Rateless Slepian-Wolf Codes

Andrew W. Eckford and Wei Yu
The Edward S. Rogers Sr. Department of ECE, University of Toronto
Email: andrew.eckford@utoronto.ca, weiyu@comm.utoronto.ca

Abstract— **The design and optimization of rateless codes for Slepian-Wolf encoding are considered. Rateless codes are proposed to address two shortcomings of currently available Slepian-Wolf schemes: their fragility to changing source statistics, and their inability to guarantee successful decoding for practical block length. We propose a novel type of optimized rateless code, called a *Matrioshka code*, to deal with the particular conditions of Slepian-Wolf encoding.**

I. Introduction

The elegant Slepian-Wolf theorem [1] demonstrates that lossless encoding of two correlated sources is possible at a rate equal to their joint entropy, even without communication between the two encoders. Recent surge of interests in this result is fueled in part by its relevance to the design of sensor networks, and in part by the discovery of a practical method known as distributed source coding using syndromes (DISCUS) [2] that uses linear error-correcting codes to achieve the joint entropy. The invention of DISCUS allows the excellent performance of very powerful codes, such as LDPC codes, to be exploited for Slepian-Wolf coding (for example, see [3]–[6]).

Although DISCUS and LDPC codes have made great strides towards the practical use of Slepian-Wolf coding, there remain two serious obstacles to widespread adoption. First, Slepian-Wolf coding is intended to be lossless, but for practical block lengths, LDPC codes cannot guarantee arbitrarily low probability of error. As a result, there is always a significant probability that the Slepian-Wolf decoding will fail, which is not acceptable. Second, a particular LDPC-based Slepian-Wolf code is always designed at a fixed rate for a particular source with a fixed correlation. When used with a different source, the LDPC code will either fail (if the source requires a higher rate) or will be inefficient (if the source requires a lower rate).

Analogous problems are found in channel coding, for which one proposal has been *rateless codes*, in which the rate can be adjusted on demand. In general for rateless codes, the encoder transmits a partial codeword until the decoder has enough information to decode, at which time an ACK message is sent; in this sense, rateless codes are related to hybrid ARQ systems. The first practical rateless code was the Luby Transform (LT) code, which is similar to an LDPC code, and intended for the binary erasure channel [7]. It can be shown that this code, if properly implemented, is *universal* for the erasure channel, i.e., it can achieve the capacity for any erasure probability. More recently, the lower-complexity Raptor codes [8] were introduced, and recent work has shown that these codes are effective in channels with symmetric noise [9]–[11].

We are motivated by the success of rateless channel codes and aim to design similar rateless Slepian-Wolf codes for sources with varying correlation. Although conceptually similar, rateless source coding and rateless channel coding are very different from a code design point of view. In the channel coding setting, coded bits are generated from the information sequence and transmitted through the noisy channel. The objective is to decode the information sequence based on noise-corrupted coded bits. In the source coding setting, coded bits are generated from the source, but transmitted noiselessly to the decoder which also has a correlated sequence from another encoder available to it. The objective is to recover the source based on the coded bits and the correlated sequence. To contrast the two problems, in the channel coding scenario, the noise process corrupts the coded bits directly; in the source coding scenario, the noise process corrupts the source bits, resulting a different coding design problem.

The result in this paper is inspired by [12], which shows that rateless Slepian-Wolf codes are fundamentally possible. The setup of [12] is as follows: the Slepian-Wolf encoders transmitted their information to the decoder until they had either transmitted their entire codeword, or until they received an ACK message from the decoder. The ACK message was sent once the decoder had enough information to correctly decode all the transmitted messages. Practically speaking, this is a very appealing framework, as it essentially guarantees successful decoding. Furthermore, assuming that the decoder has access to much greater resources than the encoders, such a simple "one-bit" feedback channel would be easy to implement with little added complexity at the encoders.

