An Efficient SF-ISF Approach for the Slepian-Wolf Source Coding Problem

Zhenyu Tu

Department of Electrical and Computer Engineering, Lehigh University, Bethlehem, PA 18105, USA
Email: zht3@ece.lehigh.edu

Jing Li (Tiffany)

Department of Electrical and Computer Engineering, Lehigh University, Bethlehem, PA 18105, USA
Email: jingli@ece.lehigh.edu

Rick S. Blum

Department of Electrical and Computer Engineering, Lehigh University, Bethlehem, PA 18105, USA
Email: rblum@ece.lehigh.edu

A simple but powerful scheme exploiting the binning concept for asymmetric lossless distributed source coding is proposed. The novelty in the proposed scheme is the introduction of a *syndrome former* (SF) in the source encoder and an *inverse syndrome former* (ISF) in the source decoder to efficiently exploit an existing linear channel code without the need to modify the code structure or the decoding strategy. For most channel codes, the construction of SF-ISF pairs is a light task. For parallelly and serially concatenated codes and particularly parallel and serial turbo codes where this appears less obvious, an efficient way for constructing linear complexity SF-ISF pairs is demonstrated. It is shown that the proposed SF-ISF approach is simple, provenly optimal, and generally applicable to any linear channel code. Simulation using conventional and asymmetric turbo codes demonstrates a compression rate that is only 0.06 bit/symbol from the theoretical limit, which is among the best results reported so far.

Keywords and phrases: distributed source coding, compression with side information at the decoder, Slepian-Wolf coding, code binning, serially concatenated convolutional codes, parallelly concatenated convolutional codes.

1. INTRODUCTION

The challenging nature of multiuser communication problems [1] has been recognized for decades and many of these problems still remain unsolved. Among them is the distributed source coding (DSC) problem, also known as distributed compression or Slepian-Wolf source coding, where two or more statistically correlated information sources are separately encoded/compressed and jointly decoded/decompressed. Having its root in network information theory, distributed source coding is tightly related to a wealth of information and communication problems and applications including, for example, the dirty paper coding, watermarking and data mining, multielement broadcasting problem and multiple description coding. The recent heat in sensor networks has further aroused a renewed interest in DSC, since it allows the intersensor correlation to be exploited in compression without expensive intersensor communication.

The theory and conceptual underpinnings of the noiseless DSC problem started to appear back in the seventies [2, 3, 4, 5]. Specifically, the seminal paper by Slepian and Wolf [2] stated that (i) separate encoding (but joint decoding) need not incur a loss in capacity compared to joint encoding and (ii) the key to DSC lies in channel coding. These refreshing findings, as well as the underlying concept of code binning (will be discussed in Section 2), lay the foundation for practical code design for DSC using linear channel codes.

The random binning concept used in the proof of the Slepian-Wolf theorem requires structured binning implementations in practice. The first practical algebraic binning scheme was proposed by Wyner in 1976 [1], where the achievability of the Slepian-Wolf boundary was demonstrated using coset codes and a generic syndrome decoder. The approach was further extended to nonsyndrome decoders by Pradham and Ramchandram many years later [6]. Since then, various practical coding schemes have been proposed for lossless DSC with binary memoryless sources,

including coset codes [6], lattice codes [7, 8], low density parity check (LDPC) codes (e.g., [9, 10, 11, 12, 13, 14]) and (convolutional) turbo codes (e.g., [15, 16, 17, 18, 19, 20]). Most of these formulations are rooted back to the binning idea, except for turbo codes where code binning has not been explicitly exploited.

While LDPC codes are also capacity-approaching channel codes, turbo codes have certain advantages. First, a turbo encoder is cheap to implement, thus appealing to applications like sensor networks where the computation on the transmitter side (i.e., sensor nodes) needs to be minimized. Second, turbo codes perform remarkably on a variety of channel models. Since the key to efficient DSC is to find a powerful channel code for the *virtual transmission channel*, where the virtual channel is specified by the source correlation (will be discussed in more detail in Section 2), turbo codes are therefore a good choice for a number of sources with different source correlations. An LDPC code, on the other side, would require specific design or optimization of the degree profile in order for it to match to the channel. Third, the code rate and length of a turbo code can be easily changed (e.g., through puncturing), making it possible for *adaptive* DSC using rate compatible turbo codes. Such flexibility is not readily available with random LDPC codes or other linear block codes.

Among the existing turbo DSC formulations, Garcia-Frias and Zhao were the first to propose an interesting turbo scheme where two sources were separately encoded and jointly decoded in an interwoven way akin to a four-branch turbo code [15]. A similar scheme that works for asymmetric compression was independently devised by Aaron and Girod [16]. In [17], Bajcsy and Mitran proposed yet another parallel turbo structure based on finite-state machine codes. The scheme was later extended to a serial turbo structure in [19]. Perhaps the only scheme that has *implicitly* explored the binning concept is that proposed by Liveris, Xiong, and Georghiades [18]. This also appears to be the only provenly optimal DSC scheme based on turbo codes.

One major reason why the binning approach has not been popular with turbo codes lies in the difficulty of constructing bins for turbo codes. While codewords are easily "binned" for coset codes and block codes (e.g., via the parity check matrix), the random interleaver in the turbo code makes the code space intractable, precluding the possibility to spell out its parity check matrix. Another reason that has possibly prevented the full exploitation of the binning idea is the lack of a general source decoding approach. In theory, only a codebook that specifies the mapping (e.g., the bins) is needed; in practice, a practically implementable source encoder and particularly a practically implementable source decoder are also needed. The latter, however, has not been well studied except for LDPC codes. We note that for LDPC codes, due to the unique characteristics in the code structure and the decoding algorithm, a syndrome sequence (i.e., the compressed sequence, see Section 2) can be easily incorporated in the message-passing decoding, making source decoding a natural extension of channel decoding [9, 10, 11, 12, 13]. However, for many other codes including turbo codes, it has not been entirely clear how to optimally exploit a syndrome sequence in the decoding approach.

The purpose of this paper is to investigate asymmetric DSC using the binning idea for binary linear channel codes in general, and parallel and serial turbo codes in particular. The focus is on the code design for practical DSC solutions that are efficient, optimal, and general. Our contributions are summarized as follows.

(1) We present the structure of a pair of universal source encoder and source decoder that are generally applicable to any linear channel code. While the idea is implicit in the binning concept [2, 8], we give an explicit presentation with a rigorous proof of its validity for binary memoryless sources. As will be discussed in Section 3, the proposed source encoder and source decoder explore the concept of *syndrome former* (SF) and *inverse syndrome former* (ISF), and are efficient as well as provenly optimal for binary memoryless sources. This thus represents a simple and universal framework that allows an existing powerful linear channel code to be readily exploited in DSC without the burden of redesigning the code or finding a matching encoding/decoding strategy. With this framework, the only task that is left to implement the DSC solution is to construct a valid SF-ISF pair, which, for many channel codes, are a pretty light and straightforward task.

