Joint Source/Channel Coding with Low Density Parity Check Matrices

Maria Fresia, Fernando Pérez-Cruz, H. Vincent Poor and Sergio Verdú

Abstract

The objectives of this paper are two-fold, first to present the problem of joint source and channel coding from a graphical model perspective and second to propose a structure that uses a new graphical model for jointly encoding and decoding of a redundant source. In the first part of the paper, relevant contributions to joint source and channel coding, ranging from the Slepian-Wolf problem to joint decoding of variable length codes with state-of-the-art source codes, are reviewed and summarized. In the second part, a double low-density parity-check (LDPC) code for joint source and channel coding is proposed. The double LDPC code can be decoded as a single bipartite graph using standard belief propagation and its limiting performance is analyzed by using extrinsic information transfer (EXIT) chart approximations.

M. Fresia was with Princeton University. She is now with Infineon Technologies AG, Munich, Germany. H. Vincent Poor and S. Verdú are with the Department of Electrical Engineering, Princeton University, Princeton, NJ (USA). F. Pérez-Cruz is with Universidad Carlos III de Madrid (Spain). E-mail: maria.fresia@infineon.com, fernando@tsc.uc3m.es, poor@princeton.edu and verdu@princeton.edu

This work was partially supported by National Science Foundation under Grants CNS-06-25637, NCR-0074277 CCR-0312879, CCF-0728445 and CCF-0635154. This work was also partially funded by Spanish government (Ministerio de Educación y Ciencia TEC2009-14504-C02-01, Consolider-Ingenio 2010 CSD2008-00010). F. Pérez-Cruz was supported by Marie Curie Fellowship 040883-AI-COM.

I. Introduction

A cornerstone of information theory is the separation principle, which states that there is nothing to be gained from joint data compression and channel transmission. This principle holds for stationary channels and sources provided that the delay is unbounded [1]. The separate design of source and channel codes allows for diverse sources to share the same digital media. However, for finite-blocklength communications, the separation principle does not apply in general and the channel decoder can exploit the residual redundancy left by the source code to reduce the overall error rate. Furthermore, as pointed out in [2], some communications standards (e.g. Universal Mobile Telecommunications System) specify block sizes that are too small for the asymptotic limits to be representative. In many such cases, in practice the sources are transmitted uncoded, while state-of-the-art block channel encoders are used to protect the transmitted data against channel errors.

The main idea behind joint source and channel (JSC) coding is that for fixed-size blocks the source (encoded or not) is redundant, and this redundancy can be exploited at the decoder side. There are two main approaches to JSC coding:

- A given source compression format (e.g. JPEG) is jointly decoded with a channel decoder (e.g. belief propagation for a sparse-graph code) [3]-[6]. These *ad hoc* approaches use high-level information from the source code to drive the channel decoding process and they are highly dependent on the source encoder.
- A Markovian model for the source is assumed and it is jointly decoded with the factor graph of the channel code [2][7]-[11]. In some of these approaches the source need not be known a priori.

This paper focuses on the latter approach, in which the graphical model structure of the source and the code are jointly exploited. Standard fixed-to-variable length sources codes (e.g. Lempel-Ziv or arithmetic codes) are not well suited for state-of-the-art channel codes (e.g. turbo or low-density parity-check codes), because they need long sequences to achieve the entropy of the source and a single error would catastrophically affect the decoded source data. These peculiar features pose strict constraints: if the source encoding is restarted after each block of the channel code (in order to circumscribe the catastrophic behavior of residual channel errors) the compression performance might be low (depending on the channel block size). On the hand, if the source encoder is not restarted for the entire source transmission (in order to obtain a good compression performance), a single residual error on the channel decoded stream might disrupt all the subsequent source decoded data.

We advocate a new structure for the JSC encoder, which includes a fixed-to-fixed length source encoder, followed by a low-density parity-check (LDPC) code, to compress the source. As the fixed-to-fixed length source code, we also use an LDPC code. Thus, at the encoder side the structure is given by two concatenated LDPC codes, which, at the decoder side, can be represented as a single bipartite graph and decoded by means of belief propagation. Thanks to the double LDPC structure, it is possible, by means of a design parameter, to find the desired trade-off between the error floor at high signal-to-noise ratio, E_b/N_0, and the bit error rate at lower E_b/N_0. In particular the error floor is due mainly to the source code, while the error rate is due to the channel code.

The rest of the paper is organized as follows. In Section II, we review the JSC coding literature and provide pointers to previous work. We present the proposed structure in Section III and analyze the performance of the double LDPC-JSC decoders in Section IV. In Section V, we optimize the LDPC codes to achieve higher compression and protection rates to approach the entropy of the source and the channel capacity. We illustrate the benefits of the proposed double-LDPC code for JSC decoding in Section VI, where we discuss several examples in which the benefits and drawbacks of the double LDPC codes can be understood.

II. Literature Review

The results in this paper draw on the ideas of many different authors. In this section we summarize their contributions and we partially review the literature on JSC coding. The standard solutions for JSC decoding assume that the source code leaves some residual redundancy, due to constraints on either delay or complexity, in the encoded sequence and that a joint decoder that exploits these redundancies can further reduce its error rate.

After turbo codes were proposed in the early nineties, some authors pursued the idea of JSC by transforming the decoding process of variables length codes, such as Huffman or arithmetic codes, typically used in image and audio coding. These joint decoding schemes are based on two central ideas: the variable length encoders re not ideal and some bit streams are not admissible; and they can be soft decoded similarly to convolutional codes. In 1995, Hagenauer proposed to use a soft-output Viterbi algorithm to perform JSC decoding using a posteriori information from the source code to control the channel decoder [12]. References [11], [13], [14], [15] propose to describe the Huffman code with a trellis structure allowing then the joint decoding with a linear channel code (in particular turbo codes are used). The authors in [16] extend this idea to arithmetic codes and in [17] to reversible variable length codes. References [18], [19], [20] propose jointly decoding a Lempel-Ziv code and a channel code. Finally, [21] develops a maximum a posteriori decoder for a variable entropy decoder with a binary symmetric channel and without any channel code.

Among the authors that use a model to describe the redundancy left in the source, we can distinguish the work by García-Frias *et al.*, in which they combine a hidden Markov source model and a turbo code for JSC decoding [7]. They extended their results to sources with unknown parameters of the Markovian source, where the decoder jointly estimated them with the source bits [9], [22]. Moreover, they extended their work including LDPC codes [23], and allowing non-binary sources [24], [25]. Other relevant contributions to solve this problem for memoryless and hidden Markov sources are proposed in [10], [26]-[30]. Reference [31] proposes the LOTUS codes, in which a recursive convolutional encoder follows a irregular LDPC codes. LOTUS codes are a general structure for error protection that includes as particular cases: turbo codes; LDPC codes; and Irregular Repeat and Accumulate (IRA) codes. In [31] the LOTUS codes are optimized for JSC coding and it was shown that Quenched BP decoding is more robust to channel erasures than arithmetic coding.

We also highlight a recent contribution [2], which uses the discrete universal denoiser (DUDE) [32] for JSC decoding. The DUDE constructs a conditional model for each symbol using its neighboring symbols and the information of the complete source. It uses this conditional model to detect errors in the source and denoise it. In [2], the authors use the DUDE posterior estimate for each symbol to iteratively decode the received source together with an LDPC decoder.

The Slepian-Wolf problem for independently compressing two correlated sources [33] that are jointly decoded, can be seen as a problem of channel coding with side information [34] and it is probably the best-known application of channel codes for source coding. In a nutshell, one source is compressed to its entropy and the other is channel encoded using a capacity-achieving code, in which only the redundant bits (parity checks) are used to compress the second source. At the decoder side, the first source is independently decoded. The first source can be interpreted as a noisy version of the second source. The parity checks bits, which describe the second source, are appended to the decoded bits of the first source, and the second source is recovered by running the channel decoder on this channel codeword.