The main contribution of this paper is a novel family of layered LDPC-like codes that are capable of achieving the joint entropy in the one-bit feedback setup described above. We call such codes *Matrioshka codes* (since they can be thought of as a set of LDPC codes that nest inside one another). These codes function at multiple rates, and do not require *a priori* knowledge of the channel. Furthermore, a given Matrioshka code, which succeeds in a particular set of channels, is based on a single LDPC degree sequence, making the code design problem easier to handle.

The universal distributed source coding problem considered in this paper is also related to the fountain-code-based universal single-source compressors proposed in [13]. However, there are also clear differences. First, a crucial assumption of [13] is that the encoder could verify, before transmitting, whether the decoder could successfully decode. Since LT codes are randomly generated, this assumption allows the method in [13] to try several LT codes to find one that has a small number of output bits. Furthermore, the authors did not optimize the fountain code degree sequence. Instead, they selected a degree sequence optimized for the binary erasure channel. In this paper, as in [12], successful decoding is only verified at the end of transmission, and the degree sequence is carefully designed. Second, Matrioshka codes use an approach that is different from LT or Raptor codes. As mentioned earlier, LT-like codes are not suitable for the Slepian-Wolf problem, as they are intended to produce a large number of coded bits to be transmitted through a noisy channel. On the other hand, in a Slepian-Wolf source coding problem, the encoder transmits encoded source symbols through a noiseless channel. Matrioshka codes explicitly avoid the use of LT codes.

The remainder of the paper is organized as follows. In Section II, we introduce the source model that is used throughout the rest of the paper, and discuss Slepian-Wolf encoding. In Section III, we discuss our method, and introduce Matrioshka codes. In Section IV, we give preliminary results found by our techniques.

II. Model

Let $x \in \{0,1\}^n$ and $y \in \{0,1\}^n$ represent binary sources observed at different Slepian-Wolf encoders (expressed as row vectors), where x_i and x_j (resp., y_i and y_j) are independent for all $i \neq j$. We assume that the source is memoryless, and that the marginal distributions are equiprobable, i.e., $p(x_i) = p(y_i) = 1/2$ for all x_i, y_i. Furthermore, we suppose that there exists a Bernoulli random sequence $z \in \{0,1\}^n$ such that $y_i = x_i \oplus z_i$ for all i. It is easy to see that this source is characterized by a single parameter, p, where $p := \mathsf{Pr}(Z_i = 1)$. For this source, the corners of the Slepian-Wolf rate region occur at

$$\begin{aligned} (H(X), H(Y|X)) &= (1, H(Z)), \text{ and} \\ (H(X|Y), H(Y)) &= (H(Z), 1); \end{aligned}$$

where $H(Z)$ is the entropy of the difference sequence z. Similarly to the corresponding channel, a pair of correlated sources with these properties are referred to as a *binary symmetric source*, and p is referred to as the *crossover probability* of the source.

Slepian-Wolf encoding of a source is accomplished most simply at a corner point, using the DISCUS technique [2]. For example, to encode a source at a corner point with rates $(H(X|Y), H(Y))$, the encoder corresponding to source y encodes its source independently at a rate of $H(Y)$, meaning that the decoder can recover y without knowing anything about x. Now, we have y available to the decoder, and under our source assumption, we know that $y_i = x_i \oplus z_i$ for all i, so recovering x is equivalent to the channel-coding problem of recovering transmitted x in the presence of noise z. For a linear code C with parity-check matrix $\mathbf{H}$, the sequence x is in a coset code $C^{(x)}$ in which all the codewords have the syndrome $s = \mathbf{H}x$. If s is sent to the decoder, then the decoder can decode the "observation" y with respect to the coset code $C^{(x)}$, which recovers x.

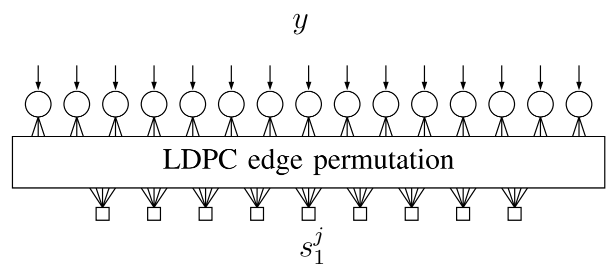

Fig. 1. LDPC-based Slepian-Wolf decoders at a corner point (top), and a non-corner point (bottom).

