y) \geq I(X; \hat{X}|Y)$, in general, $R^*(D) \geq R_{X|Y}(D)$, with equality if and only if at optimality in (11) $X \to (\hat{X}, Y) \to U$ forms a Markov chain.

The rate loss with respect to the case where also the encoder knows the side information is investigated in (Zamir, 1996). However, the Gaussian-quadratic scenario is an important

---

[2] The search can be limited to $p(\hat{x}|u, y) = \chi\{\hat{x} = h(u, y)\}$, where h is a deterministic function.

example in which $R^*(D) = R_{X|Y}(D)$. In particular, if $\mathcal{X} = \hat{\mathcal{X}} = \mathbb{R}$ and $d(x, \hat{x}) = (x - \hat{x})^2$, there are no losses if $(X, Y) \sim \mathcal{N}$ (Wyner & Ziv, 1976) and also if $X = Y + N$, with $N \sim \mathcal{N}$ independent from Y, which in turn can have *any* distribution (Pradhan et al., 2003).

In SCSI, the coding scheme used in the proof of achievability of all rates over $R^*(D)$ involves the *random generation* of a set $\mathcal{U}$ of $2^{nI(X;U)}$ codewords distributed according to $\prod_{i=1}^n p^*(u_i)$, where $p^*(u)$ is the marginal of the joint distribution achieving the minimum in (10), and the *random partitioning* of them into $2^{nI(X;U|Y)}$ bins; each bin contains approximately $2^{nI(Y;U)}$ elements. The encoder associates each x^n to the element $u^n \in \mathcal{U}$ such that $(x^n, u^n) \in \mathcal{A}_\varepsilon^{(n)}$, and sends the index j of the bin to which u^n belongs. The decoder looks for the element u^n in the j-th bin such that $(u^n, y^n) \in \mathcal{A}_\varepsilon^{(n)}$, and returns $\hat{x}_i = h^*(u_i, y_i)$, where $h^*(u, y)$ is the deterministic function achieving the minimum in (10). Note that the bins into which $\mathcal{U}$ is partitioned take the role of *good channel codes* for the channel between Y and U; in turn, $\mathcal{U}$ takes the role of a *good source code*, since via the function h it leads to a representation close to the source. Again, this scheme is practically useless.

In many cases the decoder has perfect knowledge about the SI. One may however wonder if it is possible to characterize a rate-distortion region assuming that both $X_1 = X$ and $X_2 = Y$ are independently encoded at finite rates R_1 and R_2 and jointly decoded into representations within distortions D_1 and D_2, respectively. In the general case, this rate region is currently still unknown and only inner and outer bounds have been found. Recently, the problem has been solved for the two-terminal Gaussian-quadratic case. We refer the interested reader to (Wagner et al., 2008) and to the references therein.

3. Structured Codes for Binning

In practice, in order to achieve the performance claimed by the theory there must exist *good structured* codes and feasible algorithms to search for *jointly typical codewords* over them. Transmission rates involved in the DSC/SCSI problem are also finite. Lossless reconstruction is therefore possible only when dealing with discrete source alphabets. All practical lossless coding schemes treat essentially the binary case, that is the most simple; on the other hand, the binary representation of a discrete r.v. can be always seen as a random vector with (correlated) binary entries. Lossy reconstruction is instead possible with both binary and continuous (real) variables; the Hamming distance

$$d_H(x, \hat{x}) \triangleq \begin{cases} 0, & x = \hat{x} \\ 1, & x \neq \hat{x} \end{cases} \tag{12}$$

and the quadratic distance $d(x, \hat{x}) \triangleq (x - \hat{x})^2$ are normally used as distortion function in the two cases, respectively. In the following we will review some of the practical coding schemes recently appeared in the literature for the DSC/SCSI problem.

3.1 Near Lossless Coding

In traditional binary lossless source coding there exist many *exactly* lossless codes with performance close to the Shannon bound. On the other side, practical codes for *binary channel coding* with rates close to the Shannon capacity bound, such as the turbo codes (Berrou & Glavieux, 1996), operate in a *near* lossless regime, i.e. with very low, but not zero, probability of error. As observed in (Wyner, 1974), the problem of lossless SCSI is indeed a channel coding problem. Assume that the source and the SI are actually generated from two *independent* binary

sources Y and $Z \sim \mathcal{B}(p)$ as in Fig. 2(b); then, a *virtual "correlation" channel* exists between Y and X. This channel is a *(memoryless) binary symmetric channel* (BSC) for which *linear* codes and *closest neighbor search* algorithms are asymptotically shown to achieve capacity.

Let $\mathbf{H}$ be a *parity-check matrix* of a good (n, k) code $\mathcal{C}$ for the virtual channel. The code contains all binary vectors c^n whose *syndrome* $s^{n-k} \triangleq \mathbf{H}c^n$ is equal to zero, and partitions $\mathcal{X}^n$ into 2^{n-k} *cosets* (bins) of 2^k elements; elements in each coset $\mathcal{C}_s$ have a distinctive syndrome s^{n-k}. The code $\mathcal{C}$ can be used to reliably transmit information at its *code* rate $R_{\mathcal{C}} \triangleq k/n \approx C$ bit per channel use, where $C = 1 - H(Z)$ is the channel capacity (Cover & Thomas, 2006). The *decoding function* is defined by $f_{\mathcal{C}}(s^{n-k}) \triangleq \arg\min_{c^n \in \mathcal{C}_s} d_H(c^n, 0)$, and is such that $P[f(\mathbf{H}Z^n) \neq Z^n]$ is asymptotically negligible. Then, the code defined by (see equations (7) and (8))

$$f(x^n) \;=\; \mathbf{H}x^n \,, \tag{13}$$

$$g(s^{n-k}, y^n) \;=\; y^n + f_{\mathcal{C}}(s^{n-k} + \mathbf{H}y^n) \tag{14}$$

is a good *near* lossless code for the SCSI problem. In fact, its rate is $R = (n - k)/n = 1 - R_{\mathcal{C}} \approx H(Z) = H(X|Y)$, and

$$P[g(f(X^n), Y^n) \neq X^n] = P[Y^n + f_{\mathcal{C}}(\mathbf{H}Z^n) \neq X^n] \tag{15}$$

is asymptotically negligible.

Practically, the capacity of the BSC has been approached by turbo and *low density parity check* (LDPC) codes (MacKay, 1999). These codes have been successfully used in SCSI too. The evaluation of the syndrome in order to perform encoding is straightforward in the case of LDPC codes and very easy in the case of turbo codes. SCSI approaches based on turbo and LDPC codes are investigated in, among others, (Liveris et al., 2003; Roumy et al., 2007; Tu et al., 2005) and in (Liveris et al., 2002), respectively. In all cases the decoding function is very well approximated by means of iterative algorithms; we will give some more detail on these algorithms in Section 4.

With reference to Fig. 2(b), if $Y \sim \mathcal{B}(1/2)$, then it turns out that Z is independent from X too, such that a BSC is indeed also defined by $Y = X + Z$. SCSI approaches based on *systematic* turbo codes use this interpretation to show that if the $n - k$ *parity bits* relative to each encoded k-tuple are sent to the decoder, then near lossless performance is achieved since the SI is indeed a corrupted version of the systematic portion of a codeword (Aaron & Girod, 2002; Garcia-Frias & Zhao, 2001).

Under the same condition, a coding scheme based on syndromes was also devised for the two-terminal DSC problem that achieves all rates with the minimum sum rate. Let $\mathbf{G}$ and $\mathbf{H}$ be a *generator matrix* and a *parity-check matrix* of a good (n, k) code $\mathcal{C}$ for the virtual channel between $X_2 \triangleq Y$ and $X_1 \triangleq X$. Chose two subcodes $\mathcal{C}_1$ and $\mathcal{C}_2$ of $\mathcal{C}$ that admit generator matrices $\mathbf{G}_1$ and $\mathbf{G}_2$ constructed by taking $k_1 \leq k$ and the remaining $k_2 = k - k_1$ rows of $\mathbf{G}$, respectively. Each $c^n \in \mathcal{C}$ has a unique factorization into the sum of two codewords belonging to these two subcodes; denote with π_i the function that gives the codeword relative to code $\mathcal{C}_i$. Denote with $\mathbf{H}_i$ a parity-check matrix for the code $\mathcal{C}_i$, and with $t_i(s_i^{n-k_i})$ a deterministic function that gives an n-tuple of $\mathcal{C}_{i_{s_i}}$. Define the function

$$h(s_1^{n-k_1}, s_2^{n-k_2}) \triangleq \left(t_1(s_1^{n-k_1}) + t_2(s_2^{n-k_2}) \right) + f_{\mathcal{C}} \left(\mathbf{H} \left(t_1(s_1^{n-k_1}) + t_2(s_2^{n-k_2}) \right) \right) \,. \tag{16}$$

Then, the code defined by (see equations (1) and (2))

$$f_i(x_i^n) = \mathbf{H}_i x_i^n , \tag{17}$$

$$g_i(s_1^{n-k_1}, s_2^{n-k_2}) = t_i(s_i^{n-k_i}) + \pi_i \circ h(s_1^{n-k_1}, s_2^{n-k_2}) \tag{18}$$

has rate that satisfies $R_i \geq (n-k)/n \approx H(X_1|X_2) = H(X_2|X_1)$ and $R_1 + R_2 = (2n-k)/n = 2 - R_{\mathcal{C}} \approx 1 + H(Z) = H(X_1 X_2)$, and

$$P[g_i(f_1(X_1^n), f_2(X_2^n)) \neq X_i^n] \tag{19}$$

is asymptotically negligible. This is a direct consequence of the fact that, if $c_i^n \in \mathcal{C}_i$ is such that $t_i(\mathbf{H}_i x_i^n) = x_i^n + c_i^n$, $P[h(\mathbf{H}_1 X_1^n, \mathbf{H}_2 X_2^n) \neq C_1^n + C_2^n]$ is asymptotically negligible since $X_1 + X_2 \sim \mathcal{B}(p)$.

The scheme outlined above for encoding two sources was devised in (Pradhan & Ramchandran, 2005) and essentially tested using *convolutional* codes. Convolutional codes are used in (Pradhan & Ramchandran, 2003) too in the SCSI context. The extension to $m \geq 2$ sources is delineated in (Stanković et al., 2006), and it is shown to be optimal for *uniform* sources such that $\sum_{i=1}^{m} X_i \sim \mathcal{B}(p)$; tests conducted with more performing codes (e.g. turbo) show a very good performance.

3.2 Lossy Coding

In case of reconstruction within a certain distortion, the search for practical SCSI algorithms is more involved since both a good channel and a good source code are needed. In the seminal work (Zamir et al., 2002), it is shown that *nested* linear codes and *lattices* achieve the theoretical bounds in the binary-Hamming and in the Gaussian-quadratic case, respectively.

