

Symmetric Distributed Source Coding Using LDPC Code

Baochun Bai*, Yang Yang†, Pierre Boulanger*, and Janelle Harms*

* Department of Computing Science, University of Alberta, Edmonton, Canada

Email: {baochun, pierreb, harms}@cs.ualberta.ca

† Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX, USA

Email: {yangyang}@ece.tamu.edu

Abstract—Distributed source coding is a promising enabling technology for sensor network applications. Symmetric Distributed source coding can achieve the entire Slepian-Wolf rate region and offer sensor network applications a wide range of options to compress and transmit data. In this paper, we design a new symmetric distributed coding scheme that realize the benefits of both simplified code construction and bit incorrespondence handling. In particular, we propose a general scheme to deal with the bit incorrespondence problem, which is not specifically addressed by previous approaches. With the ability to deal with the bit incorrespondence in the code structure, it is more likely for our approach to be used in real sensor network environments.

I. INTRODUCTION

Wireless sensor networks have attracted a great deal of research attention in recent years. In a typical sensor network, tens of thousands, even millions, of sensor nodes cooperate to collect data and fulfill a specific task. Different from many other wireless devices such as cell phones, PDAs, and laptops, which can have battery recharged relatively easily and have powerful computational capability, wireless sensor nodes are expected to operate under limited and unrenewable power supply and constrained computational capability. Therefore, one of the paramount goals of wireless sensor network research is to reduce the energy consumption and extend the system life time. Since data collected by different wireless sensor nodes are highly correlated, data compression is often used to reduce the amount of information transmitted and the transmission power. There are two general approaches to realize data compression: joint source coding and distributed source coding. Joint source coding needs sensors to communicate with each other and demands a computationally complicated encoder. Distributed source coding needs only a relatively cheap encoder and promises to achieve similar compression ratio while not requiring sensors to communicate with each other. Fig. 1 shows its architecture. Therefore, distributed source coding is a suitable data compression scheme for wireless sensor networks [1].

Distributed source coding is based on the theorem proved by Slepian and Wolf in 1970s [2]. The fascinating aspect of distributed source coding is that efficient compression of two or more sources can be achieved by separate encoding and joint decoding. The correlation between two sources can be modeled as a virtual channel as shown in Fig. 1.

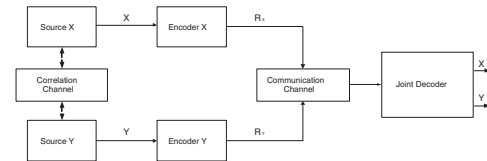


Fig. 1. Distributed Source Coding Architecture

Specifically, the Slepian-Wolf theorem gives the achievable rate region for distributed source coding. Namely, given two statistically correlated i.i.d finite-alphabet random sources X and Y . A conventional entropy encoder can achieve lossless compression with rate $R_X \geq H(X)$ and $R_Y \geq H(Y)$ when X and Y are separately encoded and decoded. When X and Y are jointly encoded and decoded using the joint source coding algorithm, lossless compression can be achieved with rate $R_{XY} \geq H(X, Y)$. Distributed source coding can achieve lossless compression with the following rate region (Fig. 2): $R_X \geq H(X|Y)$, $R_Y \geq H(Y|X)$, and $R_X + R_Y \geq H(X, Y)$.

A special case for distributed source coding problem is to compress one source X when the other source Y is available as side information at decoder. It is normally called asymmetric distributed source coding. Asymmetric lossless distributed source coding can be achieved with rate $R_X \geq H(X|Y)$ while the side information Y is losslessly compressed by using conventional entropy coding with rate $R_Y = H(Y)$ and available at the decoder. This case corresponds to the corner point in the rate region of the Slepian-Wolf theorem shown in Fig. 2. The distributed source coding scheme that can achieve an arbitrary point in the Slepian-Wolf rate region is normally referred as symmetric distributed source coding.