An LDPC code is a good choice for the linear code, since capacity-approaching channel codes are also entropy-approaching source codes. The factor graph for an LDPC-based Slepian-Wolf decoder is given in Fig. 1. For traditional LDPC-based Slepian-Wolf encoders, the rate is fixed at design time, and even for the same rate, the LDPC codes are generally optimized for particular source statistics. Thus, an encoder applied to a source other than the design source will usually be inefficient or useless. Our objective, which will be expanded upon in the next section, is to obtain a family of LDPC codes that are simple to optimize and that retain their good performance as the source statistics change.

III. Multi-rate Slepian-Wolf codes

A. At a corner point: General remarks

We consider the case of encoding and decoding at a corner point for a binary symmetric source, where the component sources are labelled x and y. In this case, since the two sources are related by $y_i = x_i \oplus z_i$, y acts like an observation of x through a binary symmetric channel with noise sequence z. We assume that y is available at the decoder before any of the encoded bits for x are transmitted.

Encoding. As in Fig. 1, the source y is known at the decoder, while x is encoded as the syndrome of an LDPC code. Let $s_1^j := \{s_1, s_2, \ldots, s_j\}$ represent the syndrome vector for the source x. Then s_1^j is the string generated by the encoder for x. As noted previously, the source y is encoded independently of x and transmitted to the receiver at rate 1 (which, as we have assumed, is the entropy of y).

Decoding. After s_1^j has been received, the decoder knows that the source string x is in a coset of the original LDPC code with s_1^j as the syndrome of the coset, and that y is an observation of this codeword through binary symmetric noise. The decoding problem is equivalent to syndrome decoding for LDPC codes, and has been discussed in [6], [13]. The sum-product algorithm may be used, with one simple modification: at any check node with odd parity, all outgoing messages are multiplied by -1. It is straightforward to show that decoding performance of a coset of an LDPC code is equivalent to the decoding performance of the original (zero-syndrome) LDPC code.

B. Revealing sources: Definition and coding

We seek an LDPC code whose rate can be modified for different sources without damaging its ability to decode at a rate close to the source's entropy. Toward this end, we consider a related, but different, Slepian-Wolf coding problem: the design of a single, universal code for a class of sources with different source statistics, but with the same joint entropy. In particular, consider a *revealing source*, where x_i and y_i are source symbols with the following properties:

- With probability b, $y_i = x_i \oplus z_i$, where z_i is a Bernoulli random variable with crossover probability p;
- With probability $1 - b$, $y_i = x_i$ (i.e., the symbol is *revealed*); and
- The decoder knows whether the symbol is revealed or not.

In this section, we will show that an LDPC code that works well for the revealing source can be used to construct a multi-rate Slepian-Wolf code.

The set of revealing sources are parameterized by the pair (b, p). and it is easy to show that the conditional entropy of a source with these parameters is given by

$$H(Y|X) = b\mathcal{H}(p), \tag{1}$$

where $\mathcal{H}(p)$ is the binary entropy function. Consider the set $\mathbb{S}_H$ of revealing sources, defined as

$$\mathbb{S}_H = \{(b, p) : b\mathcal{H}(p) = H\},$$

that is, the set of all sources with conditional entropy equal to H. For the family $\mathbb{S}_H$, there is one source where $b = 1$ (i.e., none of the symbols are revealed). We define p^* as the corresponding crossover probability, so $(1, p^*) \in \mathbb{S}_H$, and $\mathcal{H}(p^*) = H$. Some useful properties of revealing sources are given as follows:

- $(H, 1/2) \in \mathbb{S}_H$;
- For all $(b, p) \in \mathbb{S}_H$, $p^* \le p \le 1/2$ and $H \le b \le 1$; and
- For any p where $p^* \le p \le 1/2$, $(H/\mathcal{H}(p), p) \in \mathbb{S}_H$.