In practice, the suggested coding scheme requires that a *fine* linear/lattice code $\mathcal{C}_1$ is used that is a good source code for the problem of coding the source without SI. Instead of sending to the decoder the index of the codeword that is the closest to the actual realization of the source (as in traditional source coding), the encoder sends the index of the coset to which this closest codeword belongs. The codewords of $\mathcal{C}_1$ are in fact partitioned into cosets once a *coarse* linear/lattice subcode $\mathcal{C}_2 \subset \mathcal{C}_1$ is defined. If $\mathcal{C}_2$ is a good channel code for the virtual channel between the SI and this closest codeword, then the decoder can reconstruct the latter with negligible probability of error, as shown in the previous section. From it, and using also the SI, an estimate of the source is eventually obtained. The theoretical bounds are achieved because, with good linear/lattice codes, (i) the closest codeword distributes as the variable $U^n \sim \prod_{i=1}^{n} p^*(u_i)$, (ii) at the decoder, once u^n is reconstructed, the distortion is minimized by implementing the function $h^*(u, y)$, and (iii) the number of cosets, i.e. the *nesting ratio* of the two codes is approximately $I(X; U|Y)$.

In lossless SCSI the virtual channel considered for channel code design is between Y and X; here, instead, it is between Y and U. But $Y \to X \to U$ forms a Markov chain, so the channel code must be actually more robust; this phenomenon is investigated in detail in (Zamir et al., 2002) under the name of *self-noise*.

In particular, in the binary-Hamming case ($Z \sim \mathcal{B}(p)$) optimality in (10) is achieved by $U = X + Q$ with $Q \sim \mathcal{B}(D)$ independent from X, and by $h^*(u, y) = u$ (see Fig. 3(a)). In practice, dithered quantization over a fine code guaranteeing distortion D achieves *almost* this U (Zamir & Feder, 1996), but with a coarse channel code that is *exponentially* good for the BSC $U = Y + Z_e$, where $Z_e \triangleq Z + Q \sim \mathcal{B}(p * D)$ with $p * D \triangleq p(1 - D) + D(1 - p)$, the recovery of U from the SI at the decoder is still successful (Zamir et al., 2002).
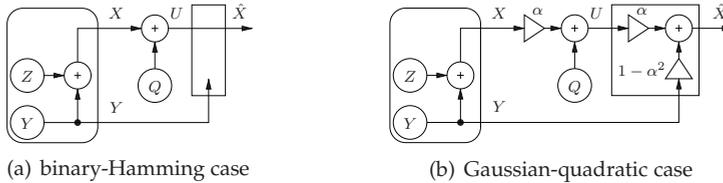
(a) binary-Hamming case   (b) Gaussian-quadratic case

Fig. 3. Relations among the random variables involved in lossy SCSI at optimality: Y, Z and Q are independent Gaussian/Bernoulli random variables. U is not actually "sent" to the decoder, but rather reconstructed relying on the knowledge of Y.

In the Gaussian-quadratic case ($Z \sim \mathcal{N}(0, \sigma_z^2)$) optimality in (10) is achieved by $U = \alpha X + Q$ with $Q \sim \mathcal{N}(0, D)$ independent from X, and by $h^*(u, y) = \alpha u + (1 - \alpha^2)y$; $\alpha \triangleq \sqrt{1 - D/\sigma_z^2} < 1$ guarantees that (i) $h^*(u, y)$ gives the minimum mean square error estimate of X from Y and U, and (ii) $E[d(X, h^*(U, Y))] = D$ (see Fig. 3(b)). In this case, dithered quantization of αX over a fine lattice guaranteeing distortion D achieves *essentially* this U (Zamir & Feder, 1996)[3]; if the coarse sublattice is *exponentially* good for the *additive white Gaussian noise* (AWGN) channel $U = \alpha Y + Z_e$, where $Z_e \triangleq \alpha Z + Q \sim \mathcal{N}(0, \sigma_z^2)$, then U can be recovered from αY at the decoder (Zamir et al., 2002).

The nested coding approach outlined above, for which the existence of good *and* nested codes was actually showed in later papers, is surely more practical than random binning; practical implementations of the Gaussian-quadratic case are indeed investigated in (Liu et al., 2006; Servetto, 2007). However, there are still practical difficulties in its implementation due to the fact that good performance is achieved only asymptotically with the length n of the code. In practice, for finite n recovering of U is possible with the coarse code having a reduced code rate with respect to the one achieving optimality, not only because the *self-noise* (Q) actually introduced is not Bernoulli/Gaussian, but also because the nesting constraint does not permit to efficiently randomize the channel code. This leads to a rate loss with respect to the achievable rate, or, equivalently, to a distortion gap, which in particular increases for fixed n with the rate of the SCSI scheme. This phenomenon was observed in (Liu et al., 2006), and a scheme was proposed in which a second coding stage (based on schemes shown in the previous section) on top of nested quantization with finite n is used to losslessly send the coset index to the SI-aware decoder; the resulting scheme is probably the most performing appeared in literature.

Other practical schemes for lossy SCSI of continuous sources are based on the observation that once traditional (vector) quantization produces an index in a discrete domain, lossless SCSI means could be used to send this index to the decoder. In this way, a good source code and a good channel code are indeed individually designed. Channel codes that are essentially based on convolutional codes are used in (Pradhan & Ramchandran, 1999; 2003; 2005); more performing channel codes are used instead in (Yang et al., 2008), where the problem of lossy distributed coding of many sources is tackled as well. As a remark, note that even if this "separated" approach has indeed been shown to be optimal for the two-terminal Gaussian-quadratic case (Wagner et al., 2008), in general it is not.

---

[3] Note that, if $Y \sim \mathcal{N}$, $U' \triangleq \alpha U$ assumes the conditional distribution $p(u'|x)$ which minimizes $I(X; U')$ under distortion D (Cover & Thomas, 2006).

3.2.1 Superposition Coding

It should be clear that the joint design of good source/channel codes for the lossy SCSI/DSC problem is very hard. Coding schemes in which optimality is also achieved if the two codes are designed independently are then preferable since they are more practical. One such scheme has been proposed and investigated for the dual problem of *channel coding with side information (at the encoder)* (CCSI) and is briefly discussed in the following. The SCSI approach we propose in Section 5 is indeed dual to this approach; more details regarding this duality can be found in (Cappellari, 2009).

Consider the additive channel $X = \hat{X} + Y + Z$, where $\hat{X}$ is the input to the channel, Y is an *interference* known to the encoder (but not to the decoder) and independent from $\hat{X}$, Z is an unknown *noise* independent from $\hat{X}$ and Y, and X is the channel output. The goal is to transfer the maximum amount of information to the decoder once a *cost* constraint is given to $\hat{X}$; usually this constraint is given as $E[d(0, \hat{X})] \leq W$ where d is a distortion function. A *capacity-cost* function $C^*(W)$ is then defined that gives this maximum amount; in particular, it was evaluated (for the general, *non*-additive setting) in (Gel'fand & Pinsker, 1980).

A random coding scheme is suggested too as follows. Generate a set $\mathcal{U}$ of codewords distributed according to a suitable distribution $\prod_{i=1}^{n} p^*(u_i)$, and partition it into bins; associate each bin to a different message to be sent. The encoder selects the u^n in the desired bin such that $(u^n, y^n) \in \mathcal{A}_\varepsilon^{(n)}$ and inputs to the channel the codeword $\hat{x}_i = h^*(u_i, y_i)$, for a suitable choice of h^*. The decoder looks for the element u^n in $\mathcal{U}$ such that $(x^n, u^n) \in \mathcal{A}_\varepsilon^{(n)}$, and returns the message corresponding to the bin to which this element belongs. Now, the bins into which $\mathcal{U}$ is partitioned take the role of *good source codes*, since via the function h a codeword satisfying the cost constraint can be formed and transmitted; in turn, $\mathcal{U}$ takes the role of a *good channel code*.

In the additive setting, nested linear codes and lattices exist that achieve optimality both in case of binary-noise Hamming-cost and in case of Gaussian-noise quadratic-cost, respectively (Zamir et al., 2002). Nevertheless, optimality is also achieved with a *superposition* of two independent codes (Bennatan et al., 2006). In particular, a *coarse* code $\mathcal{C}_2$ which is a good source code with respect to the distortion constraint W is used by the encoder for selecting the codeword to be transmitted on the channel. The *fine* code $\mathcal{C}_1$ is generated as the sum of $\mathcal{C}_2$ and a code $\mathcal{C}_0$ which is a good channel code with respect to the noise Z, i.e. $\mathcal{C}_1 = \mathcal{C}_0 + \mathcal{C}_2$. With a proper generation of $\mathcal{C}_0$, $\mathcal{C}_1$ is partitioned into bins $c_0^n + \mathcal{C}_2$, for $c_0^n \in \mathcal{C}_0$, i.e. each $c_0^n \in \mathcal{C}_0$ is in one-to-one correspondence with a bin of $\mathcal{C}_1$. At the encoder the message is then selected in terms of a $c_0^n \in \mathcal{C}_0$. At the decoder a codeword of $\mathcal{C}_1$ plus some noise is received; upon decoding this signal over the sum of $\mathcal{C}_0$ and $\mathcal{C}_2$, the best estimate of c_0^n is declared as the received message. Details and implementation of this scheme with practical codes is discussed in (Bennatan et al., 2006).

In the lossy SCSI problem, let us in principle generate a *coarse* code $\mathcal{C}_2$ which is a good channel code with respect to the virtual channel between Y and U. Then, we generate $\mathcal{C}_0$ as a good source code for Z_e with respect to the distortion D, and obtain the *fine* code $\mathcal{C}_1 = \mathcal{C}_0 + \mathcal{C}_2$. In particular, in the binary case ($Z \sim \mathcal{B}(p)$) $\mathcal{C}_2$ and $\mathcal{C}_0$ are codes made of 2^{nR_2} and 2^{nR_0} codewords with distribution $\mathcal{B}(1/2)$ and $\mathcal{B}(q)$, respectively. The encoder looks for $c_0^n + c_2^n$ such that $(x^n, c_0^n + c_2^n) \in \mathcal{A}_\varepsilon^{(n)}$ and sends the index of c_0^n (in practice, dithered quantization can be employed); there is an encoder error if $d_H(x^n, c_0^n + c_2^n) > D$.
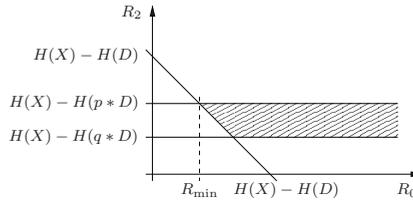
Fig. 4. Region in which there are no encoder nor decoder errors.