Slepian and Wolf's seminal work indeed has inspired other researchers to produce dozens of theoretical papers [3]. However, the practical distributed source coding algorithm was not developed until Pradhan and Ramchandran's work in 1999 [4]. Since distributed source coding is dual to channel coding [5], powerful channel codes such as lattice codes, convolutional codes, Turbo codes and low density parity check codes (LDPC), can all be used to realize distributed source coding. Algebraic binning is the basic idea to construct the practical distributed source codes, which is first illustrated by Wyner in the seventies [6]. The majority of early research

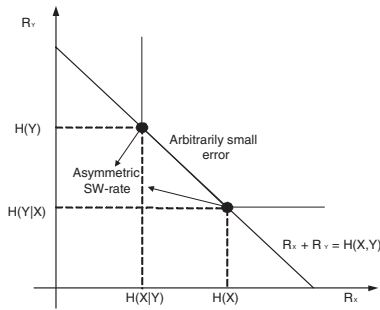


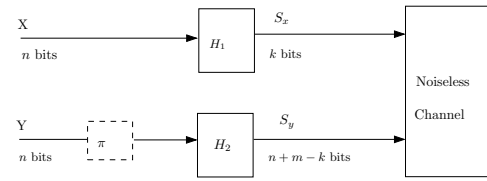
Fig. 2. Rate Region for Slepian-Wolf Coding (Two Sources)

effort is focused on asymmetric distributed source coding. A survey in 2004 [1] summarizes the early results.

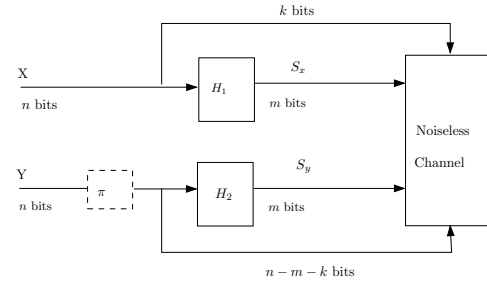
Though it appears difficult to design one channel code to achieve the entire rate region at first sight, several researchers have proposed excellent solutions to realize the symmetric distributed source coding [7]–[12]. These approaches can be classified into two classes: parity-based approaches and syndrome-based approaches. [10], [12] are parity-based approaches, where a decoder recovers the original source bits by processing the information bits and parity bits generated by an encoder. [7]–[9], [11] are syndrome-based approaches where a decoder deciphers the original source from part of source bits and syndrome bits generated by an encoder. [8], [9] need only one linear channel code to encode both sources. [7], [11], which both have roots in [13], construct independent subcodes from one main channel code and use them to realize the symmetric distributed source coding.

The aforementioned symmetric distributed source coding approaches are elegant and able to create capacity-approaching codes. However, it is still difficult to use them in the real world. All previous approaches implicitly assume that an encoder knows the exact bit correspondence between correlated sources, X and Y . However, this assumption is not true in most cases due to various reasons. For example, in a camera sensor network, pixel correspondence between two correlated images is not known at the encoder and can only be inferred at the decoder since there is no communication channel between two encoders. Without the knowledge of bit correspondence, [7], [8], [10] will fail to work since bit incorrespondence makes it impossible for the decoder to correctly recover the syndrome of difference pattern between X and Y , and thus it is unable to decipher the difference pattern, $X \oplus Y$, though they offer mathematically provable capacity-approaching codes. The decoder normally knows the bit correspondence. Though authors do not specifically address the bit incorrespondence problem, [11] can successfully decode the original mismatched sources because the process to recover the bit correspondence can be easily integrated into the decoding process as shown in Fig. 3(a) and 3(c), where bit incorrespondence between sources is modeled as a mapping, π , which is used to change the bit order of one source. However, the code partitioning technique makes it difficult to design capacity-approaching codes since it requires

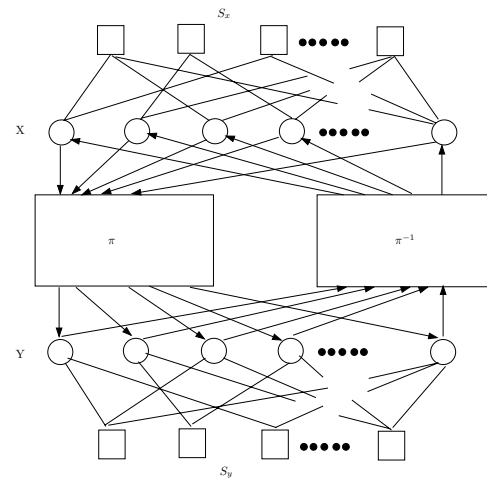
both the main code and two subcodes to be good codes [11].