(2) For parallelly and serially concatenated codes (PCC/SCC) where the SF-ISF construction appears tricky due to the random interleaver, we demonstrate an efficient and systematic way to handle it. Instead of deriving the SF-ISF pair in an overall closed form (which seems to pose unsolvable complexity problems), the proposed construction cleverly exploits the sub-SFs and sub-ISFs of the component codes in a way similar to the way concatenated code is built from its component codes [20]. The SF-ISF pairs for both parallelly and serially concatenated codes have a complexity of the order of that of the component codes, and can be conveniently implemented using linear sequential circuits. For illustrative purpose, the discussion will proceed with parallelly and serially concatenated convolutional codes (PCCC/SCCC), or parallel and serial turbo codes, as the illustrating example. However, the applicability of the proposed method goes beyond the context of concatenated convolutional codes. As addressed in Section 5, other concatenated structures, including block turbo codes (BTC) [21], can readily adopt the same SF-ISF formulation.

(3) Through the proposition of the SF-ISF formulation and the general source encoder/decoder structure, we have demonstrated the first provenly optimal turbo-DSC formulation that *explicitly* exploits the binning scheme. Compared to the approach in [22], which is also provenly optimal but which requires constructing a source encoding trellis with parallel branches, a source decoding trellis with time-varying stages, and a matching (time-varying) decoding algorithm, the proposed one is simpler and more general.

(4) One goal of our work is to come close to the theoretical limit. We show, through simulations on conventional turbo codes and asymmetric turbo codes [23], that the proposed SF-ISF based scheme yields a compression rate as close
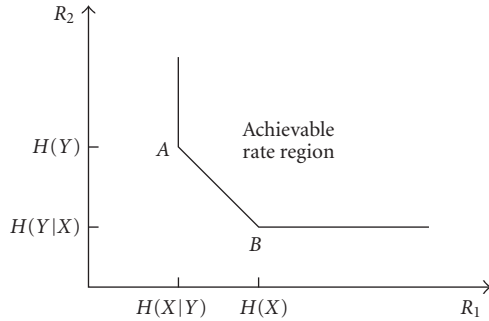
FIGURE 1: Rate region for noiseless DSC.

FIGURE 2: Asymmetric DSC can be equivalently viewed as a channel coding problem with side information at decoder.

as 0.06 bit/symbol from the theoretical limit for binary symmetric sources (BSS), which is among the best results reported so far.

The remainder of the paper is organized as follows. Section 2 formulates the DSC problem and introduces the binning concept. Section 3 presents the structure of a universal source encoder and a source decoder with a rigorous proof of its validity. Section 4 discusses in detail the construction for SF-ISF pairs for parallelly and serially concatenated codes and in particular parallel and serial turbo codes. Section 5 and 6 discuss the optimality and performance of the proposed SF-ISF approach for binary symmetric sources. Finally, Section 7 provides the concluding remarks.

2. BACKGROUND

2.1. Achievable rate region for DSC

We first formulate the setting for discussion. Consider two correlated binary memoryless sources X and Y encoded by separate encoders and decoded by a joint decoder. The achievable rate region is given by the Slepian-Wolf boundary [2]:

$$R_1 \geq H(X|Y),$$
$$R_2 \geq H(Y|X), \qquad (1)$$
$$R_1 + R_2 \geq H(X,Y),$$

where R_1 and R_2 are the compression rates for sources X and Y, respectively. A typical illustration is given in Figure 1.

For most cases of practical interest, zero-error DSC is possible only asymptotically [24]. For discrete memoryless sources of uniform distribution, corner points on the Slepian-Wolf boundary can be achieved by considering one source (e.g., Y) as the side information (SI) to the decoder (e.g., available to the decoder via a conventional entropy compression method) and compressing the other (i.e., X) to its conditional entropy ($H(X|Y)$). This is known as asymmetric compression (see Figure 2). The line connecting the corner points can be achieved through time sharing or code partitioning [12, 13]. (Unless otherwise stated, the discussion in the sequel focuses on binary sources and all the arithmetics are taken in $GF(2)$.)
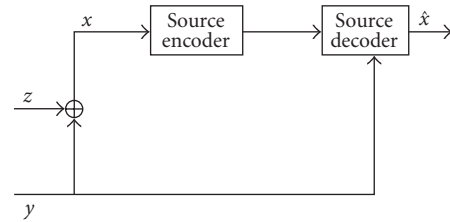
2.2. The binning concept

First introduced in [2], code binning is one of the most important ideas in distributed source coding. A thorough discussion on the binning concept and related issues can be found in [8]. Below, we provide a concise summary of this useful concept.

As the name suggests, the fundamental idea about code binning is to group sequences into bins subject to certain requirements or constraints. The information-theoretical justification for the idea is to use $2^{nH(X,Y)}$ jointly typical sequences to describe sources (X^n, Y^n), where the sequences are placed in $2^{nH(X|Y)}$ disjoint bins each containing $2^{nH(Y)}$ sequences. Clearly, $nH(X|Y)$ bits are needed to specify a bin and $nH(Y)$ bits to specify a particular sequence in the bin. From the practical point of view regarding algorithmic design, code binning consists essentially of dividing the entire codeword space of a linear channel code into disjoint subspaces (i.e., bins) such that *the same distance property is preserved in each bin*. For an (n,k) binary linear channel code, source sequences of length n are viewed as the virtual codewords (not necessarily the valid codewords of the channel code). The entire codeword space, $X^n = \{0,1\}^n$, can be evenly divided into 2^{n-k} bins/cosets with codewords having the same syndrome grouped in the same bin. It can be easily verified that the distance requirement is satisfied due to the geometric uniformity of a linear channel code. Naturally, the 2^{n-k} syndrome sequences can be used to index the bins. Hence, by transmitting the length $n - k$ syndrome sequence S^{n-k} instead of the length n source sequence X^n, a compression rate of $n : (n - k)$ is achieved. At the decoder, the syndrome sequence S^{n-k} and the decoder side information Y^n (i.e., the other source Y^n which is viewed as a noisy version of X^n due to its correlation with X^n) will be used to identify the original data sequence. The binning concept as well as the practical binning approach using linear channel codes are illustrated in Figure 3.

It should be noted that, in order for (near) lossless recovery of the original source X^n, the compression rate needs to satisfy $k/(n - k) \geq H(X|Y)$. Further, to get close to the theoretical limit, the (n,k) channel code needs to be a capacity approaching one for the *virtual transmission channel*, where the virtual channel is specified by the source correlation $P(X, Y)$.
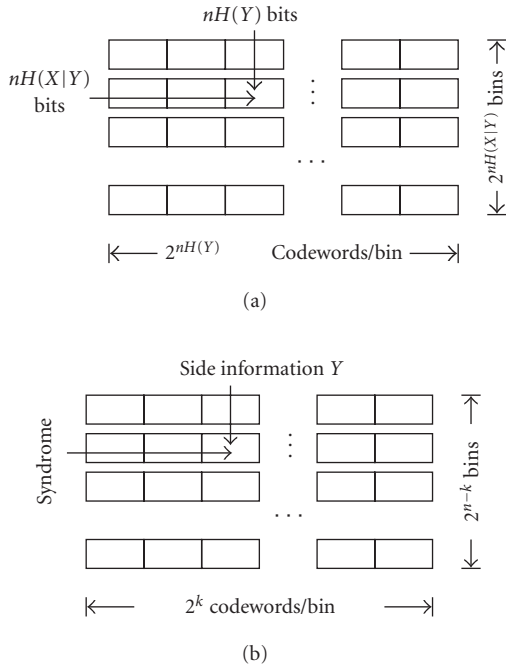
FIGURE 3: (a) Illustration of the binning concept. (b) Illustration of the algebraic binning approach using linear channel codes.