Theorem 1. *The probability of having an encoder error is negligible with $n \to \infty$ if*

$$R_2 \quad > \quad H(X) - H(q * D) \tag{20}$$
$$R_0 + R_2 \quad > \quad H(X) - H(D) \,. \tag{21}$$

Sketch of proof: Asymptotically, x^n takes values on a set of $2^{nH(X)}$ elements, and for a fixed c_2^n, the code $\mathcal{C}_0 + c_2^n$ covers at most $2^{nH(q*D)}$ of them within distortion D. Hence, there must be at least $2^{nH(X)-nH(q*D)}$ codewords in code $\mathcal{C}_2$. Then, since each n-tuple approximates at most $2^{nH(D)}$ elements within Hamming-distortion D, the superposition code must provide at least $2^{nH(X)-nH(D)}$ codewords. Once we have these two conditions the probability that a typical $(c_0^n + c_2^n)$ is within distortion D approaches one. This may be proved with an argument similar to the one used in the standard proof of the achievability of the rate-distortion function (Cover & Thomas, 2006). If both given bounds are approached, the *test channel* between $C_0 + C_2$ and X is such that $X = C_0 + C_2 - Q$, with $Q \sim \mathcal{B}(D)$ independent from C_0 and C_2 (asymptotically); for rates over the bounds a lower distortion can be achieved.

The decoder looks for $c_2^n \in \mathcal{C}_2$ such that $(c_0^n + c_2^n, y^n) \in \mathcal{A}_\varepsilon^{(n)}$ (in practice, a closest neighbor search is conducted), and reconstructs the source as $c_0^n + c_2^n$; there is a decoding error if c_2^n is not correctly recovered. But once c_0^n is known, the equivalent noise between Y and $C_2 = Y + Z + Q - C_0$ is distributed as Z_e. Hence, the following holds.

Theorem 2. *The probability of having a decoder error is negligible with $n \to \infty$ if*

$$R_2 < C_{p*D}(W) \,, \tag{22}$$

where $C_{p*D}(W) \triangleq H(W * p * D) - H(p * D) = H(X) - H(p * D)$ is the capacity of the BSC between Y and U, subject to a Hamming-cost constraint W such that $Y + Z + Q$ distributes as X (i.e. the balls related to codewords of $\mathcal{C}_2$ are packed *only over the space in which x^n lies).*

The closure of the rate region where there are no errors is shown in Fig. 4; it is not empty for any q such that $H(q * D) \geq H(p * D)$, and the minimum of R_0 approaches the rate-distortion function $R^*(D) = H(p * D) - H(D)$ for all distortions $0 \leq D \leq d_C < p$ (Wyner & Ziv, 1976). In the Gaussian case ($Z \in \mathcal{N}(0, \sigma_z^2)$), $\mathcal{C}_0$ is generated from the distribution $\mathcal{N}(0, \sigma_q^2)$ and again the scale factor $\alpha = \sqrt{1 - D/\sigma_z^2}$ is used. If R_2 and $R_0 + R_2$ are high enough, dithered quantization of αX over the superposition of codes $\mathcal{C}_2$ and $\mathcal{C}_0$ leads asymptotically to the relation $\alpha X = C_0 + C_2 - Q$, with $Q \sim \mathcal{N}(0, D)$ independent from C_0 and C_2. At the decoder, which knows c_0^n, the channel between αY and C_2 satisfies $C_2 = \alpha Y + \alpha Z + Q - C_0$, i.e. it is Gaussian, with capacity $C(W) \triangleq (1/2) \log(1 + W/(\alpha^2 \sigma_z^2 + D)) = (1/2) \log(1 + W/\sigma_z^2)$ under the quadratic cost constraint W. There are no decoding errors if the code rate R_2 is

below this capacity, and, again, the best estimate of the source at the decoder can be found by taking $\alpha(c_0 + c_2) + (1 - \alpha^2)y$. The minimum achievable rate can be in this case computed with a geometric argument: the goal is to *cover* the n-dimensional balls related to codewords of $\mathcal{C}_2$ with as least as possible balls of average quadratic-distortion D. But the quadratic-distortion of the formers cannot be smaller than σ_z^2 so that the minimum code rate of code $\mathcal{C}_0$, provided that $\sigma_q^2 + D \geq \sigma_z^2$, equals the rate-distortion function of the SCSI problem $R^*(D) = (1/2)\log(\sigma_z^2/D)$ (Wyner & Ziv, 1976).

Despite in principle the decoder/encoder for a CCSI problem can be used as the encoder/decoder of a dual SCSI problem (Pradhan et al., 2003), from a practical point of view CCSI and SCSI are very different. In CCSI source coding is done at the encoder; the code $\mathcal{C}_2$ must be structured such that exhaustive search can be employed in order to perform a closest neighbor search (e.g. be a trellis code). Channel decoding is done at the decoder, and both $\mathcal{C}_0$ and $\mathcal{C}_2$ should be performing channel codes for which a good algorithm approximating maximum likelihood search exist (e.g. turbo codes). In practice, no penalty is introduced in having $\mathcal{C}_2$ not very good from a channel coding perspective (Bennatan et al., 2006).

In SCSI channel decoding is done at the decoder; $\mathcal{C}_2$ should be a performing channel code for which a good algorithm approximating maximum likelihood search exist (e.g. turbo codes). Source coding is done at the encoder, and the code $\mathcal{C}_0 + \mathcal{C}_2$ must be structured such that exhaustive search can be employed in order to perform a closest neighbor search (e.g. be a trellis code). Unfortunately, there are currently no algorithms that perform similarly to a closest neighbor search over good codes for channel coding. Hence, in Section 5 we will actually rely on convolutional codes $\mathcal{C}_2$, paying in this case a penalty with respect to the theoretical bounds.

4. Practical Iterative Algorithms for Lossless Coding

Turbo codes are good practical channel codes, and hence are good tools for lossless (binary) SCSI. In the following we discuss the utilization of *standard* and *ready-available* turbo encoding and decoding algorithms for this problem. Differently from other contributions on this subject, we use the *factor graph*-based approach commonly taken in the LDPC-codes-related literature. For a useful tutorial article on factor graphs and *message-passing* algorithms, the reader is referred to (Kschischang et al., 2001). Under a *unified formulation*, we describe in principle the cases in which syndromes or parity bits are sent to the decoder, over binary or non-binary, lossless or lossy transmission channels, with binary or non-binary side information; more details can be found in (Cappellari & De Giusti, 2008).

4.1 Turbo Codes Review and the Parity-Based Approach

Turbo codes actually include different kinds of codes. In the most common case (*parallel concatenated convolutional codes*) they are *systematic* codes: in correspondence of a sequence of Nk outcomes from X ($\mathbf{x}$) the turbo encoder uses two systematic (n, k) *convolutional* codes to form two sequences of parity bits of $N(n - k) + z_t$ bits each[4] ($\mathbf{p}_0$ and $\mathbf{p}_1$), according to the following algorithm.

1: **function** TRBENC($\mathbf{x}$)
2:     $\mathbf{p}_0 \leftarrow$ GETPARITY$_0$($\mathbf{x}$)
3:     $\mathbf{x}' \leftarrow$ INTERLEAVE($\mathbf{x}$)

---

[4] The additional $z_t \ll N(n - k)$ bits are emitted while terminating the encoding into the zero state (*zero-tailing*).

4:       $\mathbf{p}_1 \leftarrow \text{GETPARITY}_1(\mathbf{x}')$
5:       **return** $[\mathbf{x}, \mathbf{p}_0, \mathbf{p}_1]$
6: **end function**

Once $\mathbf{x}$ and $\mathbf{p}_i$ are sent over a channel, they are received as $\mathbf{r}$ and $\mathbf{r}_i$ respectively, and the turbo decoder aims to maximize $p_{\mathbf{rr}_0\mathbf{r}_1}(\mathbf{x}) \propto l_{\mathbf{rr}_0\mathbf{r}_1}(\mathbf{x})p(\mathbf{x})$.[5] The *maximum likelihood* (ML) decoding procedure is approximated by a message-passing algorithm on the factor graph of $l_{\mathbf{rr}_0\mathbf{r}_1}(\mathbf{x})$. In particular, $l_{\mathbf{rr}_0\mathbf{r}_1}(\mathbf{x})$ factorizes into $l_{\mathbf{r}}(\mathbf{x})l_{\mathbf{r}_0}(\mathbf{x})l_{\mathbf{r}_1}(\mathbf{x})$, and $l_{\mathbf{r}_i}(\mathbf{x}) = \sum_{\mathbf{p}_i} \chi_i(\mathbf{p}_i|\mathbf{x})l_{\mathbf{r}_i}(\mathbf{p}_i)$, where $\mathbf{p}_i$ are the *true* parity sequences and $\chi_i(\mathbf{p}_i|\mathbf{x})$ are the indicator functions that are unitary if and only if $\mathbf{p}_i$ is the parity of $\mathbf{x}$ (according to the i-th convolutional code, comprehensive of the interleaver for $i = 1$). The traditional decoding algorithm operates (on the factor graph of Fig. 5(a)) as follows, where the function $\text{FBA}_i(\cdot)$ computes the APP function relative to the i-th convolutional code, assuming $q(\mathbf{x})$ as the *prior* probability and using the *forward-backward algorithm* (Bahl et al., 1974). If the prior probability $p(\mathbf{x})$ is known, it can be absorbed into $l_{\mathbf{r}}(\mathbf{x})$ in order to implement *maximum a posteriori probability* (MAP) decoding.

1: **function** $\text{TRBDEC}(l_{\mathbf{r}}(\mathbf{x}), l_{\mathbf{r}_0}(\mathbf{p}_0), l_{\mathbf{r}_1}(\mathbf{p}_1), M)$
2:       $m \leftarrow 0, i \leftarrow 1$, initialize $q(\mathbf{x})$ ▷ Initialization
3:       **repeat**   ▷ Turbo loop
4:             $m \leftarrow m + 1, i \leftarrow 1 - i$
5:             $ap^{(m)}(\mathbf{x}) \leftarrow \text{FBA}_i(l_{\mathbf{r}}(\mathbf{x}), l_{\mathbf{r}_i}(\mathbf{p}_i), q(\mathbf{x}))$
6:             $q(\mathbf{x}) \leftarrow ap^{(m)}(\mathbf{x})/[l_{\mathbf{r}}(\mathbf{x})q(\mathbf{x})]$
7:       **until** $m \geq M$
8:       **return** $ap^{(m)}(\mathbf{x})$
9: **end function**

The application of turbo codes to the SCSI problem $X = Y + Z$, with $Z \sim \mathcal{B}(p)$ and $Y \sim \mathcal{B}(1/2)$, is very straightforward. In particular, the parities are sent, and decoding is done by simply invoking $\text{TRBDEC}(l_{\mathbf{y}}(\mathbf{x}), l_{\mathbf{r}_0}(\mathbf{p}_0), l_{\mathbf{r}_1}(\mathbf{p}_1), M)$ (Aaron & Girod, 2002; Garcia-Frias & Zhao, 2001), where $l_{\mathbf{y}}(\mathbf{x}) = p_{Z^{Nk}}(\mathbf{x} - \mathbf{y})$ takes into account for the virtual channel statistics (see Fig. 5(a)). In addition, it is possible to jointly decode and estimate p with no performance loss (Garcia-Frias & Zhao, 2001). Since *puncturing* (i.e. bit removal) can be employed at the encoder before transmission (the decoder can take into account this fact by assuming uniform likelihoods in correspondence of the punctured bits), any rate $0 \leq R \leq \frac{2(n-k)}{k}$ is achievable.