(a) Encoder Using Two Subcodes. H_1 and H_2 are the parity check matrix of respective subcodes.



(b) Encoder Using One Channel Code. For EETG, H_1 and H_2 are the parity check matrix of one LDPC code and its permutation equivalent code. For SSIF, $H_1 = H_2$.



(c) Decoder Using Extended Tanner Graph

Fig. 3. Illustration of Encoder and Decoder Using Extended Tanner Graph. π is a mapping, which is used to change the bit order of a source.

In this paper, inspired by the work [8], [11], we propose an enhanced symmetric distributed source coding approach that combines both advantages of two previous approaches. Namely, at encoder we use one capacity-approaching code to encode the correlated sources, which help us circumvent the difficulty of code construction in [11]; at the decoder we use the message-passing algorithm on the extended Tanner graph, which makes it easy to handle the bit incorrespondence

problem. We termed our approach as Enhanced Extended Tanner Graph (EETG) method. In addition, we put forward a simple heuristic to construct the extended Tanner graph to achieve the better decoding performance. Furthermore, we introduce a general framework to handle the bit incorrespondence problem.

The remainder of the paper is organized as follows. In Section II, we review the two previous approaches [8], [11] in detail. In Section III, we elaborate on the proposed approach. Simulation results are presented in Section IV. Section V concludes the paper.

II. BACKGROUND

In this section, we first give some definition related with bit correspondence that will be used throughout the paper. Then we elaborate in detail on the symmetric distributed source coding schemes proposed in [11] and [8]. The approach in [11] is termed as “Two-machine Algorithm”, and the method in [8] is called “Symmetric SF-ISF (Syndrome Former - Inverse Syndrome Former) Framework (SSIF)”. Both approaches are general and can be used with any linear channel code. To facilitate easy exposition, we use LDPC codes as an example to explain the basic ideas. The virtual correlation channel is BSC (Binary Symmetric Channel).

A. Bit Correspondence

Given two n -bit sources, X and Y , bit correspondence is defined as a mapping between bit X_i and its correlated bit, Y_j . Namely, let π be the mapping, then $\pi(i) = j$. Given a bit location, i , which satisfies $1 \leq i \leq n$, and a mapping, π , let $\gamma = |i - \pi(i)|$. If $1 \leq \gamma \leq n$, the mapping π is called arbitrary mapping. If $1 \leq \gamma \leq t < n$, the mapping π is called bounded mapping. If an encoder knows the mapping as a priori knowledge, the encoder is said to be aware of bit correspondence between two sources. Otherwise, the encoder is oblivious to the bit correspondence. In the next two subsections, we assume that the mapping between two sources is an identity mapping and an encoder knows the mapping. Namely the mapping, π , shown in Fig. 3 has the function format $\pi(i) = i$.

B. Two-Machine Algorithm

The Two-machine algorithm realizes any point in the entire rate region by creating two subcodes from a main code. Both the main code and subcodes need to be capacity-approaching codes to avoid practical performance loss.

Fig. 3(a) illustrates its encoder architecture. Given the parity check matrix of two subcodes, encoding is realized by straightforward multiplication of the parity check matrix and the sources. Namely, $S_x = H_1 X$ and $S_y = H_2 Y$. The resulting syndromes, S_x and S_y , are transmitted to the decoder.

Fig. 3(c) shows the extended Tanner graph used by the message-passing decoding algorithm. The message-passing algorithm for each single Tanner graph is exactly the same as the typical message-passing algorithm used to decode LDPC codes. The only difference is that messages are also

passed between two Tanner graphs to exchange extrinsic information. The exact formulas to calculate the extrinsic information can be found in [11]. We omit it for brevity. A column dropping procedure is proposed to create parity check matrices of subcodes from the parity check matrix of the main code. The column-dropping procedure might not guarantee to generate capacity-approaching subcodes. Further research effort is needed to refine the column dropping procedure [11].