There are several publications that address the Slepian-Wolf problem with different channel codes, such as turbo codes [35], [36], LDPC codes [37], [23] or low-density generator-matrices codes [38]. Most of the approaches assume that the sources are coupled by a binary symmetric channel with known crossover probability. Others assume that this probability is unknown and it is estimated as part of the iterative decoding procedure [39], others use a Markovian model to measure the correlation between the sources for binary [40] and non-binary sources [41].

From the solution of the Slepian-Wolf problem we can generalize the use of channel codes to source coding, in which the side information is the model of the source. Most of the proposed solutions apply turbo codes to independently and identically distributed (i.i.d.) sources [39], [42], [43], [44]. Also LDPC codes have been proposed to solve the source-coding problem, but instead of using the parity-checks bits, the syndrome bits are used (as initially proposed in [45]).

LDPC codes for channel coding are well known and broadly used, thanks to their ability to approach capacity in most channels of interest and their linear coding and decoding complexity [46]. On the other hand, their application to source coding is not so widely spread [47], because fixed-to-fixed source coding is not the mainstream solution to this problem.

Let us consider an i.i.d. Bernoulli source with success probability $p < 1/2$ and an $\ell \times n$ parity check matrix $\mathbf{H}_{sc}$. We can compress the source using this parity check matrix:

$$\mathbf{b} = \mathbf{H}_{sc}\mathbf{s}, \tag{1}$$

where $\mathbf{s}$ is a column vector representing the n-source bits and $\mathbf{b}$ is the ℓ-bit column-vector compressed sequence. The decoding procedure is based on approximate syndrome decoding using loopy belief propagation (BP), just as the decoding for LDPC channel codes. LDPC for source decoding uses the prior probability from the source and iterates between variable and check nodes until the syndrome is $\mathbf{b}$.

This idea that we have outlined for i.i.d. Bernoulli sources can be extended to (hidden) Markovian sources by appending the graphical model of the source to the source bits and running the BP algorithm over the joint graph. For other sources, reference [47] proposes the application of the Burrows-Wheeler transform [48] that transforms the redundant source into an approximately piecewise i.i.d. Bernoulli source [49]. These and other ideas are exploited in [50] to build a universal source encoder using LDPC codes, which exhibits excellent performance relative to established universal data compression algorithms such as Lempel-Ziv.

III. Double LDPC Codes for JSC Coding

A. Encoder structure

A common approach to JSC coding protects a source using a rate-n/m LDPC code and decodes it jointly using the source statistics. In this paper, we propose a different structure for the encoder. First, we compress the source using an LDPC code (1). Second, we protect the compressed bits with another LDPC code:

$$\mathbf{c} = \mathbf{G}_{cc}^{\top}\mathbf{b} = \mathbf{G}_{cc}^{\top}\mathbf{H}_{sc}\mathbf{s}, \tag{2}$$

where $\mathbf{G}_{cc}$ is an $\ell \times m$ LDPC generator matrix and $\mathbf{c}$ is the m-dimensional codeword to be transmitted. We would refer to the source encoder as $\mathcal{C}_{sc} = (\mathbf{H}_{sc}, \mathbf{G}_{sc})$ and to the channel encoder as $\mathcal{C}_{cc} = (\mathbf{H}_{cc}, \mathbf{G}_{cc})$. In other words, we have included a source encoder between the redundant source and the channel encoder, but the overall rate remains unchanged and equal to n/m. This structure, which we have been pursued in [51], was introduced in [31], but the authors did not explore it in favor of the LOTUS codes.

The JSC decoder considered in this paper, is depicted in Fig. 1 as a single bipartite graph. The left box represents the source code sparse Tanner graph and the right box represents the channel code. The solid bold lines joining the ℓ factor nodes from the source code to the first variable nodes in the channel code represent the bits compressed by the source code and the message bits at the channel code. We have not included the Markovian source model over the n leftmost bits, to avoid cluttering the figure.

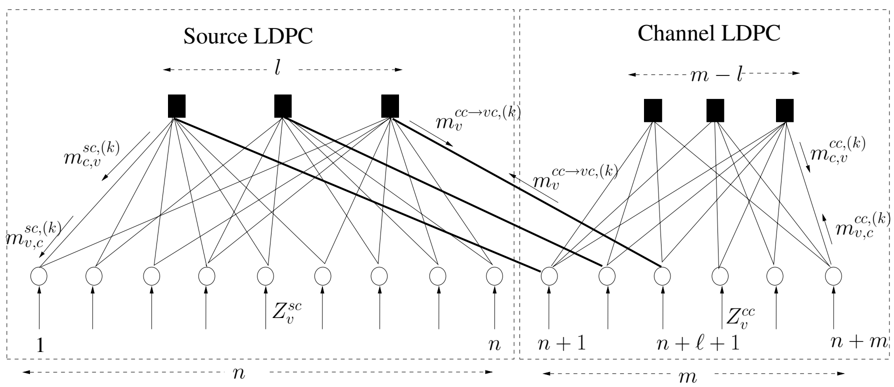

Fig. 1. Joint Tanner graph decoding scheme.

B. Decoder structure

Two sparse bipartite graphs compose the decoder, as shown in Fig. 1, where each check node of the source code (left) is connected to a single variable node of the channel code (right). The joint decoder runs in parallel. First the variable nodes inform the check nodes about their log-likelihood ratios (LLRs) and then the check nodes respond with their LLR constraints for each variable node.

Let us consider the k^{th} iteration of the decoder. For the sake of clarity, we describe the two decoders with separate notation:

- $m_{v,c}^{sc,(k)}$ and $m_{v,c}^{cc,(k)}$ are, respectively, the message passed from the v^{th} variable node to the c^{th} check node of the source code ($\mathcal{C}_{sc}$) and channel code ($\mathcal{C}_{cc}$);
- $m_{c,v}^{sc,(k)}$ and $m_{c,v}^{cc,(k)}$ are, respectively, the message passed from the c^{th} check node to the v^{th} variable node of $\mathcal{C}_{sc}$ and $\mathcal{C}_{cc}$;

- $m_v^{sc\to cc,(k)}$ is the message passed from the check node in $\mathcal{C}_{sc}$ connected to the v^{th} variable node in $\mathcal{C}_{cc}$;
- $m_v^{cc\to sc,(k)}$ is the message passed from the v^{th} variable node in $\mathcal{C}_{cc}$ connected to the c^{th} check node in $\mathcal{C}_{sc}$;
- Z_v^{sc} and Z_v^{cc} represent, respectively, the LLRs for the variable nodes for $v = 1, \ldots, n$ (i.e. the variable nodes of the source decoder) and for $v = n+1, \ldots, n+m$ (i.e. the variable nodes of the channel decoder).

$m_v^{sc\to cc,(k)}$ and $m_v^{cc\to sc,(k)}$ are indexed only by v, because each check node in $\mathcal{C}_{cs}$ is connected to only a single variable node in $\mathcal{C}_{cc}$.

For independent binary sources transmitted over a binary input additive white Gaussian noise (BI-AWGN) channel, $Z_v^{sc} = \log(\frac{1-p_v}{p_v})$ (where $p_v = \mathbb{P}[s_v = 1]$), and $Z_v^{cc} = \frac{2r_v}{\sigma_n^2}$, where $r_v = (1 - 2x_v) + n_v$, and σ_n^2 is the channel noise variance.