The key observation of this paper is the following. Suppose that it is possible to design an LDPC code with rate $R > H$ and degree sequence (λ, ρ) that decodes successfully for every source in the family $\mathbb{S}_H$. Then this single LDPC code must contain subcodes that would decode successfully for binary symmetric channels with every crossover probability $p > p^*$. The reason is as follows. In decoding an LDPC code of length n for a source with $b < 1$, we expect a fraction $n(1 - b)$ of symbols to be revealed. Symbols that are revealed are perfectly known at the encoder can be removed from the LDPC decoder's factor graph, with no effect on sum-product decoding (in fact, this is one step in sum-product decoding for an LDPC code in the erasure channel). Thus, we are left with a shorter (and higher-rate) LDPC code, having nb variables. If the LDPC code of length n successfully decodes in the revealing source with parameters (b, p), this shorter LDPC code must also decode successfully in the binary symmetric source with crossover probability p, where $p > p^*$. This is illustrated in Fig. 2. Thus, the problem of finding a universal code is reduced to the design of a single degree sequence.

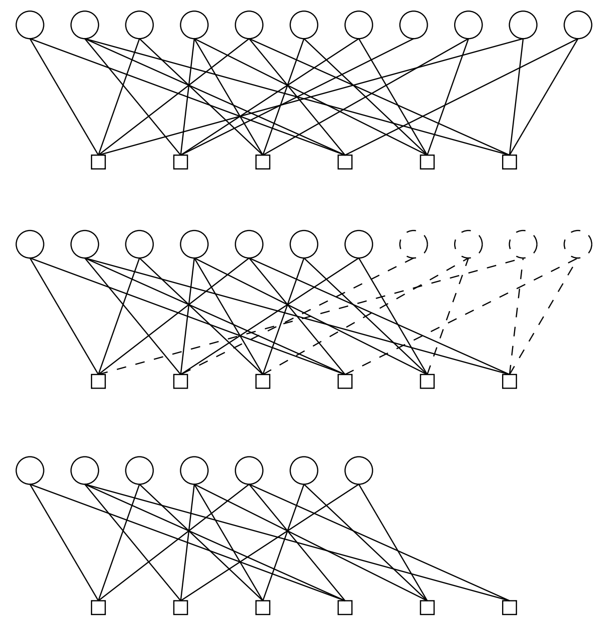

Fig. 2. LDPC codes in a revealing source. Top: The full code, when $b = 1$. Middle: $b < 1$, and the last four symbols are revealed. Bottom: The code when the last four symbols are removed, which can be used for inversion probability p.

Since the rate of the original LDPC code is R, the rate of the shorter code is R/b. For an LDPC code with rate R, define

$$\alpha := R/H$$

as the gap to the entropy, where $\alpha \ge 1$ for any code that decodes successfully. From (1) and the definition of $\mathbb{S}_H$, we have that $H = \mathcal{H}(p^*) = b\mathcal{H}(p)$. Thus, $R = \alpha\mathcal{H}(p^*)$, and $R/b = \alpha b\mathcal{H}(p)/b = \alpha\mathcal{H}(p)$. Thus, the gap to entropy is the same for both codes.

C. Layered codes

The previous section on revealing sources took a single LDPC code and broke it into a family of codes that could be used with sources having many crossover probabilities. In this section, we show how to layer these codes to obtain a multi-rate code for Slepian-Wolf encoding.

Suppose we have a source with two possible crossover probabilities: p_1 and p_2, where $p_2 > p_1$ without loss of generality. Our strategy, inspired by [12], is to transmit parity checks initially under the assumption that the crossover probability is p_1. If no acknowledgment is received, we will continue transmitting parity checks under the assumption that the crossover probability is p_2, since that is the only alternative. We ignore the possibility of decoding errors.