4.2 Syndrome-Based Approach

A turbo code can be seen as a $(N(2n - k) + 2z_t, Nk)$ systematic *block* code whose *generator matrix* is $\mathbf{G} = [\mathbf{I}_{Nk}|\mathbf{P}_0|\mathbf{P}_1]$, where $\mathbf{P}_i$ is the $Nk \times [N(n - k) + z_t]$ *parity formation* matrix corresponding to the i-th convolutional code (comprehensive of the possible interleaver). If puncturing is employed (exclusively on the parity bits), then the equivalent generator matrix is $\mathbf{G}' = [\mathbf{I}_{Nk}|\mathbf{P}'_0|\mathbf{P}'_1]$, where $\mathbf{P}'_i$ is the $Nk \times s_i$ matrix obtained removing from $\mathbf{P}_i$ the columns corresponding to the punctured parity bits. Consequently,

$$\mathbf{H}' = \begin{bmatrix} \mathbf{P}'^T_0 & \mathbf{I}_{s_0} & \mathbf{0}_{s_0 \times s_1} \\ \mathbf{P}'^T_1 & \mathbf{0}_{s_1 \times s_0} & \mathbf{I}_{s_1} \end{bmatrix} \tag{23}$$

---

[5] Given two r.v. A and B, the *likelihood* and *a posteriori probability* (APP) functions will hereafter be denoted by $l_a(b) \triangleq p(a|b)$ and $p_a(b) \triangleq p(b|a)$ respectively.

(a) parity-based approach (with $p(\mathbf{r}|\mathbf{x}) = p(\mathbf{y}|\mathbf{x})$)
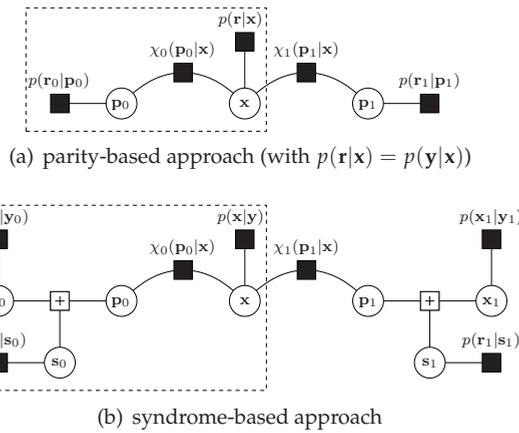

(b) syndrome-based approach

Fig. 5. Factor graphs representing the APP functions in the SCSI problem using turbo codes.

is a parity-check matrix of the turbo code. In correspondence of a sequence of $Nk + s_0 + s_1$ outcomes from X (partitioned into the three sub-sequences $\mathbf{x}$, $\mathbf{x}_0$, and $\mathbf{x}_1$ of length Nk, s_0, and s_1 respectively), the syndrome is obtained according to the following algorithm. Eventually, any rate $0 \le R \le \frac{2(n-k)}{2n-k} < 1$ bit/sample is achievable; for example, if $(2,1)$ constituent codes are employed, $0 \le R \le 2/3$ bit/sample.

```
1: function SYNENC(x, x₀, x₁)
2:     [x, p₀, p₁] ← TRBENC(x)
3:     for i ← 0, 1 do
4:         p ← PUNCTUREᵢ(pᵢ)
5:         sᵢ ← p + xᵢ
6:     end for
7:     return [s₀, s₁]
8: end function
```

We now directly tackle the problem of optimal MAP syndrome decoding, assuming that $\mathbf{s}_i$ are sent over a general channel and are received as $\mathbf{r}_i$. In particular, the decoding algorithm is derived from examining the factor graph of the APP function $p_{\mathbf{y}\mathbf{y}_0\mathbf{y}_1\mathbf{r}_0\mathbf{r}_1}(\mathbf{x}\mathbf{x}_0\mathbf{x}_1) \propto p_{\mathbf{y}\mathbf{y}_0\mathbf{y}_1}(\mathbf{x}\mathbf{x}_0\mathbf{x}_1)l_{\mathbf{r}_0\mathbf{r}_1}(\mathbf{x}\mathbf{x}_0\mathbf{x}_1)$. But $p_{\mathbf{y}\mathbf{y}_0\mathbf{y}_1}(\mathbf{x}\mathbf{x}_0\mathbf{x}_1)$ factorizes into $p_{\mathbf{y}}(\mathbf{x})p_{\mathbf{y}_0}(\mathbf{x}_0)p_{\mathbf{y}_1}(\mathbf{x}_1)$; similarly $l_{\mathbf{r}_0\mathbf{r}_1}(\mathbf{x}\mathbf{x}_0\mathbf{x}_1) = l_{\mathbf{r}_0}(\mathbf{x}\mathbf{x}_0)l_{\mathbf{r}_1}(\mathbf{x}\mathbf{x}_1)$, and $l_{\mathbf{r}_i}(\mathbf{x}\mathbf{x}_i) = \sum_{\mathbf{s}_i} l_{\mathbf{r}_i}(\mathbf{s}_i) \sum_{\mathbf{p}_i} \chi_i(\mathbf{p}_i|\mathbf{x})\chi_{\{\mathbf{p}_i+\mathbf{x}_i=\mathbf{s}_i\}}$, where $\mathbf{s}_i$ and $\mathbf{p}_i$ are the *true* syndrome sequences and the parity sequences from which they are computed respectively, and $\chi_{\{\mathbf{p}_i+\mathbf{x}_i=\mathbf{s}_i\}}$ is the indicator function of the condition in brackets. The corresponding factor graph of the APP function is shown in Fig. 5(b).

This factor graph is an extension without additional cycles of the factor graph relative to the parity-based approach. MAP decoding can be achieved by reusing the turbo decoding algorithm presented above; in particular, it is only necessary to form the correct input likelihoods

to the function TRBDEC$(\cdot)$ and post-process the output APP function $ap(\mathbf{x})$, for example using the following algorithm (*hard syndrome decoding*).[6]

1: **function** HSYNDEC$(p_{\mathbf{y}}(\mathbf{x}), p_{\mathbf{y}_0}(\mathbf{x}_0), p_{\mathbf{y}_1}(\mathbf{x}_1), l_{\mathbf{r}_0}(\mathbf{s}_0), l_{\mathbf{r}_1}(\mathbf{s}_1), M)$
2:     **for** $i \leftarrow 0, 1$ **do** $\triangleright$ Pre-processing
3:         $l_i(\mathbf{p}_i) \leftarrow \sum_{\mathbf{s}_i} l_{\mathbf{r}_i}(\mathbf{s}_i) \sum_{\mathbf{x}_i} p_{\mathbf{y}_i}(\mathbf{x}_i) \chi_{\{\mathbf{p}_i + \mathbf{x}_i = \mathbf{s}_i\}}$
4:     **end for**
5:     $ap(\mathbf{x}) \leftarrow$ TRBDEC$(p_{\mathbf{y}}(\mathbf{x}), l_0(\mathbf{p}_0), l_1(\mathbf{p}_1), M)$
6:     $\hat{\mathbf{x}} \leftarrow \arg\max_{\mathbf{x}} ap(\mathbf{x})$
7:     $[\hat{\mathbf{x}}, \hat{\mathbf{p}}_0, \hat{\mathbf{p}}_1] \leftarrow$ TRBENC$(\hat{\mathbf{x}})$
8:     **for** $i \leftarrow 0, 1$ **do** $\triangleright$ Post-processing
9:         $\hat{\mathbf{x}}_i \leftarrow \arg\max_{\mathbf{x}_i} p_{\mathbf{y}_i}(\mathbf{x}_i) l_{\mathbf{r}_i}(\hat{\mathbf{p}}_i + \mathbf{x}_i)$
10:    **end for**
11:    **return** $[\hat{\mathbf{x}}, \hat{\mathbf{x}}_0, \hat{\mathbf{x}}_1]$
12: **end function**

Using the *log-likelihood ratio* and the *log-APP ratio* defined as

$$l_a \triangleq \log \frac{l_a(1)}{l_a(0)} \, , \qquad p_a \triangleq \log \frac{p_a(1)}{p_a(0)} \, , \tag{24}$$

the j-th factor computed by the pre-processing operation (line 3) is $l_{i,j} = l_{\mathbf{r}_{i,j}} * p_{\mathbf{y}_{i,j}}$, where the *log-likelihood ratio convolution* operator is defined as $l_1 * l_2 \triangleq \log \frac{e^{l_1} + e^{l_2}}{1 + e^{l_1 + l_2}}$. Post-processing (line 9) yields $\hat{\mathbf{x}}_{i,j} = \mathbf{r}_{i,j} + \hat{\mathbf{p}}_{i,j}$; maximization translates into thresholding.

The thresholding and parity formation operations permit the reutilization of the traditional turbo algorithms. Nevertheless, they prevent the computation of the correct messages across the nodes $\mathbf{p}_i$, so the hard syndrome decoding algorithm is not optimal. But if a *full* turbo decoding function FTRBDEC$(\cdot)$ is used that outputs the APP functions $ap(\mathbf{p}_i)$ for the parity bits as well, the post-processing can be improved as follows (*soft syndrome decoding*).