The messages between variable nodes and check nodes follow the same procedure as standard belief propagation. First the variable nodes send their LLRs to the check nodes and the corresponding messages are given by

$$m_{v,c}^{sc,(k)} = Z_v^{sc} + \sum_{c' \neq c} m_{c',v}^{sc,(k-1)}, \tag{3}$$

$$m_{v,c}^{cc,(k)} = Z_v^{cc} + m_v^{sc\to cc,(k-1)} + \sum_{c' \neq c} m_{c',v}^{cc,(k-1)}, \tag{4}$$

$$m_v^{cc\to sc,(k)} = Z_v^{cc} + \sum_{c'} m_{c',v}^{cc,(k-1)} \qquad \text{and} \tag{5}$$

$$m_{v,c}^{cc,(k)} = Z_v^{cc} + \sum_{c' \neq c} m_{c',v}^{cc,(k-1)}, \tag{6}$$

where (3) runs for $v = 1, \ldots, n$; (4) and (5) for $v = n+1, \ldots, \ell$; and (6) for $v = n+\ell+1, \ldots, n+m$. Notice that $m_{c',v}^{sc,(0)} = 0$, $m_{c',v}^{cc,(0)} = 0$ and $m_v^{sc\to cc,(0)} = 0$.

The messages between the check nodes and the variables nodes are given by

$$\tanh\left(\frac{m_{c,v}^{sc,(k)}}{2}\right) = \tanh\left(\frac{m_v^{cc\to sc,(k)}}{2}\right) \prod_{v' \neq v} \tanh\left(\frac{m_{v',c}^{sc,(k)}}{2}\right), \tag{7}$$

$$\tanh\left(\frac{m_v^{sc\to cc,(k)}}{2}\right) = \prod_{v'} \tanh\left(\frac{m_{v',c}^{sc,(k)}}{2}\right) \qquad \text{and} \tag{8}$$

$$\tanh\left(\frac{m_{c,v}^{cc,(k)}}{2}\right) = \prod_{v' \neq v} \tanh\left(\frac{m_{v',c}^{cc,(k)}}{2}\right), \tag{9}$$

where (7) and (8) run for $c = 1, \ldots, \ell$, while (9) runs for $c = \ell+1, \ldots, m$.

After K iterations of the decoding process, the source bits are estimated by

$$\hat{s_v} = \begin{cases} 0 & \text{if} \quad L(s_v) \geq 0 \\ 1 & \text{if} \quad L(s_v) \leq 0, \end{cases}$$

where $L(s_v)$ is the LLR of the source bit s_v for $v = 1, \ldots, n$ computed as

$$L(s_v) = Z_v^{sc} + \sum_c m_{c,v}^{sc,(K)}.$$

IV. Asymptotic Analysis: EXIT charts

The belief propagation (BP) algorithm allows analysis of finite-length codes, but is impractical for studying the asymptotic behavior of sparse codes. To analyze the behavior of the scheme in the limit of infinite block length, the standard analysis tool for factor graph-based codes under belief propagation iterative decoding is the density evolution (DE) analysis [52], [53]. The DE analysis is typically computationally heavy, and not very well conditioned numerically. A simple and more manageable way to study the graph behavior consists of an approximation of the DE analysis called the extrinsic information transfer (EXIT) chart analysis [54], which corresponds to the DE analysis by imposing the restriction that the message densities are of a particular form. More specifically, the EXIT chart with Gaussian approximation assumes that at every iteration, the BP messages are Gaussian having a particular symmetry condition which imposes that $\sigma^2 = 2\mu$.

Since the decoder is composed of two separated LDPC decoders that exchange information, it is not possible to combine the evolution of the two decoders in a single input-output function. Even if they run in parallel exchanging information, we need to describe the evolution of the source and channel decoders separately.

The following notation is used in the rest of the section:

- $\mathsf{x}_{i_{sc}}$ [$\mathsf{x}_{i_{cc}}$] denotes the mutual information between a message sent along an edge (v, c) with "left-degree" i_{sc} [i_{cc}] and the symbol corresponding to the bitnode v for the LDPC source [channel] decoder;
- x_{sc} [x_{cc}] denotes the average of $\mathsf{x}_{i_{sc}}$ [$\mathsf{x}_{i_{cc}}$] over all edges (v, c);
- $\mathsf{y}_{j_{sc}}$ [$\mathsf{y}_{j_{cc}}$] denotes the mutual information between a message sent along an edge (c, v) with "right-degree" j_{sc} [j_{cc}] and the symbol corresponding to the bitnode v for the LDPC source [channel] decoder;
- y_{sc} [y_{cc}] denotes the average of $\mathsf{y}_{i_{sc}}$ [$\mathsf{y}_{i_{cc}}$] over all edge (c, v).

We consider the class of EXIT functions that make use of Gaussian approximation of the BP messages, which considers the well-known fact that the family of Gaussian random variables is closed under addition (i.e. the sum of Gaussian random variables is also Gaussian, and its mean is the sum of the means of the addends). Imposing the symmetry condition and Gaussianity, the conditional distribution of each message $\mathcal{L}$ in the direction $v \to c$ is Gaussian $\sim \mathcal{N}(\mu, 2\mu)$, for some value $\mu \in \mathbb{R}_+$. Hence, letting V denote the corresponding bitnode variable, we have

$$I(V; \mathcal{L}) = 1 - \mathbb{E}\left[\log_2\left(1 + e^{-\mathcal{L}}\right)\right] \triangleq J(\mu),$$

where $\mathcal{L} \sim \mathcal{N}(\mu, 2\mu)$. Notice that, by using the function $J(\cdot)$, the capacity of a BIAWGN channel with noise variance σ_n^2 can be expressed as $C = J(2/\sigma_n^2)$.

Generally, an LDPC code is defined by $\lambda(x) = \sum_i \lambda_i x^{i-1}$ and $\rho(x) = \sum_j \rho_j x^{j-1}$ which represent the degree distribution of the variable nodes and the check nodes respectively in the edge perspective, i.e., λ_i (ρ_i) is the fraction of edges connected to a degree i variable (check) node. Thus $\Lambda(x) = \sum_i \Lambda_i x^i = \frac{\int_0^x \lambda(u)du}{\int_0^1 \lambda(u)du}$, and $P(x) = \sum_j P_j x^j = \frac{\int_0^x \rho(u)du}{\int_0^1 \rho(u)du}$, represent the degree distribution of the variable nodes and the check nodes respectively in the node perspective, i.e., Λ_i (R_i) is the fraction of variable (check) nodes with degree i. The source code is then defined by the pairs $(\lambda_{sc}(x), \rho_{sc}(x))$ (or $(\Lambda_{sc}(x), P_{sc}(x))$) while the channel code is defined by $(\lambda_{cc}(x), \rho_{cc}(x))$ (or $(\Lambda_{cc}(x), P_{cc}(x))$).