Letting $H = \mathcal{H}(p_1)$, the family $\mathbb{S}_H$ contains the sources $(1, p_1)$ and $(H/\mathcal{H}(p_2), p_2)$. Our approach is to transmit a code that decodes successfully for all members of $\mathbb{S}_H$. If no acknowledgment is received, we select $1 - H/\mathcal{H}(p_2)$ (i.e., $1-b$) source symbols and form parity checks using a code that

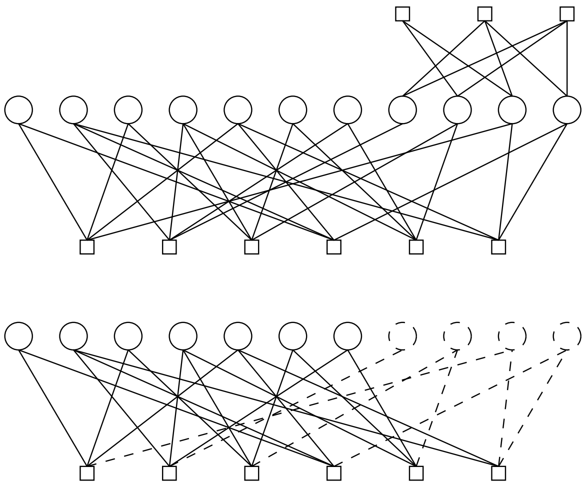

Fig. 3. Two concatenated LDPC codes. Top: New parity checks succeed with crossover probability p_2. Bottom: When the new checks are decoded, enough symbols are revealed in the original code to permit decoding.

decodes successfully for a source with crossover probability p_2, which can be obtained using the method in the previous section. At the decoder, we have two codes protecting the same symbols. The decoding strategy is *serial*: we decode the second code (which succeeds at crossover probability p_2) first, which *reveals* the $1-H/\mathcal{H}(p_2)$ symbols to which it is attached, and then decode the first code. Since the first code succeeds for any revealing source in $\mathbb{S}_H$, and since $(H/\mathcal{H}(p_2), p_2) \in \mathbb{S}_H$, decoding is successful. This approach is illustrated in Fig. 3.

We now consider the efficiency of this scheme. Once again, let $R_1 = \alpha\mathcal{H}(p_1)$ for the initial code. Since the second code comes from the same family, its rate is $R_2 = \alpha\mathcal{H}(p_2)$, as argued in the previous section. For an LDPC source code, rate is given by the ratio of parity checks to variables, so letting m_i and n_i represent the number of parity checks and variables in the ith code, respectively, we have $\alpha\mathcal{H}(p_1) = m_1/n_1$ and $\alpha\mathcal{H}(p_2) = m_2/n_2$. The overall rate of the concatenated code is given by $R_c = (m_1 + m_2)/n_1$, since all the parity checks of both codes are transmitted. Thus,

$$R_c = \frac{\alpha\mathcal{H}(p_2)n_2 + \alpha\mathcal{H}(p_1)n_1}{n_1}.$$

However, $n_2 = (1 - H/\mathcal{H}(p_2))n_1$, so

$$\begin{aligned} R_c &= \frac{\alpha\mathcal{H}(p_2)(1 - H/\mathcal{H}(p_2))n_1 + \alpha\mathcal{H}(p_1)n_1}{n_1} \\ &= \alpha\mathcal{H}(p_2), \end{aligned}$$

so the scheme has the same gap to entropy as the original code family. In particular, if $\alpha \to 1$, the scheme achieves entropy.

The same technique can be applied if the source has three or more possible crossover probabilities, with minor complications that we will not discuss in this paper. From Fig. 3, there is a nesting property to the codes, which leads us to coin the term *Matrioshka code*, after the Russian nesting dolls. It is somewhat surprising that the gap to entropy remains at α, which implies that no parity checks are "wasted" in protecting some symbols with multiple codes. Furthermore, this emphasizes our initial point concerning the ease of optimization, since *only the single parameter α needs to be minimized over a single LDPC degree sequence*.