3. A UNIVERSAL SOURCE ENCODER AND SOURCE DECODER

The above binning concept has specified the codebook, that is, the mapping between the source sequences to the compressed sequences, but sheds little insight on the implementation of a source encoder and particularly a source decoder. Below, we present the structure of a universal source encoder and source decoder that practically and optimally implements the binning concept for memoryless binary symmetric sources [25].

Before we proceed, we first introduce the concept of syndrome former and inverse syndrome former, which are essentially functions that map the codeword space $\{X^n\}$ to the syndrome space $\{S^{n-k}\}$ and vice versa. Specifically, the role of the syndrome former is, for a given source sequence or a codeword in a bin, to find its associated syndrome sequence or bin index, and the role of the inverse syndrome former is, for a given syndrome sequence or a bin index, to find an *arbitrary* source sequence that belongs to that particular coset or bin (we term the output of the ISF as the "auxiliary sequence"). It should be noted that the SF-ISF pair is not unique for a given (n, k) linear channel code. For a valid SF, that is, valid bin-index assignment, as long as the all-zero syndrome sequence is assigned to the bin that contains all the valid codewords, the rest of the assignment can be arbitrary. Hence, there can be as many as $(2^{n-k} - 1)!$ valid syndrome formers. For each syndrome former, there can be up to 2^k matching inverse syndrome formers, each producing a different set of auxiliary sequences. We note that any valid pair of SF and ISF can be used in the source en-

coder and the source decoder that we present below, but the complexity for constructing different SF-ISF pairs may vary.

(i) Source encoder: as illustrated in Figure 4, the source encoder is simply a syndrome former that maps a source sequence X^n to a syndrome sequence S^{n-k}.

(ii) Source decoder: the source decoder in Figure 4 consists of a matching inverse syndrome former and the original channel decoder. The auxiliary sequence at the output of the ISF is first subtracted from the side information Y^n, whose result is then fed into the channel decoder to perform the conventional channel decoding. If the channel code is sufficiently powerful, then the output of the channel decoder, when added back to the auxiliary sequence, will almost surely recover the original source sequence X^n.

Proof of the validity. The validity of the above source encoder follows directly from the definition of the syndrome former. The validity of the above source decoder is warranted by the fact that the same distance property is preserved in all bins. Let X^n and Y^n denote two binary, memoryless sources with correlation $P(Y^n|X^n) = (P(Y|X))^n$. The virtual transmission channel as specified by $P(Y^n|X^n)$ can be viewed as a discrete memoryless channel: $Y = X \oplus Z$, where Z is the additive binary memoryless noise $P(Z) = P(Y|X)$.

Let $\mathbf{c}(\mathbf{s})$ denote a codeword $\mathbf{c}$ with syndrome sequence $\mathbf{s}$. Assume that $\mathbf{x} = \mathbf{c}_1(\mathbf{s}_1)$ is the source sequence to be compressed. The encoder will find $\mathbf{s}_1$ and sends it to the decoder.

The decoder has side information $\mathbf{y}$, where $\mathbf{y} = \mathbf{x} \oplus \mathbf{z}$. Upon receiving $\mathbf{s}_1$, the ISF will find an arbitrary sequence, say $\mathbf{c}_2(\mathbf{s}_1)$ from the coset of $\mathbf{s}_1$. Notice that the subtraction of the auxiliary sequence $\mathbf{c}_2$ from the side information $\mathbf{y}$, that is, $\mathbf{y} \oplus \mathbf{c}_2$, forms a noisy codeword (with respect to the virtual transmission channel), since

$$\mathbf{y} \oplus \mathbf{c}_2(\mathbf{s}_1) = \mathbf{x} \oplus \mathbf{z} \oplus \mathbf{c}_2(\mathbf{s}_1)$$
$$= \underbrace{\mathbf{c}_1(\mathbf{s}_1) \oplus \mathbf{c}_2(\mathbf{s}_1)}_{\text{some valid codeword } \mathbf{c}_3(\mathbf{0})} \oplus \mathbf{z}. \qquad (2)$$

Hence, if the channel code is sufficiently powerful, that is, capacity-approaching on the virtual channel, it can recover the valid codeword $\mathbf{c}_3(\mathbf{0})$ with a vanishing error probability ϵ. Since $\mathbf{c}_3(\mathbf{0}) = \mathbf{y} \oplus \mathbf{c}_2(\mathbf{s}_1) \oplus \mathbf{z} = \mathbf{x} + \mathbf{c}_2$, adding back the auxiliary sequence $\mathbf{c}_2$ yields the original sequence $\mathbf{x}$. Clearly, the probability that the data sequence $\mathbf{x}$ is not losslessly recovered is the probability that the channel decoder fails to correctly decode $\mathbf{c}_3(\mathbf{0})$, which equals $\epsilon \to 0$. It then follows that data sequences can be decoded with a vanishing distortion using the above source decoder (and source encoder). □

4. CONSTRUCTION OF THE SYNDROME FORMER AND THE INVERSE SYNDROME FORMER

With the above universal source encoder and source decoder, asymmetric DSC becomes a straightforward two-step
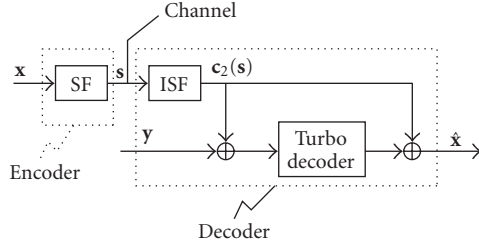
FIGURE 4: The structure of the universal source encoder and source decoder for asymmetric DSC.

process: (i) to choose a good channel code with the appropriate code rate and sufficient error correction capability for the virtual channel, and (ii) to construct a pair of valid SF and ISF for this code. The former could certainly make use of the rich results and findings developed in the channel coding research. Here, we focus on the latter issue.

For linear block codes where the code structure is well defined by the parity check matrices, SF-ISF construction is a straightforward task. For example, the parity check matrix and its left inverse can be used as a valid pair of syndrome former and inverse syndrome former. For convolutional codes, this is as convenient, although the process is less well known [26]. The real difficulty lies in the class of concatenated codes which are formed from component block/convolutional codes and random interleavers and which happen to include many powerful channel codes, such as convolutional turbo codes and block turbo codes. In theory, a concatenated code can still be treated, in a loose sense, as a linear block code and, hence, a closed-form parity check matrix still exists and can be used as a syndrome former. In practice, however, to derive such a parity check matrix is prohibitively complex, if not impossible.

In searching for practical SF-ISF solutions for concatenated codes, we have found a clever way to get around the random interleaver problem. The key idea is to adopt the same/similar parallel or serial structure as a concatenated code built from its component codes, and to construct the SF-ISF pair from the sub-SF-ISF pairs accordingly. In addition, we have found that by exploiting a specific type of sub-SF-ISF pair (with certain properties), the construction can be further simplified.

Below, we take (convolutional) turbo codes as an illustrating example and discuss in detail the proposed construction method. To start, we first introduce the SF-ISF construction for (component) convolutional codes, and then proceed to parallel turbo codes [20] and lastly serial turbo codes.