In BP, the message on (v, c) is the sum of all messages incoming to v on all other edges. The sum of Gaussian random variables is also Gaussian, and its mean is the sum of the means of the incoming messages. It follows that

$$\mathsf{x}_i = J((i-1)J^{-1}(\mathsf{y}) + J^{-1}(C)),$$

where C is the mutual information (capacity) between the bitnode variable and the corresponding LLR at the (binary-input symmetric output) channel output. In the LDPC codes, the bitnodes correspond to variables that are observed through a virtual channel by the LDPC decoder. Averaging with respect to the edge degree distribution we have

$$\mathsf{x} = \sum_i \lambda_i J((i-1)J^{-1}(\mathsf{y}) + J^{-1}(C)).$$

As far as checknodes are concerned, we use the well-known quasi-duality approximation and replace checknodes with bitnodes by changing mutual information into entropy (i.e. replacing x by $1 - \mathsf{x}$). Then

$$\mathsf{y}_j = 1 - J((j-1)J^{-1}(1 - \mathsf{x})),$$

and averaging with respect to the edge degree distribution we have

$$\mathsf{y} = 1 - \sum_j \rho_j J((j-1)J^{-1}(1 - \mathsf{x})).$$

Let us consider the "two-channel" scenario induced by the proposed JSC scheme. For the source decoder, the message x_{sc} is given by

$$\mathsf{x}_{sc} = \sum_{i_{sc}} \lambda_{i_{sc}} J_{BSC}\left((i_{sc}-1)J^{-1}(\mathsf{y}_{sc}), p\right), \tag{10}$$

where $J_{BSC}(\cdot)$ is a manipulation of the function $J(\cdot)$ to take into account that the source is binary and i.i.d. with $p = \mathbb{P}[s_v = 1]$, i.e. the equivalent channel is a binary symmetric channel (BSC) with crossover probability p (and hence capacity $1 - H(p)$). In particular, the probability density function (pdf) of the LLR output from the equivalent channel is given by $p\delta(x+L) + (1-p)\delta(x-L)$. Therefore, the function J_{BSC} can be expressed as

$$J_{BSC}(\mu, p) = (1-p)I(V; \mathcal{L}^{(1-p)}) + pI(V; \mathcal{L}^{(p)}),$$

where $\mathcal{L}^{(1-p)} \sim \mathcal{N}(\mu + L, 2\mu)$, and $\mathcal{L}^{(p)} \sim \mathcal{N}(\mu - L, 2\mu)$.

The message y_{sc} is given by

$$\mathsf{y}_{sc} = 1 - \sum_{i_{cc}, j_{sc}} \Lambda_{i_{cc}} \rho_{j_{sc}} J\left((j_{sc}-1)J^{-1}(1-\mathsf{x}_{sc}) + J^{-1}(1-c_{i_{cc}})\right), \tag{11}$$

where $c_{i_{cc}} = J(i_{cc}J^{-1}(\mathsf{y}_{cc})) + J^{-1}(C))$ is the message generated by a variable node of degree i_{cc} of the channel decoder. Notice that we average over all possible values of i_{cc} through $\Lambda_{i_{cc}}$.

For the channel decoder, the message x_{cc} is given by

$$\begin{aligned} \mathsf{x}_{cc}^{(k)} &= R_{cc} \sum_{j_{sc}, i_{cc}} P_{j_{sc}} \lambda_{i_{cc}} J\left((i_{cc}-1)J^{-1}(\mathsf{y}_{cc}) + J^{-1}(C) + J^{-1}(c_{j_{sc}})\right) \\ &+ (1-R_{cc}) \sum_{i_{cc}} \lambda_{i_{cc}} J\left((i_{cc}-1)J^{-1}(\mathsf{y}_{cc}) + J^{-1}(C)\right), \end{aligned} \tag{12}$$

where $c_{j_{sc}} = 1 - J\left(j_{sc}J^{-1}(1-\mathsf{x}_{sc})\right)$ is the message generated by a check node of degree j_{sc} of the source decoder. We average over all possible values of j_{sc} through $P_{j_{sc}}$. Notice that (12) is composed of two parts to take into account the fact that a fraction of R_{cc} variable nodes of the channel decoder are connected to the check nodes of the source decoder (i.e. they have the extra message $\downarrow c_{j_{sc}}$), while the remaining $1 - R_{cc}$ are connected only to the transmission channel. Since the data are transmitted over a BIAWGN channel, then $C = J(2/\sigma_n^2)$, and therefore $J^{-1}(C) = 2/\sigma_n^2$.

Finally, the message y_{cc} is given by

$$\mathsf{y}_{cc} = 1 - \sum_{j_{cc}} \rho_{j_{cc}} J\left((j_{cc}-1)J^{-1}(1-\mathsf{x}_{cc})\right). \tag{13}$$

After K iterations we need to obtain the conditional pdf of the LLRs output by the source bits (i.e. the variable nodes of the source code) in order to compute the bit error rate (BER). Without taking into account the message generated by the equivalent channel, these LLRs are Gaussian, i.e. for a variable node with degree i_{sc}, they are $N(\mu_{i_{sc}}, 2\mu_{i_{sc}})$, where $\mu_{i_{sc}} = i_{sc}J^{-1}(\mathrm{y}_{sc})$. Since the equivalent channel is modeled as a BSC, the pdf of the overall message is a Gaussian mixture weighted by the value of p and is given by

$$pN(\mu_{i_{sc}} - L, 2\mu_{i_{sc}}) + (1-p)N(\mu_{i_{sc}} + L, 2\mu_{i_{sc}}).$$

Averaging over all possible values we have that the BER is equal to

$$P_e = \sum_{i_{sc}} \lambda_{i_{sc}} \left[pQ(x^{p}_{i_{sc}}) + (1-p)Q(x^{1-p}_{i_{sc}}) \right], \qquad (14)$$

where

$$x^{p}_{i_{sc}} = \sqrt{\mu_{i_{sc}}/2} - L/\sqrt{2\mu_{i_{sc}}}$$

and

$$x^{1-p}_{i_{sc}} = \sqrt{\mu_{i_{sc}}/2} + L/\sqrt{2\mu_{i_{sc}}},$$

and where $Q(\cdot)$ is the Gaussian tail function

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_{x}^{+\infty} e^{-z^2/2} dz.$$

As described in Section III-B, in the finite length simulations both decoders run in parallel: first all the LDPC bitnodes are activated, then all the LDPC checknodes and so on. In the infinite length case we adopt a conceptually easier schedule: for each iteration of the LDPC source decoder, a large number of iterations of the channel decoder are performed in order to reach the fixed point equilibrium; the generated messages are incorporated as "additive messages" to the check nodes of the source LDPC decoder; all the check nodes of the channel LDPC code are activated. This provides a complete cycle of scheduling, which is repeated an arbitrarily large number of times.

The reason for adopting this scheduling instead of the practical one is related to the fact that the EXIT chart can be seen as a multidimensional dynamical system with state variables. Since we are interested in studying the fixed points and the trajectories of this system, we need to reduce the problem to an input-output function and then reduce the number of variables. From the equations above, it is clear that this cannot be done by considering the practical scheduling.

Notice that the conceptually easier schedule is useful in the source code optimization procedure described in the next section: since for the channel code a fixed point is reached at each iteration, the input-output function of the channel code can be viewed as a fixed look-up table that, for a given input, returns a fixed value. As we shall see, this allows to us to reduce the optimization to a linear optimization problem.

V. JSC CODE OPTIMIZATION

In this section we present an optimization procedure for maximizing the transmission rate of the proposed JSC code. We suggest a suboptimal procedure that gives optimal codes when the source and channel rates tend, respectively, to the entropy of the source and the capacity of the channel. First, we compute the optimal channel code assuming the input bits are i.i.d. and equally likely (worse case). This is the standard LDPC optimization for channel coding [46], [55]. Given the optimized channel code, we compute optimal degree distributions for the variable and check nodes in the source code.

Substituting (11) into (10), we can express the input-output function of the source code as

$$\mathsf{x}_{sc}^{(k)} = F_{sc}(x_{sc}^{(k-1)}, p, f_{cc}(x_{sc}^{(k-1)}, C)), \tag{15}$$

where $f_{cc}(x_{sc}^{(k-1)}, C)$ is the input-output function related to the channel code and is derived by substituting (13) into (12).

In a density evolution analysis, the convergence is guaranteed if $F_{sc}(\mathsf{x}_{sc}, p, f_{sc}(\mathsf{x}_{sc}, \sigma^2)) > \mathsf{x}_{sc}$ for $\mathsf{x}_{sc} \in [0, 1]$, which ensures convergence at the fixed point $\mathsf{x}_{sc} = 1$.