C. Symmetric SF-ISF Framework

SSIF uses one channel code to achieve an arbitrary rate pair. Its encoder structure is illustrated in Fig. 3(b). SSIF requires two encoders using the same parity check matrix, which is the syndrome former of a LDPC code. Let $H = H_1 = H_2$ and its size is $m \times n$. Then the sum rate of two sources is $m + n$. Each encoder transmits its syndrome, m bits, and complementary subset of first $n - m$ bits. Different rates between two sources are achieved by adjusting what subsets of source bits to transmit.

The decoder in SSIF proceeds in two steps. Given two binary sources, X and Y . Let Z be the difference patten of X and Y . Namely $Z = X \oplus Y$. In first step, the syndrome corresponding to Z can be obtained through $S_z = S_x \oplus S_y$. Then S_z is passed through an inverse syndrome former, which is H^{-1} for LDPC codes, to get the noise codeword of the difference pattern, Z . Z can be recovered after passing through the noise codeword into a channel decoder corresponding to H . In the second step, with the knowledge of the difference pattern, Z , X_{k+1}^{n-m} and Y_1^k can be recovered through the following equations.

$$\begin{aligned} X_{k+1}^{n-m} &= Y_{k+1}^{n-m} \oplus Z_{k+1}^{n-m} & (1) \\ Y_1^k &= X_1^k \oplus Z_1^k & (2) \end{aligned}$$

To decipher the rest of bits, SSIF partitions H into two sub-matrices:

$$H_{m \times n} = [A_{m \times (n-m)}, B_{m \times m}]$$

where B is square matrix and must have full rank. Since

$$S_x = HX = [A, B] \begin{bmatrix} X_1^{n-m} \\ X_{n-m+1}^n \end{bmatrix} = AX_1^{n-m} \oplus BX_{n-m+1}^n$$

we can obtain the remaining m source bits using

$$X_{n-m+1}^n = B^{-1}(S_x \oplus AX_1^{n-m})$$

After recovering all bits of X , the m remaining bits of Y can be recovered through $Y_{n-m+1}^n = X_{n-m+1}^n \oplus Z_{n-m+1}^n$.

From the decoding process of SSIF, it is obvious that the performance gap between SSIF and the theoretical limit solely depends on how well the channel code performs on the equivalent virtual correlation channel between two sources. In addition, SSIF imposes stringent requirements on bit correspondence at the encoder. Any bit incorrespondence will cause the decoder to fail.

III. THE PROPOSED APPROACH

In this section, we discuss the proposed symmetric distributed source coding approach (EETG) in detail. Our idea is simple and intuitive. EETG takes advantage of both benefits of the Two-machine algorithm and SSIF. It simplifies the code design and relaxes the bit correspondence requirement at an encoder. Though the idea can be used in all linear channel code, we focus our discussion in LDPC codes.

A. Encoder Design

As discussed in previous sessions, our goal is to find a symmetric distributed source coding approach that can easily construct a capacity-approaching code and can handle potential bit incoherence at encoder. It turns out that turbo-like iterative decoding is the only option since mapping and inverse mapping operation can be naturally integrated into such a decoder. Because the Two-machine algorithm [11] is essentially an iterative decoding algorithm, we decide to use the message-passing algorithm in an extended Tanner graph as the decoding algorithm. However, it is difficult to construct good subcodes if we follow the code partitioning philosophy [13]. Inspired by the observation that [7], [11] both root back to the code partitioning idea [13] and [7], [8] share the same ingredient to realize symmetric distributed source coding while [8] does not use code partitioning, we realize that one channel code without partitioning should be able to achieve similar performance when it is used in the iterative message-passing algorithm in an extended Tanner graph. Without much thought, it is evident that a channel code and its permutation equivalent code should be used. It is well known in algebraic coding theory that a channel code and its permutation equivalent code have the same weight distribution and thus the same error correction capability. Fig. 3(b) illustrates the encoder structure of the proposed approach. H_2 is formed by permutation of the columns of H_1 . It is almost the same as the encoder used in [8]. The only difference is that the two parity check matrices in EETG are permutation equivalent matrices while they are the same matrix in [8]. Compared with two same matrices, permutation equivalent parity check matrices make it easy to reduce the number of short cycles that go across two Tanner graphs and thus can improve the decoding performance. Like SSIF, EETG achieves the entire Slepian-Wolf region by adjusting which subsets of source bits are transmitted.