4.1. SF-ISF construction for convolutional codes

In his 1992 paper on trellis shaping [26], Forney described a simple way to construct syndrome formers and inverse syndrome formers for convolutional codes. For a rate k/n binary linear convolutional code with $k \times n$ generator matrix G, it is shown that the SF can be implemented using an $n/(n-k)$ linear sequential circuit specified by an $n \times (n-k)$ transfer

matrix H^T with rank $(n-k)$ such that

$$GH^T = \mathbf{0}_{k \times (n-k)}, \tag{3}$$

where $\mathbf{0}_{k \times (n-k)}$ is the $k \times (n-k)$ all-zero matrix. Clearly, the constraint in (3) makes sure that all valid codewords are associated with the all-zero syndrome 0^{n-k} and that length-n codewords/sequences have the same syndrome if and only if they belong to the same coset. (It should be noted that the generator matrix of a binary convolutional code considered here is formed of generator polynomials in the $\mathcal{D}$ domain and, hence, is different from the $\{0, 1\}$ generator matrix of a linear block code.)

Similar to the case of linear block codes, the inverse syndrome former, $(H^{-1})^T$, can be obtained by taking the left inverse of the syndrome former, that is,

$$(H^{-1})^T H^T = I_{n-k}, \tag{4}$$

where I_{n-k} is an identity matrix with rank $n - k$.

As mentioned before, the SF-ISF pair is not unique for a given code. In fact, any linear sequential circuit having the required number of inputs and outputs and meeting the constraints of (3) and (4) represents a valid construction for SF and ISF, but the complexity varies.

As an example, consider a rate 1/2 recursive systematic convolutional (RSC) code with generator matrix $G = [1, (1 + D^2)/(1 + D + D^2)]$. A simple SF-ISF construction can take the form of

$$\text{SF} : H^T = \begin{bmatrix} 1 + D^2 \\ 1 + D + D^2 \end{bmatrix}, \tag{5}$$

$$\text{ISF} : (H^{-1})^T = [1 + D, D].$$

Another equally simple construction is given by

$$\text{SF} : H^T = \begin{bmatrix} \dfrac{1 + D^2}{1 + D + D^2} \\ 1 \end{bmatrix}, \tag{6}$$

$$\text{ISF} : (H^{-1})^T = [0, 1]. \tag{7}$$

While there are many other valid constructions and while they all fulfill the roles of SF and ISF, we would like to bring special attention to the one given in (6) and (7). As illustrated in Figure 5, an interesting property about this specific ISF is that, for any given syndrome sequence, it always finds the codeword whose systematic bits are all zeros. For ease of proposition, we define this feature "zero forcing." We note that for any systematic linear channel code, there exists a zero-forcing ISF (and its matching SF). This is easily verifiable since linearity in the code space ensures that every coset/bin contains one (and only one) sequence with the all-zero systematic part. As we will show later, exploiting the zero-forcing feature can significantly simplify the SF-ISF construction for concatenated codes.
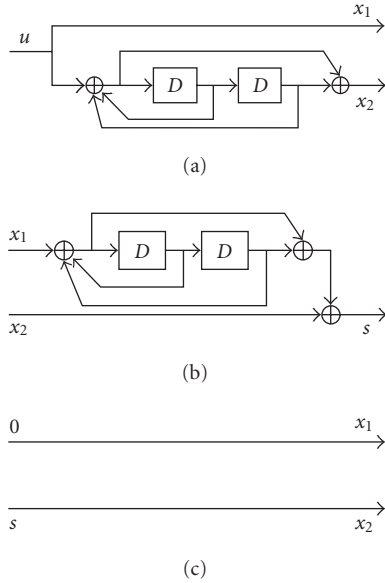
(a)

(b)

(c)

FIGURE 5: A rate 1/2 RSC code with generator matrix $G = [1, (1 + D^2)/(1 + D + D^2)]$ and its SF and ISF. (a) The encoder. (b) The linear sequential circuit implementation of a valid syndrome former $H^T = [(1 + D^2)/(1 + D + D^2), 1]^T$. (c) The matching inverse syndrome former $(H^{-1})^T = [0, 1]$.

4.2. SF-ISF construction for parallel turbo codes

Consider a typical parallel turbo code formed from two component RSC codes connected by a random interleaver. Let $R_1 = k/n_1$, $R_2 = k/n_2$, $G_1 = [I_k, P_1]$, and $G_2 = [I_k, P_2]$ denote the code rates and the generator matrices of the first and the second component RSC code, respectively, where I_k is the $k \times k$ identity matrix for generating k systematic bits, and P_1 and P_2 are $k \times (n_1 - k)$ and $k \times (n_2 - k)$ matrices for generating $(n_1 - k)$ and $(n_2 - k)$ parity check bits of the first branch and second branch. Since the systematic bits from the second branch are a scrambled version of those from the first branch, they are not transmitted. Hence, the overall code rate is given by $R = k/(n_1 + n_2 - k) = R_1 R_2/(R_1 + R_2 - R_1 R_2)$.

Let $\mathbf{x}$ denote a source sequence to be compressed. Since it is viewed as a virtual codeword of this parallel turbo code, it consists of three parts: the systematic bits from the first branch, $\mathbf{x}_s$, the parity bits from the first branch, $\mathbf{x}_1$, and the parity bits from the second branch, $\mathbf{x}_2$. Clearly, these three segments can form two virtual subcodewords, $[\mathbf{x}_s, \mathbf{x}_1]$ for the first component code and $[\pi(\mathbf{x}_s), \mathbf{x}_2]$ for the second component code, where $\pi(\cdot)$ denotes the interleaving operation. On the other hand, the length $(n_1 + n_2 - 2k)$ syndrome sequence of the turbo code can also be decomposed into two subsyndrome sequences: $\mathbf{s}_1$ (of length $(n_1 - k)$) for the first component code, and $\mathbf{s}_2$ (of length $(n_2 - k)$) for the second component code. This observation leads to the natural idea of constructing the SF and the ISF of the turbo code by concatenating those of the component codes.

Following the discussion in the previous subsection, it is easy to obtain a valid pair of SF and ISF for each of the component RSC codes. Specifically, we limit the choice to the
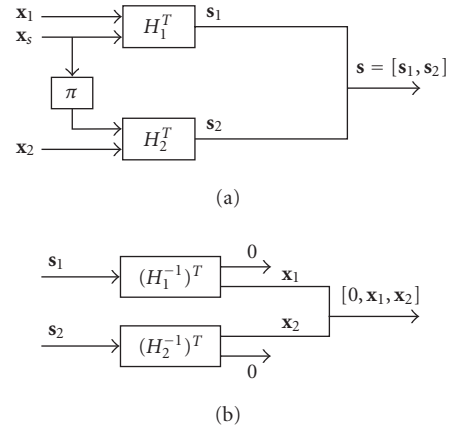

(a)

(b)

FIGURE 6: (a) The proposed SF for a general parallel turbo code. (b) The matching ISF. Note that both of the sub-ISFs, $(H_1^{-1})^T$ and $(H_2^{-1})^T$, need to be zero forcing, and the interleaver between the two sub-SFs is the same interleaver that is used in the turbo code.

zero-forcing SF-ISF pair for both component codes:

$$
\begin{aligned}
&\text{SF 1}: H_1^T = \begin{bmatrix} P_1 \\ I \end{bmatrix}_{n_1 \times (n_1 - k)}, \\
&\text{ISF 1}: (H_1^{-1})^T = [\mathbf{0}, I]_{(n_1 - k) \times n_1}, \\
&\text{SF 2}: H_2^T = \begin{bmatrix} P_2 \\ I \end{bmatrix}_{n_2 \times (n_2 - k)}, \\
&\text{ISF 2}: (H_2^{-1})^T = [\mathbf{0}, I]_{(n_2 - k) \times n_2}.
\end{aligned}
\tag{8}
$$

These sub-SFs and ISFs are then used to form the overall SF and ISF for the parallel turbo code, whose structures is shown in Figure 6.