D. From multi-rate to rateless

For the code family obtained in the previous section, it is more appropriate to call them multi-rate codes rather than rateless codes. The proposed serial decoding algorithm makes it necessary to wait for "chunks" of parity bits, representing the parity bits in each layer of the code, to arrive before attempting to decode. As a result, the allowed rates are quantized to a few values. However, the serial decoding algorithm is proposed only for convenience in describing the system. Since the entire code can be drawn on a single factor graph, as in the top diagram from Fig. 3, there is no reason in practice why the codes must be decoded serially. Indeed, simultaneous decoding of the code can only help, and may avoid problems of error propagation.

If simultaneous decoding is used, then as parity checks arrive at the decoder, the decoder may immediately incorporate them into its factor graph and immediately attempt to decode using that factor graph. Once enough parity checks have arrived so that decoding is successful, the decoder can transmit its acknowledgment to stop the encoder. However, since the code would be determined by the properties of the layered codes, the performance could only be guaranteed at the end of chunks.

IV. Optimization and Results

A. Optimization

For some family of revealing sources $\mathbb{S}_H$, we require the lowest rate source code that is capable of decoding successfully for every source in $\mathbb{S}_H$. The LDPC degree sequence (λ, ρ), which specifies the distribution of the check and variable degrees in the factor graph, must be optimized to minimize the source code rate (equivalent to maximizing the channel code rate) subject to successful decoding in $\mathbb{S}_H$. To accomplish this, we use a design tool from the literature that reduces the LDPC code design problem to linear programming [15]. The tool uses a form of extrinsic information transfer (EXIT) chart [16] that tracks the message probability of error from iteration to iteration. The key observation is that variable nodes of each possible degree induce their own EXIT chart (known as an *elemental EXIT chart*), and that the true probability of error is a linear combination of the elemental EXIT charts, where the coefficients are the elements of the variable degree sequence. Thus, the criterion to ensure successful decoding, which is that probability of error always decreases from iteration to iteration, can be formulated as a linear constraint. Furthermore, if the check degree sequence is fixed, maximization of the rate can be posed as a linear objective, so linear programming may be used.

The design tool from [15] is normally used to optimize LDPC codes with respect to a single source, but we quantize the parameter space and augment the constraints of linear programming to ensure successful decoding at each quantized point. Finally, once an optimized degree sequence is obtained, we verify successful decoding at each quantized point using density evolution [14], which has higher accuracy than the design tool.

For convenience in using the method in [15], our optimization focused on LDPC codes with a particular factor graph structure. For an LDPC code of length n, we partition the variables into two groups of lengths n_1 and n_2. When symbols are revealed, they are only revealed from the second group. The connections of the parity checks are constrained so that, for each check of degree d_c, at least $\lfloor (n_1/n)d_c \rfloor$ edges are connected to the first group, and at least $\lfloor (n_2/n)d_c \rfloor$ edges are connected to the second group. We emphasize that there is nothing mandatory about this factor graph structure, and that we propose it solely for ease of use with the design tool in [15]. Randomly connected LDPC factor graphs should be no worse in performance.

B. Results

Using the optimization method outlined in the previous section, we obtained an LDPC code for the family $\mathbb{S}_H$, where $H = 0.4541$, corresponding to $p^* = 0.0953$. Before revealing (i.e., $b = 1$), the degree sequence for this LDPC code is given by

$$\begin{aligned} \lambda_2 &= 0.1763, \\ \lambda_3 &= 0.3943, \\ \lambda_{10} &= 0.3206, \\ \lambda_{50} &= 0.1088; \\ \rho_8 &= 1, \end{aligned}$$

and zero for all other entries. After revealing with $b = 0.5652$, and removing all the revealed symbols, the variable degree sequence λ is the same, while the new check degree sequence ρ is given by

$$\begin{aligned} \rho_4 &= 0.4232, \\ \rho_5 &= 0.5768, \end{aligned}$$

and zero for all other entries. The rate of the unrevealed code is $R = 0.4925$, so the gap to entropy is $\alpha = 1.0845$ (i.e., the code is within 8.5% of entropy).