B. Decoder Design

Given a capacity-approaching channel code, the key to the decoder design is to construct a good extended Tanner graph to improve the performance of the message-passing algorithm. In the case of LDPC codes, given a LDPC code ensemble profile (λ, ρ) , the question is how we should choose a parity check matrix and its permutation equivalent matrix to construct a good extended Tanner graph. The naive approach would be to randomly choose a code from the code ensemble as the parity check matrix and get a permutation equivalent matrix by randomly permuting the columns of the known parity check

matrix and then construct the extended Tanner graph to decode the sources. Experiment results in Section IV show that this method performs poorly. We propose a simple heuristic to construct a parity check matrix and its permutation equivalent parity check matrix from a given channel code ensemble profile. Algorithm 1 gives the pseudo code to construct the parity check matrix and its permutation equivalent parity check matrix. Fig. 4 illustrates the structure of the constructed extended Tanner graph.

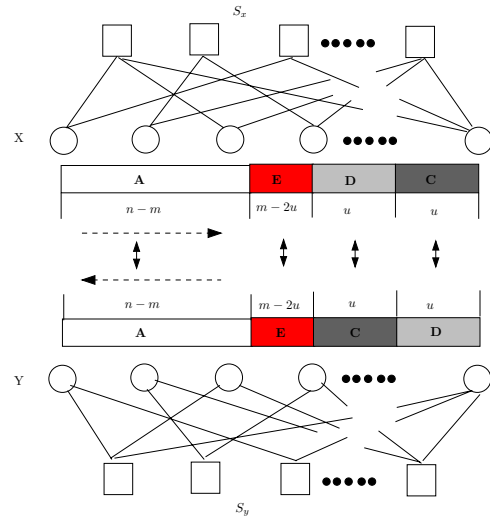


Fig. 4. Illustration of Permutation Equivalent Parity Check Matrix Construction. A represents the set of information variable nodes. C represents the set of internal reachable variable nodes. D represents the set of external reachable variable nodes. E represents isolated variable nodes. The dashed arrow indicates the degree decreasing direction of variable nodes in A for each matrix (Tanner graph).

We construct the extended Tanner graph by following the following basic guidelines: (1) reduce as many short cycles as possible; (2) let the transmitted source bits associate with variable nodes with large degree; (3) let the extrinsic information propagate into ambiguous nodes as soon as possible. The heuristic approach shown in Algorithm 1 is a specific realization of the above principles. Variable nodes in A consist of the set of information variable nodes since their initial log-likelihood ratio (LLR) is ∞ or $\log \frac{1-p}{p}$, where p is the crossover probability of the BSC virtual correlation channel, and includes most information about their original bits. Variable nodes in C consist of the set of internal reachable variable nodes since extrinsic information can be directly obtained from variable nodes in A from the same Tanner graph. Variable nodes in D consist of the set of external variable nodes since they can only obtain their extrinsic information initially from the other Tanner graph. The variable nodes in E consists of the set of isolated variable nodes since they only obtain their extrinsic information after all other nodes have their extrinsic information. The rationale to partition H_1 into A and B is inspired by the second decoding step of SSIF. We let the initially most ambiguous bits, whose initial LLR is 0, associate with B and hope that if all $n-m$ bits in A are known and the remaining m bits can be quickly decoded. In

Algorithm 1 Pseudo Code to Construct A Parity Check Matrix and Its Permutation Equivalent Parity Check Matrix

- 1: λ : variable node degree distribution
- 2: ρ : check node degree distribution
- 3: H_1 : parity check matrix
- 4: H_2 : permutation equivalent parity check matrix
- 5: n : the code length
- 6: m : the syndrome length