It is easy to show that this construction is both valid and efficient. For the syndrome former, with every $(n_1 + n_2 - k)$ data bits (a virtual turbo codeword) at the input, H_1^T produces $(n_1 - k)$ subsyndrome bits and H_2^T produces $(n_2 - k)$ subsyndrome bits, which combined form a length $(n_1 + n_2 - 2k)$ syndrome sequence at the output. Further, codewords in the same coset are mapped to the same syndrome sequence and a valid turbo codeword is always mapped to the all-zero syndrome sequence. Hence, this represents a valid SF formulation which can be efficiently implemented using linear sequential circuits.

For the inverse syndrome former, we wish to emphasize that the simple formulation in Figure 6 is made possible by the zero-forcing sub-ISFs. Recall that the role of the (sub-)ISF is to find an arbitrary codeword associated to the given syndrome sequence. However, in order for the two sub-ISFs to jointly form an ISF for the turbo code, they need to *match* each other. By match, we mean that the systematic bits produced by the second sub-ISF need to be a scrambled version of those produced by the first sub-ISF. This seems to suggest the following two subtasks. First, one needs to have control over the *exact* codeword that each sub-ISF produces; in other words, an arbitrary mapping or an arbitrary ISF does not work. Second (and the more difficult one), since a matching

pair of sub-ISFs will be *interleaver dependent*, one needs to find a general rule to guide a possible "match." At first sight, these subtasks appear difficult to solve. However, a deeper investigation reveals that the zero-forcing sub-ISFs can fulfill both requirements simultaneously. Since the all-zero systematic bits are invariant regardless of what interleaver is used, zero-forcing sub-ISFs thus offer a simple solution to solve the potential "mismatching" problem for all interleavers!

4.3. SF-ISF construction for serial turbo codes

Serial turbo codes, as an extension to parallel turbo codes, have exhibited equally remarkable error correcting performance. Before proceeding to discuss their SF-ISF construction, we note that a serial turbo code, or more generally a serially concatenated code, needs to have a recursive inner code in order to achieve interleaving gain[1] [27]. Here we focus on serial turbo codes whose inner codes are both recursive and systematic. Again, the key is to exploit the sub-SF-ISF pairs of the component codes.

While the general idea is the same, the case of serial turbo codes is slightly more difficult, especially the construction of the inverse syndrome former. We consider a serial turbo code formed of an outer convolutional code (not necessarily recursive nor systematic) with rate $R_o = k/n_o$ and generator matrix G_o, a random interleaver (denoted as π), and an inner RSC code with rate $R_i = n_o/n$ and generator matrix $G_i = [I, P]$, where I is an identity matrix. The overall code rate is $R = k/n = R_o R_i$. For a block of k data bits, this serial turbo code produces a codeword of n bits. Hence, the corresponding syndrome sequence needs to contain $n - k = (n - n_o) + (n_o - k)$ bits. This suggests that a syndrome sequence $\mathbf{s}$ may be formed from two disjoint parts: a subsyndrome sequence of length $(n - n_o)$ from the inner code, denoted as $\mathbf{s}_i$, and a complementary part of length $(n_o - k)$ from the outer code, denoted as $\mathbf{s}_o$.

Consider a source sequence $\mathbf{x}$ of length n to be compressed to its syndrome sequence $\mathbf{s} = [\mathbf{s}_o, \mathbf{s}_i]$. For the (n, n_o) inner recursive systematic convolutional code, the entire sequence $\mathbf{x}$ can be viewed as a codeword that is formed from a length n_o "systematic" part, $\mathbf{x}_s$, and a length $(n - n_o)$ "parity" part, $\mathbf{x}_p$. According to what we have discussed about convolutional codes, the entire sequence $\mathbf{x}$ can thus be fed into the sub-SF of the inner code to generate $\mathbf{s}_i$. For the outer code, note that only the systematic part $\mathbf{x}_s$ is relevant, that is, $\mathbf{x}_s$ is the codeword of the outer code. Hence, $\mathbf{x}_s$, after deinterleaving, can be fed into the sub-SF of the outer code to generate $\mathbf{s}_o$. The combination of $\mathbf{s}_i$ and $\mathbf{s}_o$ thus completes the entire syndrome sequence. The overall structure of the SF for the serial turbo code is illustrated in Figure 7a.

The construction of a matching ISF is less obvious. We first present the structure before explaining why it works. As illustrated in Figure 7b, a valid ISF that matches to the above SF consists of four parts: the sub-ISFs of the outer and the inner component code, $(H_o^{-1})^T$ and $(H_i^{-1})^T$, the random interleaver, π, and the (sub-) encoder of the inner code, G_i. Similar to the case of parallel turbo codes, the interleaver is the same interleaver that is used in the serial turbo code, and the sub-ISF of the inner RSC code is a zero-forcing one: that is, $(H_i^{-1})^T = [\mathbf{0}, J]$, where J is a square matrix.

Below, we prove its validity by showing that the output of this ISF (i.e. the virtual codeword), when fed into the SF in Figure 7a, will yield the original syndrome sequence. Mathematically, this is to show that, for a given sequence $\mathbf{x}$ in the codeword space, where $\mathbf{x} = [\mathbf{x}_s, \mathbf{x}_p] = \mathit{ISF}([\mathbf{s}_o, \mathbf{s}_i])$, we have

$$\mathcal{SF}([\mathbf{x}_s, \mathbf{x}_p]) \implies [\mathbf{s}_o \mathbf{s}_i], \qquad (9)$$

where the notation $\mathcal{SF}(\mathbf{a}) \Rightarrow \mathbf{b}$ denotes that the SF will produce $\mathbf{b}$ at the output for $\mathbf{a}$ at the input. Similar notations will also be used for $\mathit{ISF}(\cdot)$, $\mathcal{H}_i^{-1}(\cdot)$ and the like.