1: **function** SSYNDEC$(p_{\mathbf{y}}(\mathbf{x}), p_{\mathbf{y}_0}(\mathbf{x}_0), p_{\mathbf{y}_1}(\mathbf{x}_1), l_{\mathbf{r}_0}(\mathbf{s}_0), l_{\mathbf{r}_1}(\mathbf{s}_1), M)$
2:     **for** $i \leftarrow 0, 1$ **do** $\triangleright$ Pre-processing
3:         $l_i(\mathbf{p}_i) \leftarrow \sum_{\mathbf{s}_i} l_{\mathbf{r}_i}(\mathbf{s}_i) \sum_{\mathbf{x}_i} p_{\mathbf{y}_i}(\mathbf{x}_i) \chi_{\{\mathbf{p}_i + \mathbf{x}_i = \mathbf{s}_i\}}$
4:     **end for**
5:     $[ap(\mathbf{x}), ap(\mathbf{p}_0), ap(\mathbf{p}_1)] \leftarrow$ FTRBDEC$(p_{\mathbf{y}}(\mathbf{x}), l_0(\mathbf{p}_0), l_1(\mathbf{p}_1), M)$
6:     **for** $i \leftarrow 0, 1$ **do** $\triangleright$ Post-processing
7:         $ap'(\mathbf{x}_i) \leftarrow \sum_{\mathbf{s}_i} l_{\mathbf{r}_i}(\mathbf{s}_i) \sum_{\mathbf{p}_i} \frac{ap(\mathbf{p}_i)}{l_i(\mathbf{p}_i)} \chi_{\{\mathbf{p}_i + \mathbf{x}_i = \mathbf{s}_i\}}$
8:         $ap(\mathbf{x}_i) \leftarrow p_{\mathbf{y}_i}(\mathbf{x}_i) ap'(\mathbf{x}_i)$
9:     **end for**
10:    **return** $[ap(\mathbf{x}), ap(\mathbf{x}_0), ap(\mathbf{x}_1)]$
11: **end function**

Now, in order to estimate $\mathbf{x}_{i,j}$ thresholding will be applied to $ap_{\mathbf{x}_{i,j}} = p_{\mathbf{y}_{i,j}} + l_{\mathbf{r}_{i,j}} * (ap_{\mathbf{p}_{i,j}} - l_{i,j})$, which is certainly more accurate than before. If the transmission channel is error-free

---

[6] The operations in the pre- and post-processing (that do not involve the punctured parity bits) are written in the most general fashion, but in practice they are *symbol-wise* operations between marginal functions.

(i.e. $|l_{\mathbf{r}_{i,j}}| = \infty$) it is easy to show that $ap_{\mathbf{x}_{i,j}}$ equals $ap_{\mathbf{p}_{i,j}}$ or $-ap_{\mathbf{p}_{i,j}}$ if $\mathbf{r}_{i,j}$ is 0 or 1 respectively, because in either case the contributions of $p_{\mathbf{y}_{i,j}}$ and $l_{i,j}$ cancel out; then, $\mathbf{x}_{i,j}$ can be again estimated by $\hat{\mathbf{x}}_{i,j} = \mathbf{r}_{i,j} + \hat{\mathbf{p}}_{i,j}$, where $\hat{\mathbf{p}}_{i,j}$ is the estimate of the corresponding parity bit obtained thresholding $ap_{\mathbf{p}_{i,j}}$ rather than the one obtained invoking the function TRBENC($\hat{\mathbf{x}}$). On the contrary, if the TC is not error-free $|l_{\mathbf{r}_{i,j}}| < \infty$, and hence $ap_{\mathbf{x}_{i,j}}$ must be actually computed in order to correctly estimate $\mathbf{x}_{i,j}$.

4.3 Experimental Results and Comparisons

A virtual channel has been simulated with $Y \sim \mathcal{B}(1/2)$ and $Z \sim \mathcal{B}(p)$, for different values of p; error-free transmission has been considered for comparison purposes with previous literature on this subject. Both parity- and syndrome-based approaches have been simulated. The turbo code uses two identical $(n, k) = (2, 1)$, 16-state, systematic constituent codes with generator matrix $\mathbf{G}(D) = \left[1 \ \frac{1+D+D^2+D^4}{1+D^3+D^4} \right]$; $\lfloor LR \rfloor$ parity/syndrome bits are sent in correspondence to each data *frame* of $L = 2^{16} = 65536$ samples. Puncturing is performed such that rates R of $2/3$ or $1/2$ bit/samples are achieved. $2^{13} = 8192$ frames have been generated for each p, such that the average *bit error ratio* (BER) is eventually estimated over $2^{29} \simeq 5 \cdot 10^8$ bits. The decoding routines TRBDEC and SSYNDEC are set for a number of runs of the FBA algorithm M equal to 40.

Comparisons are given in Fig. 6, in which the BER is shown as a function of $H(p)$. When $R = 2/3$, the proposed method 65536-SSYNDEC outperforms the coding performance of the "SF+ISF" method given in (Tu et al., 2005). Despite the different syndrome formation procedure used in the latter (which does not rely on a standard turbo encoding engine), these two methods are very similar in the way they work. Hence, it is reasonable to think that the different performance is the result of better coding parameters (i.e. frame length, convolutional code, interleaver and puncturer). Despite the very large interleaver length, the "Syn. trellis" method proposed in (Roumy et al., 2007) has very poor performance, which is even worse than the performance of the parity-based method 65536-TRBDEC. When $R = 1/2$, the proposed method 65536-SSYNDEC has again a good performance, which are surpassed only by the LDPC-based systems reported in (Liveris et al., 2002) (which employ a longer frame size) and by the "P&C trellis" method proposed in (Liveris et al., 2003), which makes use of longer frames and of different 16-state constituent codes (specifically tailored for heavy data puncturing). Again, despite its smart formulation and very long frame size, the "Syn. trellis" method (Roumy et al., 2007) has very poor performance.

5. Continuous-Valued Syndromes for Lossy Coding

Traditional (discrete) syndromes of a linear code are good for lossless SCSI. In this section we discuss a coding method based on *continuous-valued* syndromes of a lattice for lossy SCSI of continuous sources with quadratic distortion (Cappellari & Mian, 2006b). This method embodies the superposition coding approach described in Section 3.2.1.

Consider a lattice $\Lambda \subset \mathbb{R}^n$. Being a subgroup of $\mathbb{R}^n$, it induces the partition of $\mathbb{R}^n$ into the cosets of $\mathbb{R}^n / \Lambda$. Each coset is uniquely identified by one of its elements; in particular we assume as *coset leader* of $L \in \mathbb{R}^n / \Lambda$ the element $l(L) \triangleq \arg\min_{\lambda \in L} d(\lambda, 0)$. We call *continuous-valued syndrome* (CVS) of $x^n \in \mathbb{R}^n$, relative to Λ, the element $s_\Lambda(x^n) \triangleq l(L)$ such that $x^n \in L$. If we define the *quantizer* $Q_\Lambda(x^n) \triangleq \arg\min_{\lambda \in \Lambda} d(x^n, \lambda)$ (with the further condition $\lambda + s_\Lambda(x^n) = x^n$ in case of ambiguity), the CVS satisfies $s_\Lambda(x^n) = x^n - Q_\Lambda(x^n)$.
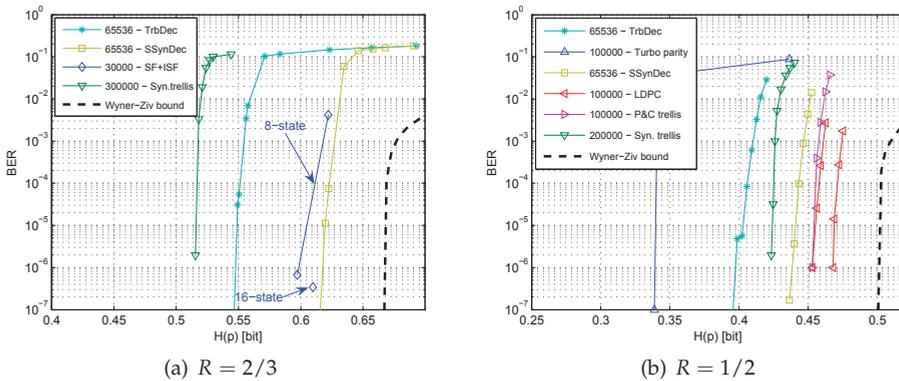
(a) $R = 2/3$

(b) $R = 1/2$

Fig. 6.   Comparison between different SCSI methods.   The label "SF+ISF" refers to the syndrome-based method in (Tu et al., 2005) (results for two different convolutional codes are shown); the label "Syn. trellis" refers to the syndrome-based method in (Roumy et al., 2007), where 16-state constituent codes are employed. The label "Turbo parity" refers to the parity-based method in (Aaron & Girod, 2002), that uses two $(5,4)$ 16-state constituent codes. The label "LDPC" refers to the syndrome-based method in (Liveris et al., 2002) (results relative to two irregular LDPC codes are shown); the label "P&C trellis" refers to the syndrome-based method in (Liveris et al., 2003) that uses 16-state constituent codes. The frame length is reported too.

We consider the additive virtual channel $X = Y + Z$. If $P[Q_\Lambda(Z^n) \neq 0]$ is asymptotically negligible (i.e. the realizations of Z^n lie in the *fundamental Voronoi region* of Λ), then $s_\Lambda(x^n)$ permits near lossless reconstruction at the decoder; in fact

$$x^n = y^n + z^n \stackrel{(1)}{=} y^n + s_\Lambda(z^n) \stackrel{(2)}{=} y^n + s_\Lambda\left(s_\Lambda(x^n) + s_\Lambda(-y^n)\right) , \tag{25}$$

where $^{(1)}$ holds with high probability and $^{(2)}$ follows from linear properties of the CVS. In practice, the reconstruction from $s_\Lambda(x^n)$ and y^n is obtained by a single quantization as

$$x^n = s_\Lambda(x^n) - Q_\Lambda\left(s_\Lambda(x^n) - y^n\right) . \tag{26}$$

Of course, we would need a channel with infinite capacity in order to transmit the CVS to the decoder. Hence, we assume that a quantized version $\hat{s}_{\beta\Lambda}(x^n)$ of the syndrome $s_{\beta\Lambda}(x^n)$, such that $S_{\beta\Lambda} = \hat{S}_{\beta\Lambda} - Q$ with Q independent from $\hat{S}_{\beta\Lambda}$, is actually transmitted by the encoder.[7] For $\beta \geq 1$, the realizations of Z^n and, reasonably, the ones of the *error* Q^n, lie in the fundamental Voronoi region of $\beta\Lambda$. The reconstruction, according to (25), satisfies

$$\tilde{x}^n = y^n + s_{\beta\Lambda}\left(s_{\beta\Lambda}(x^n) + s_{\beta\Lambda}(q^n) + s_{\beta\Lambda}(-y^n)\right) = y^n + s_{\beta\Lambda}\left(z^n + q^n\right) \tag{27}$$

$$= x^n + \left(q^n - Q_{\beta\Lambda}\left(z^n + q^n\right)\right) \triangleq x^n + (q^n + q^n_{ol}) , \tag{28}$$

---

[7] If the Voronoi regions of $\beta\Lambda$ are asymptotically spherical, $\hat{S}_{\beta\Lambda}$ and Q exist that are approximately Gaussian.
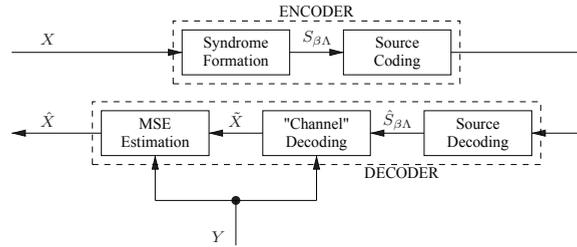
Fig. 7. Wyner-Ziv coding using continuous-valued syndromes.

where Q_{ol} is the *overload error*. If $Q_t \triangleq Q + Q_{ol}$ is independent from Y and Z (and the mean of all random variables is zero), $Y \to X \to \tilde{X}$ forms a Markov chain, such that the minimum mean square error *linear* estimate for X computed at the decoder is eventually given by

$$\hat{X} = \frac{\sigma_z^2}{\sigma_z^2 + \sigma_{q_t}^2} \tilde{X} + \frac{\sigma_{q_t}^2}{\sigma_z^2 + \sigma_{q_t}^2} Y \,, \tag{29}$$

and the achieved distortion D satisfies $1/D = 1/\sigma_z^2 + 1/\sigma_{q_t}^2$, where $\sigma_z^2 \triangleq E[Z^2]$ and $\sigma_{q_t}^2 \triangleq E[Q_t^2]$. This coding scheme is shown in Fig. 7.