- 7: Randomly construct a Tanner graph as H_1 based on the channel code profile (λ, ρ) ;
- 8: Partition H_1 into two matrices $A_{m \times (n-m)}$ and $B_{m \times m}$. Make sure that B is full rank and the degree of variable nodes in B is as small as possible;
- 9: Partition B into three matrices $C_{m \times u}$, $D_{m \times u}$, and $E_{m \times (m-2u)}$. Suppose that all $n - m$ variable nodes in A are known, make sure that all u variable nodes in C are successfully decoded under BEC assumption and no variable nodes in D and E can be decoded using message-passing algorithm in the Tanner graph of H_1 . Suppose that all $n - m$ variable nodes in A and $2u$ nodes in C and D are known, make sure that all $m - 2u$ variable nodes in E can be successfully decoded under BEC assumption using message-passing algorithm in the Tanner graph of H_1 ;
- 10: Sort the columns in A based on variable node degree so that the degree of variable node associated with each column is in non-increasing order from left to right;
- 11: $H_1 = [AEDC]$;
- 12: Sort the columns in A based on variable node degree so that the degree of variable node associated with each column is in non-decreasing order from left to right;
- 13: Randomly permute columns in $C D E$;
- 14: $H_2 = [AEC D]$;
- 15: Construct the extended Tanner graph based on H_1 and H_2 ;

addition, the partition makes the asymmetric case naturally fit into the scheme. In the asymmetric case, one node transmits $n - m$ source bits and m syndrome bits in our scheme; the other node transmits m syndrome bits. Since our construction guarantees that $n - m$ bits correspond to A and B is full rank, the n original source bits can be easily decoded by solving m linear equations. The other source can then be decoded using the sum-product algorithm in a Tanner graph. The reason that we let the variable nodes in B have low degree is to mitigate the negative effect of the most ambiguous bits on the decoding of other bits. According to *tanh* rule of the sum-product algorithm in check nodes, if the message in one of the neighbor nodes is zero, output message of the check node is zero. Therefore, if the degree of variable nodes in B is large, the extrinsic information propagation will be slow and some bits will be hard to decode. Similarly, our intention to sort the variable nodes in A based on their degree is to let the transmitted source bits, whose LLR is ∞ or $-\infty$, to associate

with large degree variable nodes and thus maximize their positive impact to decode other source bits. The exchange of C and D in the parity check matrix and its permutation matrix aims at accelerating the propagation of extrinsic information since variable nodes in C can get extrinsic information directly from internal iteration and variable nodes in D initially have no way to get extrinsic information from internal iterations. BEC assumption is used in the pseudo code because the extrinsic information propagation from known variable nodes to the ambiguous variable nodes is much like the decoding of erasure bits.

C. Handling The Bit Incorrespondence

Though our approach relaxes the stringent requirement on bit correspondence at the encoder, it still cannot totally eliminate all the problems created by the oblivion of bit incorrespondence at an encoder. The major problem created by the bit incorrespondence is that there is no way for two encoders to send non-overlapped source bits if the mapping between two sources is arbitrary and unknown at the encoder. Overlapped source bits will induce performance loss at the decoder since less information is available for the decoder to decipher the original source bits.

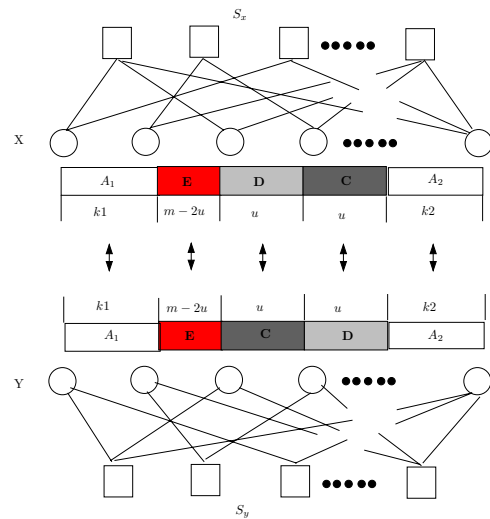


Fig. 5. Handle The Bit Incorrespondence

However, if the mapping is bounded and the bound, t , is less than $\frac{m}{2}$, we can modify the parity check matrices and the extended Tanner graph to avoid sending overlapping bits by encoders. For example, as shown in Fig. 5, we can move the matrix B to the middle portion of the parity check matrix H and let one encoder transmit the first $k1$ source bits and let the other decoder transmit the last $k2 = n - m - k1$ bits. Then no bit overlapping will occur. Thus the proposed approach can tolerate some degree of bit incorrespondence and is more suitable to be used in the real world.

IV. SIMULATION RESULTS

In this section, we present some simulation results. We demonstrate the feasibility and efficiency of the proposed

symmetric distributed source coding approach using a $1/2$ irregular (n, k) LDPC code. The degree distribution pair in example 2 of [14] is used to generate LDPC codes. In our simulation, two independent and identically distributed (i.i.d) binary sources, X and Y , are generated with a BSC(p) correlation.