Notice that $[\mathbf{x}_s, \mathbf{x}_p] = [\hat{\mathbf{x}}_s, \hat{\mathbf{x}}_p] \oplus [\bar{\mathbf{x}}_s, \bar{\mathbf{x}}_p]$ (see Figure 7b). By the linearity of the syndrome former, we have

$$\mathcal{SF}([\mathbf{x}_s, \mathbf{x}_p]) = \mathcal{SF}([\hat{\mathbf{x}}_s, \hat{\mathbf{x}}_p]) \oplus \mathcal{SF}([\bar{\mathbf{x}}_s, \bar{\mathbf{x}}_p]). \qquad (10)$$

Since $[\hat{\mathbf{x}}_s \hat{\mathbf{x}}_p]$ is a valid codeword of the inner code G_i, the sub-syndrome former $(H_i)^T$ will map it to the all-zero syndrome sequence, that is,

$$\mathcal{H}_i^T([\hat{\mathbf{x}}_s, \hat{\mathbf{x}}_p]) \implies \mathbf{0}. \qquad (11)$$

Since H_i^T and $(H_i^{-1})^T$ are a valid SF-ISF pair, we have

$$\mathcal{H}_i^T([\bar{\mathbf{x}}_s, \bar{\mathbf{x}}_p]) = \mathcal{H}_i^T\left((\mathcal{H}_i^{-1})^T(\mathbf{s}_i)\right) \implies \mathbf{s}_i. \qquad (12)$$

Gathering (10), (11) and (12), we have

$$\mathcal{H}_i^T([\mathbf{x}_s, \mathbf{x}_p]) \implies \mathbf{s}_i. \qquad (13)$$

On the other side, since $\bar{\mathbf{x}}_s$ is an all-zero vector, $\mathbf{x}_s$ is identical to $\hat{\mathbf{x}}_s$. Since G_i is a systematic encoder, we can see from Figure 7b that

$$\mathbf{x}_s = \hat{\mathbf{x}}_s = \tilde{\mathbf{w}} = \boldsymbol{\pi}(\mathbf{w}) = \boldsymbol{\pi}\left((\mathcal{H}_o^{-1})^T(\mathbf{s}_o)\right), \qquad (14)$$

that is, $\mathbf{x}_s$ is precisely the interleaved version of the output from the sub-ISF $(H_o^{-1})^T$ for which the input is $\mathbf{s}_o$. Hence, passing $\mathbf{x}_s$ into the deinterleaver and subsequently the sub-SF H_o^T will reproduce $\mathbf{s}_o$. This is exactly what the upper branch of the SF in Figure 7a performs:

$$\mathcal{H}_o^T(\boldsymbol{\pi}^{-1}(\mathbf{x}_s)) = \mathcal{H}_o^T\left(\boldsymbol{\pi}^{-1}\left(\boldsymbol{\pi}\left((\mathcal{H}_o^{-1})^T(\mathbf{s}_o)\right)\right)\right) \implies \mathbf{s}_o. \qquad (15)$$

Comparing (13) and (14) with the SF structure in Figure 7a, it becomes clear that (10) is satisfied. Hence, the proposed SF-ISF construction still holds for serial turbo codes.

---

[1]To be precise, a serially concatenated code needs to have an inner code which is recursive, an outer code (not necessarily recursive) which has a minimum distance of at least three, and a random interleaver between them in order to achieve interleaving gain on codeword error rate [27].
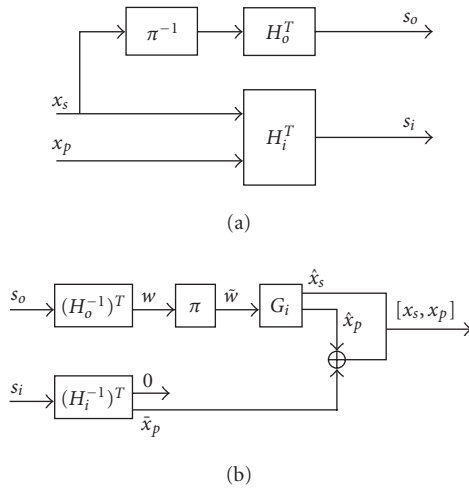
(a)

(b)

FIGURE 7: (a) The proposed SF for a general serial turbo code with an RSC inner code. (b) The matching ISF. Note that the inner sub-ISF, $(H_i^{-1})^T$, needs to be zero forcing.

5. COMMENTS ON THE PROPOSED SF-ISF APPROACH

The proposed SF-ISF approach provides a method for the direct exploitation of the binning idea discussed in Section 2. For memoryless binary symmetric sources, the approach is clearly optimal, as is guaranteed by the intrinsic optimality of the binning concept [2]. It is worth noting that this optimality holds for infinite block sizes as well as finite block sizes. (A constructive example demonstrating the optimality of the binning approach for finite block sizes can be found in [6].)

The construction of the syndrome former and the inverse syndrome former we demonstrated is simple and general. All operations involved are linear and reside in the binary domain, thus allowing cheap and efficient implementation using linear sequential circuits.

Besides simplicity and optimality, a particularly nice feature about the proposed SF-ISF scheme is its direct use of an existing (powerful) channel code. This allows the rich results available in the literature on channel codes to serve immediately and directly the DSC problem at hand. For example, a turbo code that is known to perform close to the capacity on BSC channels will also perform close to the theoretical limit for the DSC problem with binary "BSC-correlated" sources (i.e., $P(X \neq Y) = p$). Using a stronger component code (one that has a longer memory size and/or a better generator matrix) or simply increasing the codeword length (i.e., exploiting the interleaving gain of the turbo code) will achieve a better compression rate. In addition to conventional binary turbo codes, asymmetric turbo codes (which employ a different component code at each branch) (e.g., [23]) and nonbinary turbo codes, which are shown to yield better performances, can also be exploited for capacity-approaching DSC.

The last comment is on the generality of the proposed approach. Clearly, the proposed source encoder and source decoder are applicable to any binary linear channel code.

The proposed SF-ISF formulation has further paved the way for concatenated codes, breaking the tricky task of constructing the overall SF and ISF to a much simpler one of finding only the relevant sub-SFs and sub-ISFs of the component codes. This allows many powerful serially and parallelly concatenated codes to be readily exploited in DSC. In addition to the aforediscussed case of parallel and serial turbo codes, block turbo codes, also known as turbo product codes or, simply, product codes, are another good example. Product codes are formed of arrays of codewords from linear block codes (i.e. component codes) in a multidimensional fashion [21]. Depending on whether there are "parity-on-parity" bits, a 2-dimensional product code can be equivalently viewed as a serial (i.e., with "parity-on-parity") or a parallel (i.e., without "parity-on-parity") concatenation of the row code and the column code. Since the component codes of a product code are typically (simple) systematic linear block codes such as Reed-Solomon codes, BCH codes, Hamming codes, and single-parity check codes, sub-SFs and sub-ISFs are easy to construct. Further, since many product codes can be efficiently decoded on binary symmetric channels (BSC), for example, using the majority logic algorithm or the binary bit-flipping algorithm, they can potentially find great application in distributed compression where sources are binary and BSC correlated. To the best of the authors' knowledge, this is the only work thus far that has provided a DSC formulation for product codes.

6. SIMULATIONS

Despite the theoretical optimality of the proposed SF-ISF approach, computer simulations are needed to provide a true evaluation of its performance. In this section, we present the results of the proposed approach using rate-1/3 parallel turbo codes and rate-1/4 serial turbo codes. Appropriate clip-values are also used in the simulation to avoid numerical overflows and/or downflows in decoding.

The 8-state parallel turbo code considered has the same component codes as those in [15, 18]: $G_1 = G_2 = [1, (1+D+D^2+D^3)/(1+D^2+D^3)]$. A length 10^4 S-random interleaver with a spreading factor 17 and a length 10^3 S-random interleaver with a spreading factor 11 are used in the code, and 10 decoding iterations are performed before the turbo decoder outputs its estimates.