For a fixed transmission rate R the best possible syndrome is sent, and the parameter β is experimentally tuned in order to minimize the variance of D. In theory, in the Gaussian case ($Z \sim \mathcal{N}(0, \sigma_z^2)$) with Λ being a good channel code for the virtual AWGN channel between Y and X, $\beta = 1/\alpha = 1/\sqrt{1 - D^*(R)/\sigma_z^2}$ guarantees that the rate-distortion function $R^*(D)$ is achieved ($D^*(R) = \sigma_z^2 2^{-2R}$ is its inverse). In fact, in this case, (i) since the syndrome distributes uniformly over the fundamental Voronoi region of $\beta\Lambda$ and Λ is a good channel code for the noise Z, $S_{\beta\Lambda} \sim \mathcal{N}(0, \beta^2 \sigma_z^2)$, (ii) the minimum distortion in sending the syndrome at rate R is such that $E[Q^2] = \beta^2 \sigma_z^2 2^{-2R} = \beta^2 D^*(R)$, (iii) since $\sigma_z^2 + E[Q^2] = \beta^2 \sigma_z^2$ and $\beta\Lambda$ is a good channel code for an AWGN of power $\beta^2 \sigma_z^2$, there is no overload error, i.e. $\sigma_{q_t}^2 = E[Q^2]$, (iv) $D = \sigma_z^2 E[Q^2]/(\sigma_z^2 + E[Q^2]) = E[Q^2]/\beta^2 = D^*(R)$.

A superposition of two codes is indeed used where the coarse code is the lattice code $\beta\Lambda$ and the additive code is the source code used for quantization of $S_{\beta\Lambda}$. The encoder is dual to the decoder of a *multiple access channel* problem that operates by *interference cancellation*: first, the codeword of the code with the lower code rate (the coarse one) is computed; then, this is subtracted from the source and the codeword to be transmitted to the decoder is formed according to the additive code. In practice, since we must be able to conduct an exact closest neighbor search on the coarse code, it cannot be a very performing channel code, i.e. we must cope in general with an overload error Q_{ol} (which corresponds to having a decoder error, as defined in Section 3.2.1). The parameter β permits to balance the contributions of the two errors. The higher is β, the lower is the overload error; but, for fixed transmission rate R, the higher is β, the higher is the variance of the *granular* error Q.

5.1 Experimental Results and Comparisons

We simulated the AWGN channel $X = Y + Z$ with a Gaussian input $Y \sim \mathcal{N}(0, \sigma_y^2 = 1)$ and a noise Z with various variances σ_z^2. We employ *trellis-coded quantization* (TCQ) based on the partition $a\mathbb{Z}/4a\mathbb{Z}$ for syndrome formation (Marcellin & Fisher, 1990), which defines the lattice
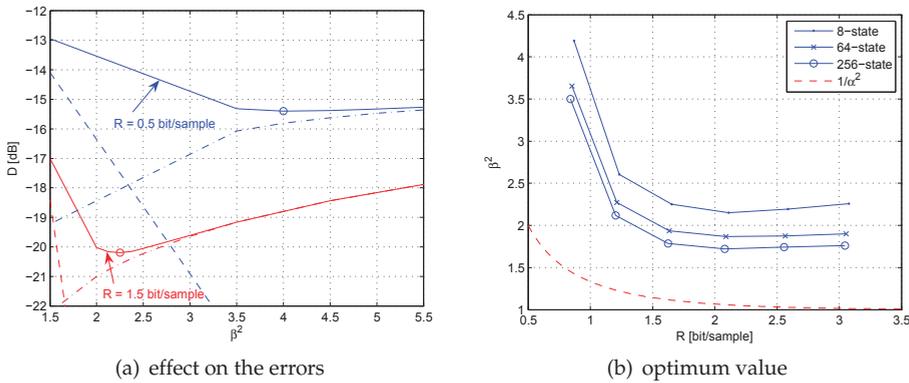
(a) effect on the errors   (b) optimum value

Fig. 8. Optimization of the volumetric parameter β^2.

Λ; the length of each input data frame is $n = 1000$; a is tuned such that the experimental second moment per dimension of the Voronoi regions of Λ equals σ_z^2. Any suitable source coding algorithm can be used to code the n-dimensional CVS; again, we employ TCQ, based on the partition $b\mathbb{Z}/4b\mathbb{Z}$ and on 8-state trellises. For each rate R, b is tuned such that the normalized volume of the Voronoi regions induced by this quantizer is 2^{2R} times less than the corresponding parameter of $\beta\Lambda$. Actually, in the following, R denotes the average entropy for describing the quantized syndrome, measured with the 2-supersets context-based method proposed in (Marcellin, 1994).

The effect of β^2 on the variances of the granular and of the overload error is shown in Fig. 8(a), for $\sigma_y^2/\sigma_z^2 = 14.0$ dB and at two different target rates. At low β^2, the total error (solid curves) is approximated by the overload error (dashed curves); at high β^2, the total error is approximated by the granular error (dot-dashed curves). The circles indicate the optimum value of β^2 and the corresponding distortion. The optimum β^2 does not depend on the variance ratio σ_y^2/σ_z^2 but only on the coding rate R of the system. In particular, this optimum is shown in Fig. 8(b) and it is decreasing with the rate. By increasing the number of states of the trellis, Λ gets closer to a good channel code and consequently β^2 decreases; for comparison, the figure also shows $1/\alpha^2$, i.e. the lower bound for β^2.

The experimental performance loss with respect to the *distortion-rate* function $D^*(R)$ is shown in Fig. 9(a). More precisely, the measurements are relative to a CVS-based system in which the TCQ used for syndrome quantization has been optimized with an algorithm adapted from (Chou et al., 1989). Similarly to the optimum value of β, the performance loss does not depend on σ_y^2/σ_z^2; for each rate, the value shown is the average over the various values of σ_z^2 (σ_y^2/σ_z^2 is in the range $9 \div 19$ dB). The error bars show the average of the 95 % confidence intervals; for each value of σ_z^2 the confidence interval is estimated over 5000 independent simulations. The experiments show that the performance of the proposed system is within $3 \div 4$ dB from the theoretical bound at rates between 0.5 and 3.0 bit/sample. A comparison is given in Fig. 9(b) against another practical SCSI system, namely the "DISCUS" system (Pradhan & Ramchandran, 2003). By simply choosing the right scaling factor a, tuned with respect to σ_z^2, the proposed system adapts to any correlation and gives the same performance loss, while the "DISCUS" system should be optimized for different correlations. This would not be an

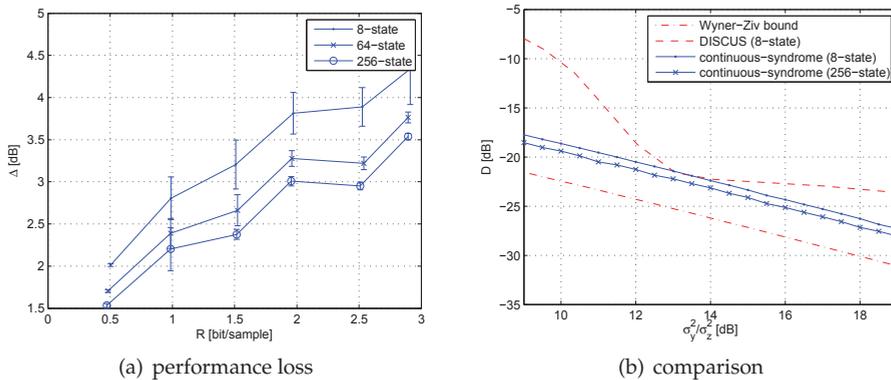(a) performance loss

(b) comparison

Fig. 9. Experimental performance of CVS-based SCSI.

easy task since it would involve the redesign of a source and of a channel code. Moreover, while only integer rates can be achieved by the "DISCUS" system, any rate can be achieved by SCSI based on CVS. This can be simply obtained by choosing the right value of b in case of syndrome coding with TCQ or using another ad-hoc source coding method for transmitting $S_{\beta\Lambda}$.

The proposed coding method turns out to be competitive with the SCSI methods in which the channel code component is a convolutional code; in addition, it allows for easy adaptation to the virtual channel statistics and to the desired transmission rate. More involved decoding algorithms for the actual case in which the virtual channel is not exactly known, and applications of this coding scheme in the video coding scenario have been proposed too and are briefly discussed in the following.

5.2 Iterative Algorithms for Unknown Virtual Channels

The factor graph approach used in Section 4 for lossless SCSI turns out to be useful for optimized CVS decoding too, in place of the simple operation given in (26). In particular, in (Cappellari, 2008) a factor graph-based decoding method is discussed for the case where the virtual channel statistics is *time-varying* and not exactly known at the encoder. More precisely, with the hypothesis of a negligible overload error such that $\tilde{X} \approx Y + \tilde{Z}$, where $\tilde{Z} \triangleq Z + Q$ is independent from Y, a doubly stochastic *hidden Markov model* (HMM) (Rabiner, 1989) is assumed for $\tilde{Z}$. The model has L states and the distribution corresponding to the j-th state is the *generalized Gaussian distribution* (GGD) $\mathcal{G}_\alpha(\mu_j, \sigma_j^2)$. The simulated n-dimensional realizations of Z^n are identified within another HMM with possibly different number of states and state variances in order to simulate a partial knowledge regarding the virtual channel; this information is used by the encoder and transmitted to the decoder (the required bit-rate is taken into account) for syndrome formation and decoding, respectively.