Given $(\lambda_{cc}(x), \rho_{cc}(x))$ (i.e. fixing the channel code) and $\rho_{sc}(x)$[1], Eq. (15) is linear with respect to the coefficients of $\lambda_{sc}(x)$, and thus the optimization problem can be written as

$$\max \textstyle\sum_{i_{sc} \geq 2} \frac{\lambda_{i_{sc}}}{i_{sc}} \tag{16}$$

$$\text{subject to} \quad \textstyle\sum_{i_{sc}} \lambda_{i_{sc} \geq 2} = 1, \; 0 \geq \lambda_{i_{sc}} \leq 1 \tag{17}$$

$$F_{sc}(\mathsf{x}_{sc}, p, f_{cc}(\mathsf{x}_{sc}, C)) > x_{sc} \tag{18}$$

$$\lambda_2 < \frac{1}{2\sqrt{p(1-p)}} \cdot \frac{1}{\sum_{j_{sc}} \rho_{j_{sc}}(j-1)}, \tag{19}$$

where (19) represents the stability condition [56].

The optimization sketched above is based on the procedure proposed in [56]. In contrast to [56], we deal with two codes that iterate in parallel and then we add the input-output function of the channel code (i.e. $f_{cc}(x_{sc}^{(k-1)}, C)$) to the optimization.

VI. EXPERIMENTAL RESULTS

In this section, we illustrate the advantages of using two concatenated LDPC codes for joint source-channel coding instead of one structure as typically proposed in JSC coding. We have performed four sets

[1]According to [56], we consider a concentrated right degree distribution of the form $\rho(x) = \rho x^{k-1} + (1-\rho)x^k$ for some $k \geq 2$ and $0 \leq \rho \leq 1$.

of experiments with regular and irregular LDPC codes. In the first we compare the double LDPC code with standard JSC decoding, in which the overall rate protects the source. In the second, we illustrate the advantage of using two concatenated LDPC codes when the overall rate compresses the source. In the third experiment, we show why joint decoding is better than cascade decoding: a channel decoder followed by an independent source decoder. In the final experiment, we explore the use of optimized irregular LDPC codes instead of regular ones. For all the experiments, we use additive white Gaussian noise (AWGN) channels and symmetric Markovian sources.

In the first experiment, we consider a Markovian source with two states, in which the probability of switching states is equal to 0.07. The entropy of this source is 0.366. We transmit this source through an AWGN channel using a rate 1/2 regular-LDPC code with 3 ones per column [57]. We first decode the sequence ignoring the source model and we plot the frame error rate (FER) with a dash-dotted line in Fig. 2. We also decode the sequence incorporating the Markovian source model to the factor graph of the LDPC code. We represent the FER by a dashed line in the figure. The channel encoder benefits from the information from the source to reduce its FER at a significantly lower E_b/N_0.

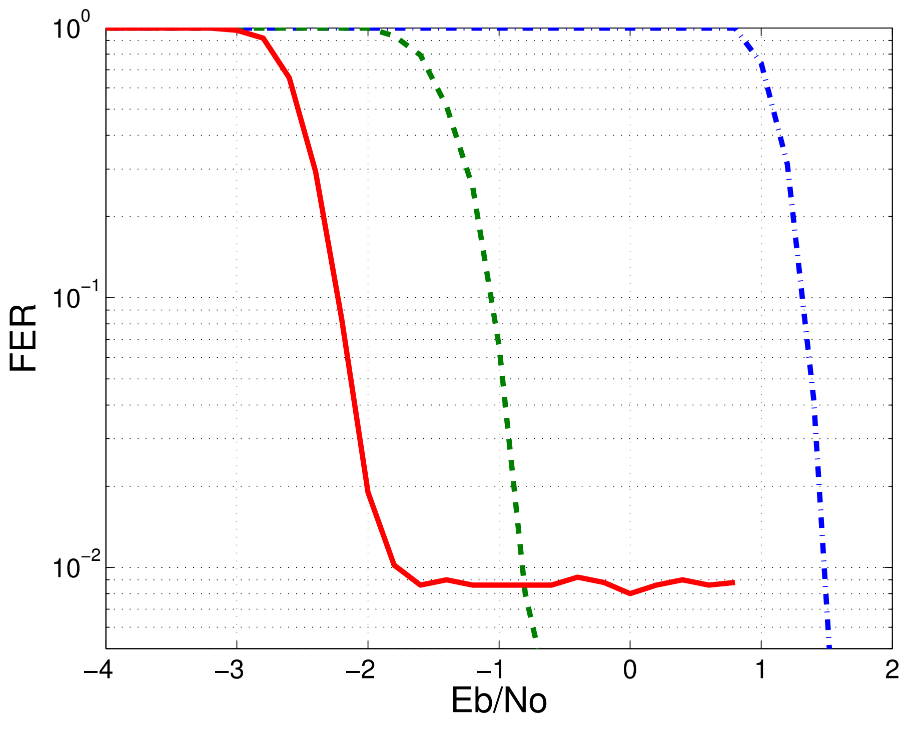

Fig. 2. FER for an LDPC channel decoder (dash-dotted), a standard LDPC-JSC decoder (dashed) and the double LDPC-JSC decoder $n = 3200$ bits.

We finally apply our double LDPC encoder with a rate-1/2 regular-LDPC source code and a rate-1/4 regular-LDPC channel code. The overall rate is 1/2, as in the previous case. We depict the FER in Fig. 2 with a solid line. The FER is reduced at an even lower E_b/N_0 at the expense of an error floor. The error floor is due to residual decoding errors introduced by the fixed-to-fixed source code, as some less-likely words are decoded into a more-likely (incorrect) word. This error floor can be reduced by either increasing the block size or by increasing the compression rate (i.e. less compression). The additional coding gain is due to the lower rate channel encoder. The proposed double LDPC source and channel encoder has a design parameter $\ell \in (0, n)$ that trades off the coding gain and the error floor. The larger ℓ is, the smaller the coding gain and the lower the error floor are.

In order to illustrate the trade-off introduced by the design parameter ℓ, for the second experiment we use three codes. The first scheme, denoted as LDPC-8-2-4, consists of two concatenated LDPC codes with rates $R_{sc} = 2/8$ and $R_{cc} = 2/4$ for source and channel coding respectively. The second scheme, LDPC-8-3-4, consists of two concatenated LDPC codes with rates $R_{sc} = 3/8$ and $R_{cc} = 3/4$. The last scheme, LDPC-8-4, consists of a single LDPC code whose compression rate is $R_{sc} = 4/8$. This code only compresses the source and does not add any redundancy to the transmitted bits.

In Fig. 3, we show the BER as a function of E_b/N_0 for a Markovian source with two states and the probability of switching states is equal to 0.02. For each code there is a set of three plots: the BER predicted by the EXIT chart (dash-dotted line), the BER for codewords with 3200 bits (solid lines) and the BER for codewords with 1600 bits (dashed lines). The three leftmost plots are for LDPC-8-2-4; the three middle plots are for LDPC-8-3-4 and the rightmost plots for LDPC-8-4.

In Fig. 3 we observe the standard behavior of the joint decoder for two concatenated LDPC codes, one for channel coding and the other for source coding. As E_b/N_0 increases there is a sharp decline in the BER due to the channel code operating below capacity and, as expected, this transition is sharper as the code length increases. There is a residual BER at high E_b/N_0, only observable for LDPC-8-2-4 in Fig. 3, due to the source decoder's inability to correctly decompress every word for finite-length codes. This residual BER tends to zero as the codeword length increases, because R_{sc} is above the entropy of the source. The residual BER does not show for LDPC-8-3-4, because its R_{sc} is higher.