Table 1 lists the simulation results where n_π denotes the interleaver length. The interleaving gain can be easily seen from the table. If a normalized distortion of 10^{-6} is considered near-lossless, then this parallel turbo coding scheme with an interleaver length 10^4 can work for BSC-correlated sources with a correlation of $P(X \neq Y) = p = 0.145$. Since the compression rate is 2/3, there is a gap of only $2/3 - H(0.145) = 0.07$ bit/symbol from the theoretical limit. This gap is comparable to, in fact slightly better than, those reported in [15, 18], which are about 0.09 and 0.15 bit/symbol, respectively. It should be noted that in [15, 18], the same turbo code with the same interleaver size is used, but the code rate is different.

TABLE 1: Performance of the proposed SF-ISF scheme using parallel turbo codes.

Crossover prob.	Distortion	
p	$n_\pi = 10^3$	$n_\pi = 10^4$
0.10	0	0
0.11	1.5×10^{-6}	0
0.14	8.0×10^{-4}	0
0.145	4.0×10^{-3}	6.7×10^{-7}
0.155	3.5×10^{-2}	4.2×10^{-3}

TABLE 2: Performance of the proposed SF-ISF scheme using serial turbo codes.

	p	Distortion
	0.13	1.6×10^{-5}
$n_\pi = 2 \times 10^3$	0.15	3.3×10^{-5}
	0.16	9.0×10^{-5}
	0.165	5.0×10^{-4}
	p	Distortion
	0.17	7.6×10^{-7}
$n_\pi = 2 \times 10^4$	0.174	8.6×10^{-7}
	0.176	1.6×10^{-5}
	0.178	3.5×10^{-4}

In addition to conventional binary turbo codes, asymmetric turbo codes which employ a different component code at each branch are also tested for capacity-approaching DSC. Asymmetric turbo codes bear certain advantages in joint optimizing the performance at both the water-fall region and the error floor region [23]. We simulated the NP16-P16 (nonprimitive 16-state and primitive 16-state) turbo code in [23], where $G_1 = [1, (1 + D^4)/(1 + D + D^2 + D^3 + D^4)]$ and $G_2 = [1, (1 + D + D^2 + D^4)/(1 + D^3 + D^4)]$. A length 10^4 S-random interleaver with a spreading factor 17 is applied and 15 turbo decoding iterations are performed. Simulation results show that the proposed scheme provides a distortion of 3.4×10^{-7} when $p = 0.15$. This translates to a gap of only about 0.06 bit/symbol from the theoretical limit.

For the proposed SF-ISF scheme with serial turbo codes, we simulated a rate 1/4 serial turbo code whose outer code and inner code are given by generator matrices $G_o = [1, (1 + D + D^2 + D^3)/(1 + D^2 + D^3)]$ and $G_i = [1, 1/(1 + D)]$, respectively. A length 2×10^3 S-random interleaver with a spreading factor 15 and a length 2×10^4 S-random interleaver with a spreading factor 40 are used, and 10 decoding iterations are performed. The results are shown in Table 2. At a normalized distortion of 10^{-6}, we see that this serial turbo coding scheme with an interleaver size 2×10^4 can work for BSC-correlated sources of $p = 0.174$. The gap from the theoretical limit is only $1 - R - H(p) = 1 - 3/4 - H(0.174) = 0.08$ bit/symbol, which is again among the best results reported so far. For example, the DSC scheme using a rate 1/3 serial turbo code proposed in [19] has a gap of around 0.12 bit/symbol to the theoretical limit. The serial turbo code therein used specifically designed component codes, a length 10^5 S-random interleaver with a spreading factor of 35, and 20 decoding iterations [19].

7. CONCLUSION

This paper considers asymmetric compression for noiseless distributed source coding. An efficient SF-ISF approach is proposed to exploit the binning idea for linear channel codes in general and concatenated codes in particular. For binary symmetric sources, the proposed approach is shown to be simple and optimal. Simulation using serial and parallel turbo codes demonstrates compression rates that are very close to the theoretical limit. In light of the large amount of literature that exists on powerful linear channel codes and particularly capacity-approaching concatenated codes, the proposed approach has provided a useful and general framework that enables these channel codes to be optimally and efficiently exploited in distributed source coding.

While the discussion in the paper has demonstrated the efficiency of the proposed scheme, many interesting problems remain to be solved. For example, instead of revoking to time sharing, is there an optimal way to perform symmetric DSC to achieve a rate-versus-load balance? The works of [12, 13, 15] have certainly shed useful insight, but how about a general linear channel code? Notice that most of the works thus far have focused on uniform sources, but nonuniform sources are not uncommon in reality. For example, many binary images (e.g., facsimile images) may have a source distribution as biased as $p_0 = 0.96$ and $p_1 = 0.04$ [28]. For most communication and signal processing problems, nonuniform sources are not a concern since entropy compression can be performed to balance the source distribution prior to the intended task. For distributed source coding, however, such a preprocess will either ruin the intersource correlation or make the correlation analytically intractable and, hence, is not possible. It has been shown in [28] that for nonuniform sources, the conventional algebraic binning approach that uses the fixed-length syndrome sequences as the bin indexes is no longer optimal, and that a better approach should use variable-length bin indexes. Are there other and hopefully better approaches? Nonbinary sources are also interesting [29]. Will we employ nonbinary codes like turbo codes over $GF(q)$ or over rings, or are binary codes sufficient? How about adaptive DSC? Can we make use of punctured turbo codes and/or rate-compatible turbo codes with the proposed approach? How to construct IS-ISF pairs for punctured codes? These are only a few of the many interesting issues that need attention.

REFERENCES

[1] A. D. Wyner, "Recent results in the Shannon theory," *IEEE Trans. Inform. Theory*, vol. 20, no. 1, pp. 2–10, 1974.

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

[3] Y. Oohama and T. S. Han, "Universal coding for the Slepian-Wolf data compression system and the strong converse theorem," *IEEE Trans. Inform. Theory*, vol. 40, no. 6, pp. 1908–1919, 1994.

[4] A. Wyner, "On source coding with side information at the decoder," *IEEE Trans. Inform. Theory*, vol. 21, no. 3, pp. 294–300, 1975.

[5] S. Shamai and S. Verdú, "Capacity of channels with side information," *European Transactions on Telecommunications*, vol. 6, no. 5, pp. 587–600, 1995.

[6] S. S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): design and construction," *IEEE Tran. Inform. Theory*, vol. 49, no. 3, pp. 626–643, 2003.

[7] S. Servetto, "Quantization with side information: lattice codes, asymptotics, and applications in wireless networks," submitted to IEEE Trans. Inform. Theory, 2002.

[8] R. Zamir, S. Shamai, and U. Erez, "Nested linear/lattice codes for structured multiterminal binning," *IEEE Trans. Inform. Theory*, vol. 48, no. 6, pp. 1250–1276, 2002.

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

[10] G. Caire, S. Shamai, and S. Verdú, "A new data compression algorithm for sources with memory based on error correcting codes," in *Proc. IEEE Information Theory Workshop*, pp. 291–295, Paris, France, March 2003.

[11] J. Muramatsu, T. Uyematsu, and T. Wadayama, "Low density parity check matrices for coding of correlated sources," in *Proc. IEEE International Symposium on Information Theory (ISIT '03)*, pp. 173–176, Yokohama, Japan, June 2003.

[12] D. Schonberg, K. Ramchandran, and S. S. Pradhan, "Distributed code constructions for the entire Slepian-Wolf rate region for arbitrarily correlated sources," in *Proc. IEEE Data Compression Conference (DCC '04)*, pp. 292–301, Snowbird, Utah, USA, March 2004.