Since $\hat{S} = S + Q$ (we will omit subscripts for clarity) is with good approximation the syndrome corresponding to $\tilde{X} = X + Q$, $Y \to \tilde{X} \to \hat{S}$ forms a Markov chain, and optimal MAP decoding amounts to maximizing

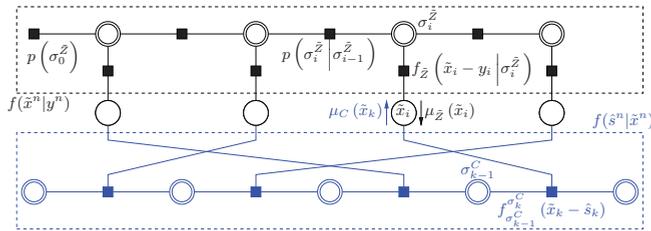$$f(\tilde{x}^n | y^n, \hat{s}^n) \propto f(\tilde{x}^n | y^n) f(\hat{s}^n | \tilde{x}^n) \,, \tag{30}$$

Fig. 10. Factor graph-based CVS decoding ($n = 4$).

where the terms $f(\tilde{x}^n|y^n) = f_{\tilde{Z}}(\tilde{x}^n - y^n)$ and $f(\hat{s}^n|\tilde{x}^n)$ take account of the virtual channel structure and of the syndrome formation algorithm, respectively. Once we define the *hidden state variables* $\sigma_i^{\tilde{Z}}$, the *state transition probabilities* $p\left(\sigma_i^{\tilde{Z}} \middle| \sigma_{i-1}^{\tilde{Z}}\right)$, and the *state probability densities*

$$f_{\tilde{Z}}\left(a \middle| j\right) = \frac{\alpha/2}{\sqrt{2\Gamma(1/\alpha)^2 \gamma_j^2}} \exp\left\{-\left(\frac{|a-\mu_j|}{\sqrt{2\gamma_j^2}}\right)^{\alpha}\right\}, \text{ with } \gamma_j^2 \triangleq \sigma_j^2 \Gamma\left(1/\alpha\right)/2\Gamma\left(3/\alpha\right), f_{\tilde{Z}} \text{ is found by}$$

marginalizing

$$f_{\tilde{Z}}\left(z^n; \sigma^{\tilde{Z}}\right) = p\left(\sigma_0^{\tilde{Z}}\right) f_{\tilde{Z}}\left(z_0 \middle| \sigma_0^{\tilde{Z}}\right) \prod_{i=1}^{m-1} p\left(\sigma_i^{\tilde{Z}} \middle| \sigma_{i-1}^{\tilde{Z}}\right) f_{\tilde{Z}}\left(z_i \middle| \sigma_i^{\tilde{Z}}\right) . \tag{31}$$

The (TCQ-based) syndrome formation is instead a deterministic transformation, i.e. $f(\hat{s}^n|\tilde{x}^n)$ is a Dirac's delta function that, given $\tilde{x}^n$, reveals its syndrome. Equivalently, it is a delta function that reveals the event $\{\tilde{x}^n - \hat{s}^n \in \beta\Lambda\}$; by introducing the *trellis state variables* σ_k^C, $f(\hat{s}^n|\tilde{x}^n)$ is found marginalizing

$$f\left(\hat{s}^n; \sigma^C \middle| \tilde{x}^n\right) = \prod_{k=0}^{m-1} \chi_{\sigma_{k-1}^C}^{\sigma_k^C} \sum_{b \in \mathcal{B}_{\sigma_{k-1}^C}^{\sigma_k^C}} \delta\left(\tilde{x}_k - \hat{s}_k - b\right) , \tag{32}$$

where χ_j^l and $\mathcal{B}_j^l$ are the indicator function and the set of reconstruction values relative to the transition from state j to l, respectively.

In order to increase the independence between the information about $\tilde{X}$ brought by Y and by $\hat{S}$, the syndrome formation algorithm is operated on a randomly scrambled version of x^n, such that the actual factor graph used in CVS decoding is similar to the one sketched in Fig. 10. At the m-th iteration, *model messages* $\mu_{\tilde{Z}}^{(m)}(\tilde{x}_i)$ and *code messages* $\mu_C^{(m)}(\tilde{x}_k)$ are exchanged between the upper and the lower part of the factor-graph and viceversa, until they converge to a fixed value. The final estimate $\hat{x}^n$ for x^n is given by

$$\hat{x}_i = \arg\max_{\tilde{x}_i} f(\tilde{x}_i|y^n) f(\hat{s}^n|\tilde{x}_i) . \tag{33}$$

We report some experimental results for the case where $Y \sim \mathcal{N}(0, \sigma_y^2 = 1)$ and Z is a 3-state Gaussian Markov process ($\alpha = 2$). The distortion-rate function obtained with traditional ML
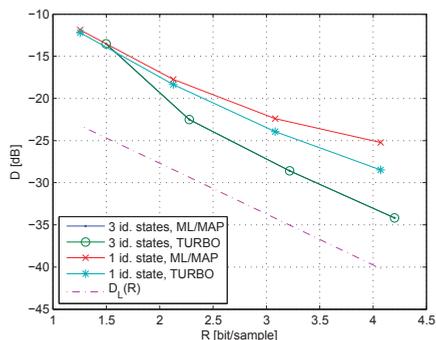
Fig. 11. Experimental results for factor graph-based CVS decoding; $D_L(R)$ is an estimate of the theoretical distortion-rate function of the considered SCSI problem.

decoding, with MAP decoding (i.e. considering only one iteration in the decoding process), and with TURBO decoding (i.e. at convergence) are measured. The syndrome is uniformly quantized and entropy coded for transmission at the decoder; 1000 sequences of $n = 1000$ samples each are generated. If the encoder has knowledge about the virtual channel statistics (i.e. the same model is artificially used for hidden state identification[8]), the TURBO decoding algorithm does not perform any better than the MAP decoding algorithm, that in turn was shown to perform like the ML decoding (at least in the Gaussian case). If a single-state HMM is used for hidden state identification (i.e. the encoder only knows the average variance of Z), the MAP decoding performs again like the ML decoding, but TURBO decoding leads to about one order of magnitude less reconstruction errors, which translate into an up to 3 dB decrease of the mean error variance (see Fig. 11) at bit-rates $R = 3 \div 4$ bit/sample.

5.3 Video Coding Applications

Traditional video coding standards, e.g. H.264/AVC (ITU-T & ISO/IEC: JTC1/SC29/WG11, 2007), are based on *predictive* coding for exploiting the high temporal correlation between adjacent frames. This implies that (i) the algorithms used during encoding are computationally heavy with respect to the ones used at the decoder, and (ii) the coded representation is very sensible to possible packet losses on the transmission channel. To alleviate these problems, in order to permit effective video encoding and transmission on wireless and battery-operated devices, several research groups have recently explored SCSI-based methods for video compression. A review on these methods can be found in (Girod et al., 2005).

In practice, a frame is encoded assuming that some adjacent frames are already available as SI at the decoder. Hence, (i) the encoding algorithm is more light since it does not exploit the *inter-frame* correlation, and (ii) packet losses are not so bad as long as several adjacent frames are stored at the decoder that can be used as SI.

The CVS method has been applied to video coding in (Cappellari, 2007; Cappellari & Mian, 2006a). In particular, it has been tested in both the *discrete cosine transform* (DCT) and the *discrete wavelet transform* (DWT) domain. In both cases, every other frame is sent as an *intra-frame* (without referencing any adjacent frame); the remaining ones are sent as inter-frames. The decoder performs *motion compensated interpolation* for each couple of consecutive intra-frames

---

[8] There may be still uncertainties about the *actual* hidden states.
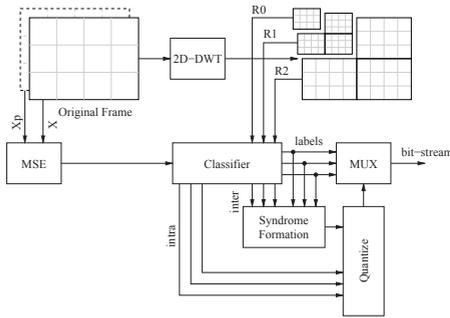
(a) DWT-domain encoder
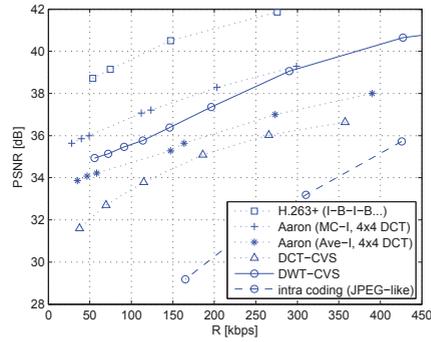
(b) performance comparison

Fig. 12. Scheme of the proposed DWT-domain CVS-based video coding method and performance of different video codecs, averaged on the first 100 frames of the sequence foreman (QCIF resolution) at 25 frames per second (data relative to intra frames are not taken into account).

in order to construct the SI for decoding the inter-frame between them. In the DCT case, inter-frames are partitioned into 8×8 blocks of *pixels* and the DCT coefficients corresponding to the lower spatial frequencies of each block form the signal to be coded relying on co-positioned coefficients of the SI; the remaining coefficients are sent in intra mode. In the DWT case (see Fig. 12(a)), the DWT coefficients are classified into the classes intra, inter, and *skip*, for coefficients that are expected to have little, medium and high correlation with the co-positioned coefficients of the SI, respectively. Intra coefficients are sent in intra mode, no information is sent for the skip coefficients (the corresponding SI is taken directly as reconstruction), and the remaining ones form the signal to be coded using SI.

Classification and correlation estimation are in both cases performed using the block-based MSE between the inter-frame to be coded and the previous one. These estimates are used for proper TCQ-based syndrome formation and decoding. Syndromes are quantized with an *embedded* uniform quantizer, such that *quality scalable* reconstruction is achieved at the decoder. The typical performance of the proposed coders is shown in Fig. 12(b) and compared with results from (Aaron et al., 2004); the average *peak signal-to-noise ratio* (PSNR) is reported as a function of the average transmission rate. The DWT-CVS coder outperforms both the DCT-CVS coder and one of the coders (Ave-I) from (Aaron et al., 2004), while being very close to the other solution (MC-I). The superiority of this (turbo code-based) method is probably due to the utilization of a feedback channel between decoder and encoder for estimation of the actual SI-source correlation. This solution is very good for this purpose, but also highly unpractical with respect to correlation estimation at the encoder only.

6. Conclusion and Future Research

In this chapter, we presented the SCSI problem and discussed several practical solutions. Our main contribution is the coding method based on continuous-valued syndromes, which is an embodiment of the theoretically optimal superposition coding approach. We showed that this coding method is very practical due to the separation between channel and source coding.

In particular, we showed that it can be quite performing in the actual scenarios, where for example the actual source-SI correlation is not exactly known and/or very complex, as in the case of video coding.