In the plots we observe about $1dB$ gain when we compare LDPC-8-2-4 with LDPC-8-3-4 for BER larger than the residual BER. This gain is due to the additional redundancy in LDPC-8-2-4. The price we pay for this gain is a higher residual BER. We can trade off the expected gain and the residual BER by parametrizing the source $R_{sc} = \ell/8$ and channel $R_{cc} = \ell/4$ code rates with ℓ. The value of $\ell \in (0, 4]$ is inversely proportional to the gain and to the residual BER. We can also decrease the residual BER

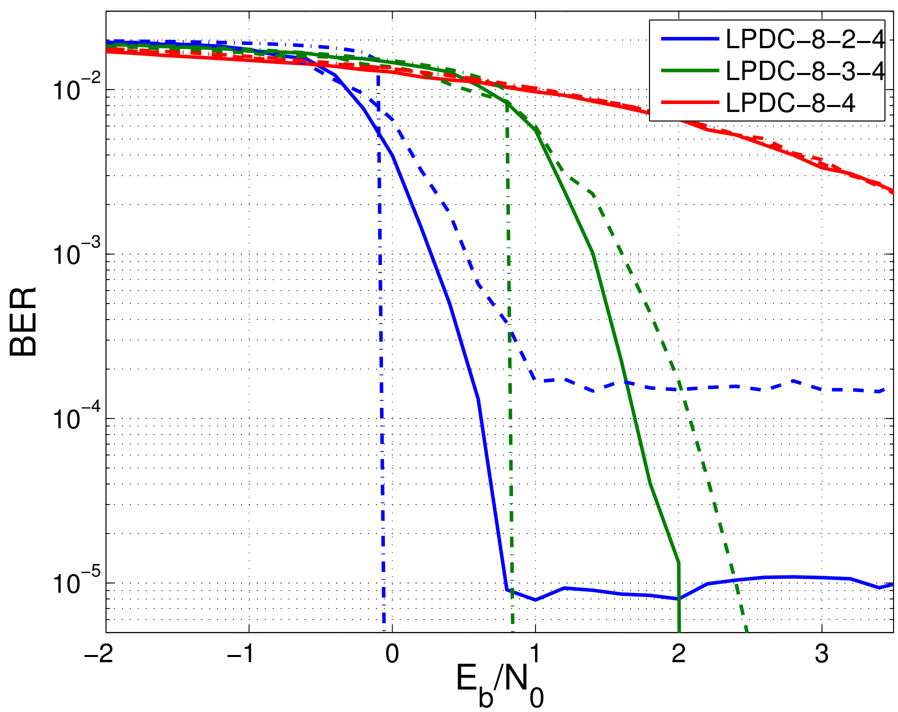

Fig. 3. BER versus E_b/N_0 for LDPC-8-2-4, LDPC-8-3-4 and LDPC-8-4.

by increasing the code length, as illustrated in Fig. 3. For LDPC8-4, the BER does not improve as we increase the code length (the three lines superimpose), even when the decoder has the information about the source statistics.

In the third experiment, we decode the LDPC-8-2-4 scheme with a cascade decoder and compare it with the joint scheme proposed in this paper. The cascade decoder first decodes the channel code assuming the compressed bits are equally likely and i.i.d. and then it decodes the source bits using the LLRs output by the channel decoder.

In Fig. 4, we plot the BER for both decoders as a function of E_b/N_0 for a Markovian source with two states, in which the probability of switching states is equal to 0.02. For each decoder there are two plots: the BER estimated by the EXIT chart (dash-dotted line) and the BER for codewords with 3200 bits (solid line). The two leftmost plots are for the joint decoding and the rightmost plots for the cascade decoding.

For low E_b/N_0 neither decoding procedures are able to decode the transmitted word and they return high BER. The signal to noise ratio is below capacity and the redundancy is not high enough to decode correctly the transmitted words. For high E_b/N_0 both decoding procedures return the same residual

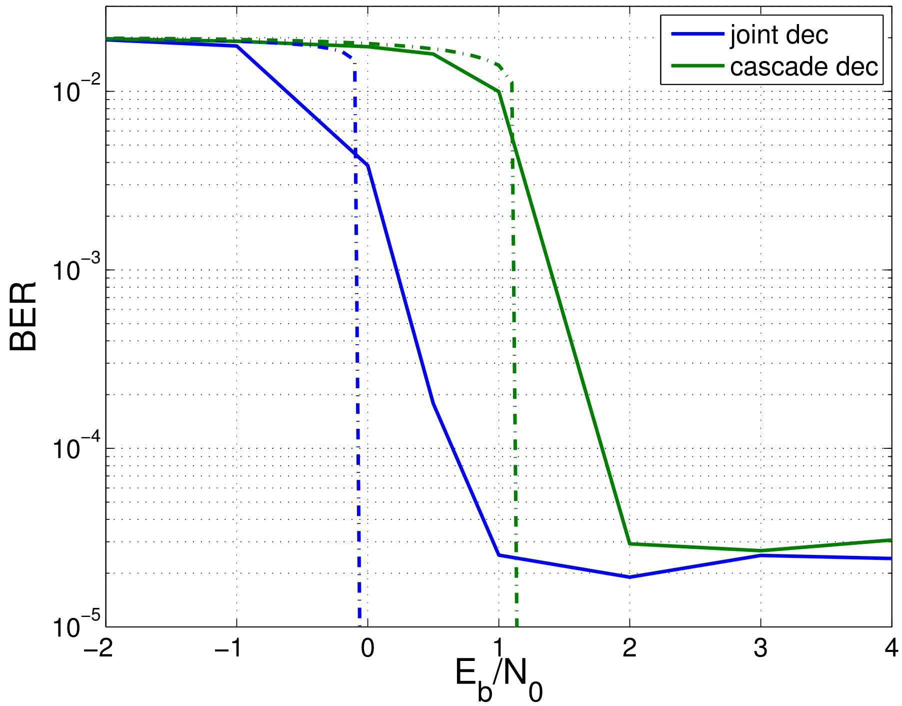

Fig. 4. BER versus E_b/N_0 for the joint and cascade decoder for LDPC-8-2-4.

BER. There are no errors due to the channel decoder and the residual BER is solely due to the source decoder failing to return the correct word. There is a range in E_b/N_0 between 0 and $1dB$, in which the joint decoder returns the residual BER (low BER) and the cascade decoder returns chance level performance (high BER). The difference in performance in this E_b/N_0 range is explained by the fact that the source and channel decoders work together to estimate the correct word. The redundancy not removed by the source encoder gives additional information to the channel decoder to return the correct word. This information is not present in the cascade decoder and therefore the channel decoder is unable to exploit the left redundancy and estimate the correct word. The joint decoder provides, in this particular example, a $1dB$ gain with respect to the cascade decoder. This gain remains unchanged as the codeword length tends to infinity and the performance is studied by means of EXIT chart approximation. This gain disappears only if the rates of the source and channel encoders tend to the entropy and capacity, respectively, as the codeword length increases. But for finite-length codes, the rate of the source cannot approach capacity and we obtain a gain from a joint source-channel decoder.

Finally, we present the performance of irregular optimized LDPC codes obtained by using the method described in Section V. For the channel code, we adopt the first LDPC code with $R_{cc} = 1/2$ in [55].

For the source code, we use a Markovian source with two states, in which the probability of switching states is equal to 0.03. By fixing the degree distribution of the check nodes equal to $\rho(x) = 0.5x^{21}+0.5x^{22}$ and using the optimization procedure described in Section V, we obtain the degree distribution for the variable nodes: $\lambda(x) = 0.098x + 0.274x^3 + 0.025x^7 + 0.292x^9 + 0.075x^{33} + 0.234x^{34}$.