[13] V. Stankovic, A. D. Liveris, Z. Xiong, and C. N. Georghiades, "Design of Slepian-Wolf codes by channel code partitioning," in *Proc. IEEE Data Compression Conference (DCC '04)*, pp. 302–311, Snowbird, Utah, USA, March 2004.

[14] R. Hu, R. Viswanathan, and J. Li, "A new coding scheme for the noisy-channel Slepian-Wolf problem: Separate design and joint decoding," in *Proc. IEEE Global Telecommunications Conference (GLOBECOM '04)*, vol. 1, pp. 51–55, Dallas, Tex, USA, November 2004.

[15] J. Garcia-Frias and Y. Zhao, "Compression of correlated binary sources using turbo codes," *IEEE Commun. Lett.*, vol. 5, no. 10, pp. 417–419, 2001.

[16] A. Aaron and B. Girod, "Compression with side information using turbo codes," in *Proc. IEEE Data Compression Conference (DCC '02)*, pp. 252–261, Snowbird, Utah, USA, April 2002.

[17] J. Bajcsy and P. Mitran, "Coding for the Slepian-Wolf problem with turbo codes," in *Proc. IEEE Global Telecommunications Conference (GLOBECOM '01)*, vol. 2, pp. 1400–1404, San Antonio, Tex, USA, November 2001.

[18] A. D. Liveris, Z. Xiong, and C. N. Georghiades, "Distributed compression of binary sources using conventional parallel and serial concatenated convolutional codes," in *Proc. IEEE Data Compression Conference (DCC '03)*, pp. 193–202, Snowbird, Utah, USA, March 2003.

[19] I. Deslauriers and J. Bajcsy, "Serial turbo coding for data compression and the Slepian-Wolf problem," in *Proc. IEEE Information Theory Workshop*, pp. 296–299, Paris, France, March 2003.

[20] Z. Tu, J. Li, and R. Blum, "Compression of a binary source with side information using parallelly concatenated convolutional codes," in *Proc. IEEE Global Telecommunications Conference (GLOBECOM '04)*, vol. 1, pp. 46–50, Dallas, Tex, USA, November 2004.

[21] R. M. Pyndiah, "Near-optimum decoding of product codes: block turbo codes," *IEEE Trans. Commun.*, vol. 46, no. 8, pp. 1003–1010, 1998.

[22] A. Liveris, Z. Xiong, and C. N. Georghiades, "Compression of binary sources with side information using low-density parity-check codes," in *Proc. IEEE Global Telecommunications Conference (GLOBECOM '02)*, vol. 2, pp. 1300–1304, November 2002.

[23] O. Y. Takeshita, O. M. Collins, P. C. Massey, and D. J. Costello Jr., "A note on asymmetric turbo-codes," *IEEE Commun. Lett.*, vol. 3, no. 3, pp. 69–71, March 1999.

[24] I. Csiszar, "Linear codes for sources and source networks: error exponents, universal coding," *IEEE Trans. Inform. Theory*, vol. 28, no. 4, pp. 585–592, 1982.

[25] J. Li, Z. Tu, and R. S. Blum, "How optimal is algebraic binning approach: a case study of the turbo-binning scheme," in *Proc. 38th Annual Conference on Information Sciences and Systems (CISS '04)*, Princeton, NJ, March 2004.

[26] G. D. Forney Jr., "Trellis shaping," *IEEE Trans. Inform. Theory*, vol. 38, no. 2, pp. 281–300, 1992.

[27] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Serial concatenation of interleaved codes: performance analysis, design, and iterative decoding," *IEEE Trans. Inform. Theory*, vol. 44, no. 3, pp. 909–926, 1998.

[28] J. Li, Z. Tu, and R. Blum, "Slepian-Wolf coding for nonuniform sources using turbo codes," in *Proc. IEEE Data Compression Conference (DCC '04)*, pp. 312–321, Snowbird, Utah, USA, March 2004.

[29] Y. Zhao and J. Garcia-Frias, "Joint estimation and data compression of correlated non-binary sources using punctured turbo codes," in *Proc. 36th Annual Conference on Information Sciences and Systems (CISS '02)*, Princeton, NJ, USA, March 2002.

Zhenyu Tu received the B.S. degree in electrical engineering from Nanchang University, Jiangxi, China, in 1998, and the M.S. degree in circuits & systems from Beijing University of Posts & Telecommunications, Beijing, China, in 2001. He was with Huawei Technologies Co., Shenzhen, China in the summer of 2001. In the summer of 2004, he was a summer intern at the InterDigital Communication Corporation Incubation Center. He is a Graduate Research Assistant in the Signal Processing and Communication Research Lab, ECE Department at Lehigh University, Bethlehem, PA, and is currently working toward the Ph.D. degree in electrical engineering. His research interests include channel/source coding and information theory for multiuser communications.
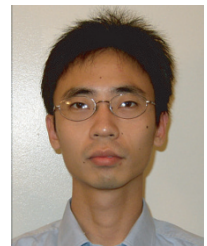
Jing Li (Tiffany) received the B.S. in computer science from Peking University, Beijing, China, in 1997 and the M.E. and the Ph.D. in electrical engineering from Texas A&M University, College Station, Texas, in 1999 and 2002, respectively. She worked with Seagate Research Laboratory and TycoTelecommunications Laboratory in the Summer of 2000 and 2001, respectively. Since January 2003, she has been a member of the faculty of the Electrical Engineering Department at Lehigh University, Bethlehem, Pennsylvania. Her research interests fall in the general area of communication and network systems, with emphasis on algorithm design and analysis, forward error correction coding, source-channel coding, cooperative networks, and signal processing for data storage and optical systems. She is the recipient of the Texas A&M University Ethel Ashworth-Tsutsui Memorial Award for Research (2001) and the JW Van Dyke Memorial Award for Academic Excellence (2001), the Peking University Outstanding Honors Science Student Award (1996), the Chinese Academy of Science Striving Scholarship (1994–1997), and the Zhejiang Province Zhang Pen-Xi Memorial Scholarship (1993). She was also the national finalist and the third-place prize winner in Chinese Mathematics Olympiad in 1993. She is currently an associate editor for IEEE Communications Letters.

Rick S. Blum received the B.S. in electrical engineering from the Pennsylvania State University in 1984 and the M.S. and Ph.D. in electrical engineering from the University of Pennsylvania in 1987 and 1991, respectively. From 1984 to 1991 he was a member of technical staff at General Electric Aerospace in Valley Forge, Pennsylvania and he graduated from GE's Advanced Course in Engineering. Since 1991, he has been with the Electrical and Computer Engineering Department at Lehigh University in Bethlehem, Pennsylvania, where he is currently a Professor and holds the Robert W. Wieseman Chair in electrical engineering. His research interests include signal detection and estimation and related topics in the areas of signal processing and communications. He is currently an associate editor for the IEEE Transactions on Signal Processing and for IEEE Communications Letters. He was a member of the Signal Processing for Communications Technical Committee of the IEEE Signal Processing Society. He is a member of Eta Kappa Nu and Sigma Xi, and holds a patent for a parallel signal and image processor architecture. He was awarded an ONR Young Investigator Award in 1997 and an NSF Research Initiation Award in 1992.