In practice, despite its optimality, the superposition coding approach for SCSI suffers some performance loss because there are currently no efficient algorithms for source coding over codes with a "random structure". Indeed, a good channel code should be used as coarse code, but all good channel codes have this "random structure" (e.g. turbo codes rely on random interleaving). Convolutional codes are less performing but are still the best ones for which a good closest neighbor search algorithm exists. In the future, it may be possible that message-passing algorithms will be developed that permit quantization over *sparse* codes that are good for channel coding; some effort in this direction is discussed in (Ciliberti et al., 2005; Martinian & Yedidia, 2003).

In the near future, we plan to be more concerned with the problem of correlation estimation in the actual SCSI scenarios. In fact, all SCSI schemes proposed in literature are usually investigated under the hypothesis of a toy virtual channel $Y = X + Z$ in which the statistics of Z is known at the encoder and at the decoder. But in practice the source-SI correlation is not known and is more complex, so that by using these simple assumptions we incur into some performance degradation. For example, the performance of the SCSI-based video coders is still under the one of the traditional coders, at least in the scenarios with no losses on the transmission channel. We hope that our investigative efforts into statistical model aided decoding could be eventually used towards improving the efficacy of SCSI-based coding not only in video coding applications, but also in several other practical cases.

7. References

Aaron, A. & Girod, B. (2002). Compression with side information using turbo codes, *Proc. of IEEE Data Compression Conf.*, pp. 252–261.

Aaron, A., Rane, S., Setton, E. & Girod, B. (2004). Transform-domain Wyner-Ziv codec for video, *Proc. Visual Commun. and Image Proc. (VCIP-2004)*, San Jose, CA, USA.

Bahl, L., Cocke, J., Jelinek, F. & Raviv, J. (1974). Optimal decoding of linear codes for minimizing symbol error rate, *IEEE Trans. Inf. Theory* **20**(2): 284–287.

Bennatan, A., Burshtein, D., Caire, G. & Shamai, S. (2006). Superposition coding for side-information channels, *IEEE Trans. Inf. Theory* **52**(5): 1872–1889.

Berrou, C. & Glavieux, A. (1996). Near optimum error correcting coding and decoding: turbo-codes, *IEEE Trans. Commun.* **44**(10): 1261–1271.

Cappellari, L. (2007). Wavelet-domain distributed video coding based on continuous-valued syndromes, *Proc. of European Signal Process. Conf. (EUSIPCO)*, Poznań, Poland, pp. 1422–1426.

Cappellari, L. (2008). Statistical model-aided decoding of continuous-valued syndromes for source coding with side information, *Proc. of European Signal Process. Conf. (EUSIPCO)*, Lausanne, Switzerland.

Cappellari, L. (2009). On superposition coding for the Wyner-Ziv problem.
    URL: *http://arxiv.org/pdf/0904.0879*

Cappellari, L. & De Giusti, A. (2008). Binary data compression with and without side information at the decoder: the syndrome-based approach using off-the-shelf turbo codecs.
    URL: *http://arxiv.org/pdf/0902.0562*

Cappellari, L. & Mian, G. A. (2006a). An algorithm for intra-frame video coding based on continuous-valued syndromes, *Conf. Rec. of 40th IEEE Asilomar Conf. on Signals, Syst. and Comput.*, Pacific Grove, CA, U.S.A., pp. 1090–1094.

Cappellari, L. & Mian, G. A. (2006b). A practical algorithm for distributed source coding based on continuous-valued syndromes, *Proc. of European Signal Process. Conf. (EUSIPCO)*, Florence, Italy.

Chou, P. A., Lookabaugh, T. & Gray, R. M. (1989). Entropy-constrained vector quantization, *IEEE Trans. Acoust., Speech, Signal Process.* **37**(1): 31–42.

Ciliberti, S., Mezard, M. & Zecchina, R. (2005). Message passing algorithms for non-linear nodes and data compression.
URL: *http://arxiv.org/pdf/cond-mat/0508723*

Cover, T. M. & Thomas, J. A. (2006). *Elements of Information Theory*, John Wiley & Sons, Inc., Hoboken, NJ, USA.

Garcia-Frias, J. & Zhao, Y. (2001). Compression of correlated binary sources using turbo codes, *IEEE Commun. Lett.* **5**(10): 417–419.

Gel'fand, S. & Pinsker, M. (1980). Coding for channel with random parameters, *Probl. Contr. Inf. Theory* **9**(1): 19–31.

Girod, B., Aaron, A. M., Rane, S. & Rebollo-Monedero, D. (2005). Distributed video coding, *Proc. IEEE* **93**(1): 71–83.

ITU-T & ISO/IEC: JTC1/SC29/WG11 (2007). Advanced video coding for generic audiovisual services, ITU-T Recommendation H.264, ISO/IEC 14496-10 (MPEG-4 AVC). (including SVC extension).

Kschischang, F. R., Frey, B. J. & Loeliger, H.-A. (2001). Factor graphs and the sum-product algorithm, *IEEE Trans. Inf. Theory* **47**(2): 498–519.

Liu, Z., Cheng, S., Liveris, A. D. & Xiong, Z. (2006). Slepian-Wolf coded nested lattice quantization for Wyner-Ziv coding: High-rate performance analysis and code design, *IEEE Trans. Inf. Theory* **52**(10): 4358–4379.

Liveris, A. D., Xiong, Z. & Georghiades, C. N. (2002). Compression of binary sources with side information at the decoder using LDPC codes, *IEEE Commun. Lett.* **6**(10): 440–442.

Liveris, A. D., Xiong, Z. & Georghiades, C. N. (2003). Distributed compression of binary sources using conventional parallel and serial concatenated convolutional codes, *Proc. of IEEE Data Compression Conf.*, pp. 193–202.

MacKay, D. (1999). Good error-correcting codes based on very sparse matrices, *IEEE Trans. Inf. Theory* **45**(2): 399–431.

Marcellin, M. W. (1994). On entropy-constrained trellis coded quantization, *IEEE Trans. Commun.* **42**(1): 14–16.

Marcellin, M. W. & Fisher, T. R. (1990). Trellis coded quantization of memoryless and Gauss-Markov sources, *IEEE Trans. Commun.* **38**(1): 82–93.

Martinian, E. & Yedidia, J. S. (2003). Iterative quantization using codes on graphs, *Proc. of 41st Annual Allerton Conf. on Commun., Control and Comput.*, pp. 1317–1326.

Pradhan, S. S., Chou, J. & Ramchandran, K. (2003). Duality between source coding and channel coding and its extension to the side information case, *IEEE Trans. Inf. Theory* **49**(5): 1181–1203.

Pradhan, S. S. & Ramchandran, K. (1999). Distributed source coding using syndromes (DISCUS): design and construction, *Proc. of IEEE Data Compression Conf.*, pp. 158–167.

Pradhan, S. S. & Ramchandran, K. (2003). Distributed source coding using syndromes (DISCUS): design and construction, *IEEE Trans. Inf. Theory* **49**(3): 626–643.

Pradhan, S. S. & Ramchandran, K. (2005). Generalized coset codes for distributed binning, *IEEE Trans. Inf. Theory* **51**(10): 3457–3474.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition, *Proc. IEEE* **77**(2): 257–286.

Roumy, A., Lajnef, K. & Guillemot, C. (2007). Rate-adaptive turbo-syndrome scheme for slepian-wolf coding, *Conf. Rec. of 41st IEEE Asilomar Conf. on Signals, Syst. and Comput.*, pp. 545–549.

Servetto, S. D. (2007). Lattice quantization with side information: Codes, asymptotics, and applications in sensor networks, *IEEE Trans. Inf. Theory* **53**(2): 714–731.

Shannon, C. E. (1948). A mathematical theory of communication, *The Bell Syst. Tech. J.* **27**: 379–423, 623–656.

Shannon, C. E. (1959). Coding theorems for a discrete source with a fidelity criterion, *IRE Convention Record*, Vol. 4, pp. 142–163.

Slepian, D. & Wolf, J. K. (1973). Noiseless coding of correlated information sources, *IEEE Trans. Inf. Theory* **19**(4): 471–480.

Stanković, V., Liveris, A. D., Xiong, Z. & Georghiades, C. N. (2006). On code design for the Slepian-Wolf problem and lossless multiterminal networks, *IEEE Trans. Inf. Theory* **52**(4): 1495–1507.

Tu, Z., Li, J. & Blum, R. S. (2005). An efficient SF-ISF approach for the Slepian-Wolf source coding problem, *EURASIP J. Appl. Signal Process.* **2005**(6): 961–971.

Wagner, A. B., Tavildar, S. & Viswanath, P. (2008). Rate region of the quadratic gaussian two-encoder source-coding problem, *IEEE Trans. Inf. Theory* **54**(5): 1938–1961.

Wyner, A. D. (1974). Recent results in the Shannon theory, *IEEE Trans. Inf. Theory* **20**(1): 2–10.

Wyner, A. D. & Ziv, J. (1976). The rate-distortion function for source coding with side information at the decoder, *IEEE Trans. Inf. Theory* **22**(1): 1–10.

Yang, Y., Stanković, V., Xiong, Z. & Zhao, W. (2008). On multiterminal source code design, *IEEE Trans. Inf. Theory* **54**(5): 2278–2302.

Zamir, R. (1996). The rate loss in the Wyner-Ziv problem, *IEEE Trans. Inf. Theory* **42**(6): 2073–2084.

Zamir, R. & Feder, M. (1996). On lattice quantization noise, *IEEE Trans. Inf. Theory* **42**(4): 1152–1159.

Zamir, R., Shamai, S. & Erez, U. (2002). Nested linear/lattice codes for structured multiterminal binning, *IEEE Trans. Inf. Theory* **48**(6): 1250–1276.
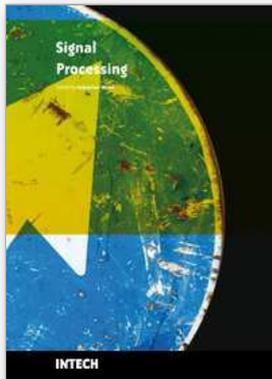
Signal Processing

Edited by Sebastian Miron

This book intends to provide highlights of the current research in signal processing area and to offer a snapshot of the recent advances in this field. This work is mainly destined to researchers in the signal processing related areas but it is also accessible to anyone with a scientific background desiring to have an up-to-date overview of this domain. The twenty-five chapters present methodological advances and recent applications of signal processing algorithms in various domains as telecommunications, array processing, biology, cryptography, image and speech processing. The methodologies illustrated in this book, such as sparse signal recovery, are hot topics in the signal processing community at this moment. The editor would like to thank all the authors for their excellent contributions in different areas of signal processing and hopes that this book will be of valuable help to the readers.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

INTECH
open science | open minds