The rate of the source code is $R_{sc} \approx 0.24$ and the overall coding rate is around 2.08. We denote this scheme as LDPCi-8-2-4 and we compare it with LDPC-8-2-4, as their rates are similar. In Fig. 5, we plot the BER for the two codes as a function of the E_b/N_0. For each code there is a set of three plots: the BER for codewords with 3200 bits (solid lines), the BER for codewords with 6400 bits (dash-dotted lines) and the BER for codewords with 12800 bits (dashed lines). The three top plots are for LDPC-8-2-4; the three bottom plots are for LDPCi-8-2-4.

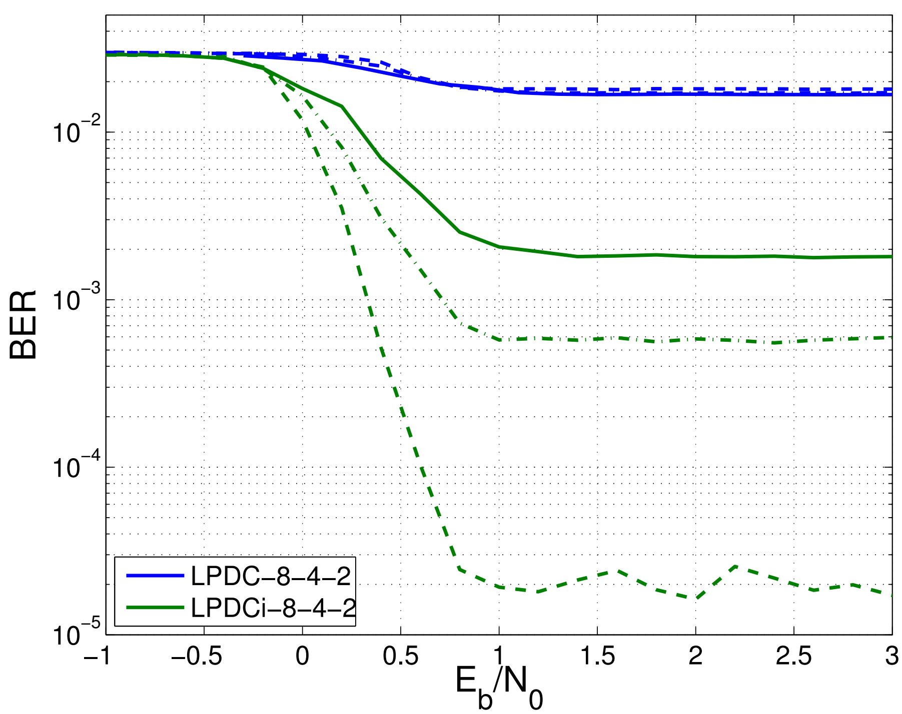

Fig. 5. BER versus E_b/N_0 for the codes LDPC-8-2-4 and LDPCi-8-2-4.

In Fig. 5 we observe that for the regular code as the codeword length increases the residual BER remains constant, while the irregular code is able to reduce its residual BER gradually with the code length. This result is similar to the typical results for LDPC codes for channel coding. Regular codes cannot approach capacity and need a margin in their rate to be able to reduce the BER towards zero, while irregular LDPC codes can achieve capacity as the code length increases. Therefore, if the source code rate approaches the entropy of the source, we need optimized irregular codes in order to reduce the BER with the code length.

REFERENCES

[1] S. Vembu, S. Verdú, and Y. Steinberg, "The source-channel separation theorem revisited," *IEEE Trans. on Inform. Theory*, vol. 41, no. 1, pp. 44–54, Jan. 1995.

[2] E. Ordentlich, G. Seroussi, S. Verdú, and K. Viswanathan, "Universal algorithms for channel decoding of uncompressed sources," *IEEE Trans. on Inform. Theory*, vol. 54, no. 5, pp. 2243–2262, May 2008.

[3] N. Ramzan, S. Wan, and E. Izquierdo, "Joint source-channel coding for wavelet-based scalable video transmission using an adaptive turbo code," *J. Image and Video Processing*, vol. 2007, no. 1, pp. 1–12, Jan. 2007.

[4] M. Zhonghui and W. Lenan, "Joint source-channel decoding of Huffman codes with LDPC codes," *J. Electronics of China*, vol. 23, no. 6, pp. 806–809, Nov. 2006.

[5] L. Pu, Z. Wu, A. Bilgin, M. W. Marcellin, and B. Vasic, "LDPC-based iterative joint source-channel decoding for JPEG2000," *IEEE Trans. on Image Processing*, vol. 16, no. 2, pp. 577–581, Feb. 2007.

[6] L. Yin, J. Lu, and Y. Wu, "LDPC-based joint source and channel coding scheme for multimedia communications," in *Proc. 8th Int. Conf. on Commun. Systems*, Singapore, Nov. 2002.

[7] J. Garcia-Frias and J. D.Villasenor, "Combining hidden Markov source models and parallel concatenated codes," *IEEE Commun. Letters*, vol. 1, no. 7, pp. 111–113, Jul. 1997.

[8] T. Hindelang, J. Hagenauer, and S. Heinen, "Source-controlled channel decoding: Estimation of correlated parameters," in *Proc. 3rd Int. ITG Conf. on Source and Channel Coding*, Munich, Germany, Jan. 2000.

[9] J. Garcia-Frias and J. D.Villasenor, "Joint turbo decoding and estimation of hidden Markov models," *IEEE J. Sel. Areas Commun.*, vol. 19, no. 9, pp. 1671–1679, Sep. 2001.

[10] G.-C. Zhu and F. Alajaji, "Joint source-channel turbo coding for binary Markov sources," *IEEE Trans. on Wireless Commun.*, vol. 5, no. 5, pp. 1065–1075, May 2006.

[11] A. Guyader, E. Fabre, C. Guillemot, and M. Robert, "Joint source-channel turbo decoding of entropy coded sources," *IEEE J. Sel. Areas Commun.*, vol. 19, no. 9, pp. 1680–1696, Sep. 2001.

[12] J. Hagenauer, "Source-controlled channel decoding," *IEEE Trans. on Commun.*, vol. 43, no. 9, pp. 2449–2457, Sep. 1995.

[13] J. Hagenauer and R. Bauer, "The turbo principle in joint source channel decoding of variable length codes," in *Proc. IEEE Inform. Theory Workshop*, Cairns, Australia, Sep. 2001.

[14] R. Bauer and J. Hagenauer, "Symbol-by-symbol MAP decoding of variable length codes," in *Proc. 3rd Int. ITG Conf. on Source and Channel Coding*, Munich, Germany, Jan. 2000.

[15] ——, "On variable length codes for iterative source/channel decoding," in *Proc. IEEE Data Compression Conference*, Snowbird, Utah, Mar. 2001.

[16] M. Grangetto, P. Cosman, and G. Olmo, "Joint source/channel coding and MAP decoding of arithmetic codes," *IEEE Trans. on Commun.*, vol. 53, no. 6, pp. 1007–1016, Jun. 2005.

[17] K. Lakovic and J. D. Villasenor, "Parallel concatenated codes for iterative source-channel decoding," in *Proc. 39th Annual Allerton Conf. on Commun., Computing and Control*, Monticello, IL, Oct. 2001.

[18] S. Lonardi and W. Szpankowski, "Joint source-channel LZ77 coding," in *Proc. IEEE Data Compression Conference*, Snowbird, Utah, Mar. 2003.

[19] J. A. Storer and J. H. Reif, "Error resilient optimal data compression," *SIAM J. Computing*, vol. 26, no. 4, pp. 934–949, Aug. 1997.

[20] S. Lonardi, W. Szpankowski, and M. Ward, "Error resilient LZ'77 data compression: Algorithms, analysis, and experiments," *IEEE Trans. on Inform. Theory*, vol. 53, no. 5, pp. 1799 – 1813, May 2007.