Using density evolution, this code was tested for revealing sources over the range of parameters p in which $0.0953 \leq p \leq 0.2451$, corresponding to a range of entropies from $0.4541 \leq \mathcal{H}(p) \leq 0.8034$. Simulation results for the extreme points of this range are given in Fig. 4, in which the code length was $n = 20001$, and the results show good agreement with the predictions of density evolution.

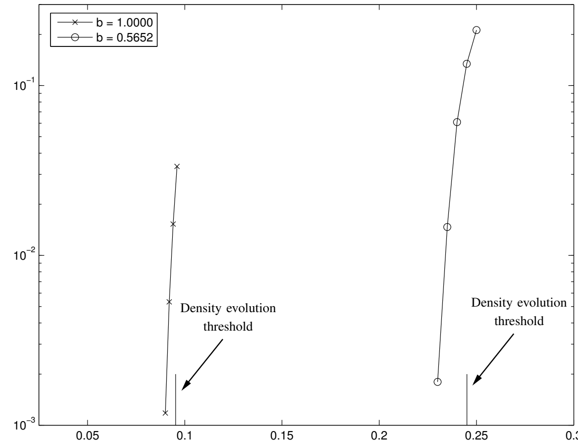

Fig. 4. Experimental results for the given code, for sources $(1, 0.0953)$ and $(0.5652, 0.2451)$.

V. Conclusion

This paper describes Matrioshka codes, which are LDPC-like codes with variable rates, as a rateless solution to the Slepian-Wolf problem. These codes have the attractive property of being easy to optimize and analyze, since they are obtained from a single LDPC degree sequence. Experimental results indicate that these codes have good performance for sources with a wide range of joint entropies.

References

[1] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inform. Theory*, vol. 19, no. 4, pp. 471-480, Jul. 1973.

[2] S. S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): Design and construction," *IEEE Trans. Inform. Theory*, vol. 49, no. 3, pp. 626-643, Mar. 2003.

[3] D. Schonberg, K. Ramchandran, and S. S. Pradhan, "LDPC codes can approach the Slepian Wolf bound for general binary sources," in *Proc. 40th Allerton Conf. on Commun., Control, and Computing*, 2002.

[4] 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.

[5] T. P. Coleman, A. H. Lee, M. Médard, and M. Effros, "On some new approaches to practical Slepian-Wolf compression inspired by channel coding," in *Proc. Data Compression Conference*, 2004.

[6] A. W. Eckford and W. Yu, "Density evolution for the simultaneous decoding of LDPC-based Slepian-Wolf source codes," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, Adelaide, Australia, 2005.

[7] M. Luby, "LT codes," in *Proc. 43rd Annual IEEE Symposium on Foundations of Computer Science*, 2002.

[8] M. A. Shokrollahi, "Raptor codes," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, 2004.

[9] R. Palanki and J. S. Yedidia, "Rateless codes on noisy channels," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, 2004.

[10] O. Etesami, M. Molkaraie, and M. A. Shokrollahi, "Raptor codes on symmetric channels," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, 2004.

[11] J. Castura and Y. Mao, "Rateless coding over fading channels," in *Proc. Canadian Workshop on Information Theory*, 2005.

[12] S. Draper, "Universal incremental Slepian-Wolf coding," in *Proc. 42nd Annual Allerton Conference on Communications, Control, and Computing*, 2004.

[13] G. Caire, S. Shamai, A. Shokrollahi, and S. Verdú, "Universal variable-length data compression of binary sources using fountain codes," in *Proc. IEEE Information Theory Workshop (ITW)*, 2004.

[14] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 599-618, Feb. 2001.

[15] M. Ardakani and F. R. Kschischang, "A more accurate one-dimensional analysis and design of irregular LDPC codes," *IEEE Trans. Commun.*, vol. 52, no. 12, pp. 2106-2114, Dec. 2004.

[16] S. ten Brink, "Convergence of iterative decoding," *Electronics letters*, vol. 35, no. 10, pp. 806-808, May 1999.