[21] K. P. Subbalakshmi and J. Vaisey, "On the joint source-channel decoding of variable-length encoded sources: the BSC case," *IEEE Trans. on Commun.*, vol. 49, no. 2, pp. 2052–2055, Dec. 2001.

[22] J. Garcia-Frias and J. D. Villasenor, "Turbo decoding of hidden Markov sources with unknown parameters," in *Proc. IEEE Data Compression Conference*, Snowbird, Utah, Apr. 1998.

[23] J. Garcia-Frias and W. Zhong, "LDPC codes for compression of multi-terminal sources with hidden Markov correlation," *IEEE Commun. Letters*, vol. 7, no. 3, pp. 115–117, Mar. 2003.

[24] Y. Zhao and J. Garcia-Frias, "Joint estimation and compression of correlated nonbinary sources using punctured Turbo codes," *IEEE Trans. on Commun.*, vol. 53, no. 3, pp. 385– 390, Mar. 2005.

[25] ——, "Joint estimation and data compression of correlated non-binary sources using punctured Turbo codes," in *Proc. Conf. Information Sciences and Systems*, Princeton, NJ, Mar. 2002.

[26] G.-C. Zhu and F. Alajaji, "Turbo codes for nonuniform memoryless sources over noisy channels," *IEEE Commun. Letters*, vol. 6, no. 2, pp. 64–66, Feb. 2002.

[27] ——, "Design of turbo codes for non-equiprobable memoryless sources," in *Proc. 39th Annual Allerton Conf. on Commun., Computing and Control*, Monticello, IL, Oct. 2001.

[28] G.-C. Zhu, F. Alajaji, J. Bajcsy, and P. Mitran, "Non-systematic turbo codes for non-uniform I.I.D sources over AWGN channels," in *Proc. Conf. Information Sciences and Systems*, Princeton, NJ, Mar. 2002.

[29] ——, "Transmission of nonuniform memoryless sources via nonsystematic turbo codes," *IEEE Trans. on Commun.*, vol. 52, no. 8, pp. 1344–1354, Aug. 2004.

[30] Y. Zhong, F. Alajaji, and L. L. Campbell, "On the joint source-channel coding error exponent for discrete memoryless systems," *IEEE Trans. on Inform. Theory*, vol. 52, no. 4, pp. 1450–1468, Apr 2006.

[31] G. Caire, S. Shamai, and S. Verdú, "Almost-noiseless joint source-channel coding-decoding of sources with memory," in *Proc. 5th Int. ITG Conf. on Source and Channel Coding*, Munich, Germany, Jan. 2004.

[32] T. Weissman, E. Ordentlich, G. Seroussi, S. Verdú, and M. Weinberger, "Universal discrete denoising: Known channel," *IEEE Trans. on Inform. Theory*, vol. 47, no. 1, pp. 5–28, Jan. 2003.

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

[34] A. Orlitsky and K. Viswanathan, "One-way communication and error-correcting codes," *IEEE Trans. on Inform. Theory*, vol. 49, no. 7, pp. 1781 – 1788, Jul. 2003.

[35] J. Bajcsy and P. Mitran, "Coding for the Slepian-Wolf problem with turbo codes," in *Proc. IEEE Global Commun. Conf.*, San Antonio, Tx., Nov. 2001.

[36] A. Aaron and B. Girod, "Compression with side information using turbo codes," in *Proc. IEEE Data Compression Conference*, Snowbird, Utah, Apr. 2002.

[37] A. D. Liveris, Z. Xiong, and C. N. Georghiades, "Compression of binary sources with side information at the decoder using LDPC codes," *IEEE Commun. Letters*, vol. 6, no. 10, pp. 440–442, Oct. 2002.

[38] J. Garcia-Frias, W. Zhong, and Y. Zhao, "Turbo-like codes for source and joint source-channel coding," in *Proc. 3rd International Symposium On Turbo Codes and Related Topics*, Sep. 2003.

[39] J. Garcia-Frias and Y. Zhao, “Compression of binary memoryless sources using punctured turbo codes,” *IEEE Commun. Letters*, vol. 6, no. 9, pp. 394–396, Sep. 2002.

[40] K. Bhattad and K. R. Narayanan, “A decision feedback based scheme for Slepian-Wolf coding of sources with hidden Markov correlation,” *IEEE Commun. Letters*, vol. 10, no. 5, pp. 378–380, May 2006.

[41] J. Garcia-Frias and Y. Zhao, “Data compression of unknown single and correlated binary sources using punctured turbo codes,” in *Proc. 39th Annual Allerton Conf. on Commun., Computing and Control*, Monticello, IL, Oct. 2001.

[42] J. Haghighat, M. R. Soleymani, and W. Hamouda, “Code detection in turbo source coding,” *IEEE Commun. Letters*, vol. 10, no. 4, pp. 225–227, Apr. 2006.

[43] J. Haghighat, W. Hamouda, and M. R. Soleymani, “Design of lossless turbo source encoders,” *IEEE Signal Processing Letters*, vol. 13, no. 8, pp. 453–456, Aug. 2006.

[44] J. Hagenauer, A. Barros, and A. Schaefer, “Lossless Turbo source coding with decremental redundancy,” in *Proc. 5th Int. ITG Conf. on Source and Channel Coding*, Munich, Germany, Jan. 2004.

[45] T. Ancheta, “Syndrome source coding and its universal generalization,” *IEEE Trans. on Inform. Theory*, vol. 22, no. 4, pp. 432 – 436, Jul. 1976.

[46] T. Richardson and R. Urbanke, *Modern Coding Theory*. Cambridge, UK: Cambridge University Press, 2008.

[47] G. Caire, S. Shamai, and S. Verdú, “Noiseless data compression with low-density parity-check codes,” in *Advances in Network Inform. Theory*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, American Mathematical Society, Piscataway, NJ, 2004.

[48] M. Burrows and D. J. Wheeler, “A block-sorting lossless data compression algorithm,” System Research Center 124, Tech. Rep., 1994.

[49] M. Effros, K. Visweswariah, S. Kulkarni, and S. Verdú, “Data compression based on the Burrows-Wheeler transform: Analysis and optimality,” *IEEE Trans. on Inform. Theory*, vol. 48, no. 5, pp. 1061–1081, May 2002.

[50] G. Caire, S. Shamai, and S. Verdú, “Universal data compression with LDPC codes,” in *Proc. 3rd International Symposium On Turbo Codes and Related Topics*, Brest, France, Sep. 2003.

[51] M. Fresia, F. Pérez-Cruz, and H. V. Poor, “Optimized concatenated LDPC codes for joint source-channel coding,” in *Proc. IEEE Int. Symp. on Inform. Theory*, Seoul, Korea, 2009.

[52] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. Spielman, “Analysis of low-density codes and improved designs using irregular graphs,” in *Proc. 30th ACM Symp. Theory of Computing*, Dallas, TX, 1998.

[53] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, “Design of capacity-approaching irregular low-density parity-check codes,” *IEEE Trans. on Inform. Theory*, vol. 47, pp. 619–637, Feb. 2001.

[54] S. T. Brink, “Designing iterative decoding schemes with the extrinsic information transfer chart,” *AEU Int. J. Electron. Commun.*, vol. 54, pp. 389–398, Dec. 2000.

[55] R. Urbanke, “http://lthcwww.epfl.ch/research/ldpcopt/,” 2002.

[56] S.-Y. Chung, T. J. Richardson, and R. Urbanke, “Analysis of sum-product decoding of low-density-parity-check codes using Gaussian approximation,” *IEEE Trans. on Inform. Theory*, vol. 47, no. 2, pp. 657–670, Feb. 2001.

[57] D. J. C. MacKay, *Information Theory, Inference and Learning Algorithms*. Cambridge, UK: Cambridge University Press, 2003.