We first show the effectiveness of the heuristic structural code construction approach. Fig. 6 compares the performance of the code generated by the random construction and code generated by structural construction. Results indicate that the code generated by the random construction has a consistent error rates regardless of the joint entropy rate. The code created by structural construction is significantly better than the code generated by random construction. For each data point, one million bits are simulated and the results are averaged. Experiments on other code length have similar results.

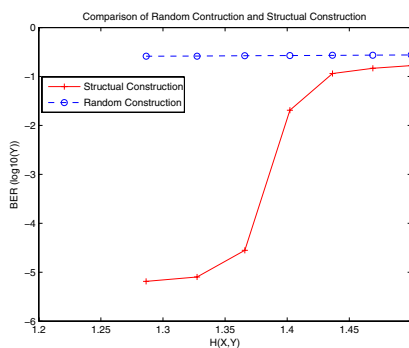


Fig. 6. Random Code Construction vs. Structural Code Construction. The code length is 10000. The rate pair is (0.5,0.5).

We next study the performance of compression of both sources at different rate pairs. Fig. 7 shows the log-scale bit error rate (BER). Results shows that asymmetric rate pair (1,0.5) has better performance than other symmetric rate pairs. The reason might be that the iterative message-passing algorithm is sub-optimal in the extended Tanner graph. The results of symmetric rate pairs are comparable to other symmetric distributed coding approaches using iterative decoding procedure such as [11], [12].

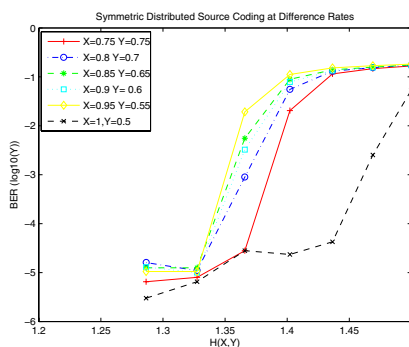


Fig. 7. Simulation Results under Various Rate Pairs. The code length is 10000.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we propose an enhanced symmetric distributed source coding approach that realizes the benefits of previous approaches. The idea is simple and effective. The approach simplifies the code construction procedure and relaxes the stringent requirement of bit correspondence at an encoder. A simple code construction heuristic is put forward to construct good extended Tanner graphs from a LDPC code profile. We also propose a general scheme to circumvent bit incorrespondence problem that frequently occurs in real sensor network environments. In the future, we would like to investigate using mathematical tools to analyze the code performance in the extended Tanner graph.

VI. ACKNOWLEDGMENT

The first author would like to thank Killam Trusts to make this research possible. We are also grateful to the grant support from NSERC (Natural Science and Engineering Research Council of Canada).

REFERENCES

- [1] Z. Xiong, A. Liveris, and S. Cheng, "Distributed source coding for sensor networks," *IEEE Signal Processing Magazine*, vol. 21, pp. 80–94, 2004.
- [2] J. D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Transaction on Information Theory*, pp. 471–480, 1973.
- [3] R. Zamir, S. Shamai, and U. Erez, "Nested linear/lattice codes for structured multiterminal binning," *IEEE Transactions on Information Theory*, vol. 48, no. 6, pp. 1250–1276, 2002.
- [4] S. S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (discus): Design and construction," in *Proc. of IEEE Data Compression Conference*, Snowbird, UT, USA, Mar. 1999, pp. 158–167.
- [5] S. S. Pradhan, J. Chou, and K. Ramchandran, "Duality between source coding and channel coding and its extension to the side information case," *IEEE Transactions on Information Theory*, vol. 49, no. 5, pp. 1181–1203, May 2003.
- [6] A. D. Wyner, "On source coding with side information at the decoder," *IEEE Transaction on Information Theory*, no. 5, pp. 294–300, May 1975.
- [7] V. Stankovic, A. D. Liveris, Z. Xiong, and C. N. Georghiades, "On code design for the slepian-wolf problem and lossless multiterminal networks," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1495–1507, 2006.
- [8] P. Tan and J. Li, "A general and optimal framework to achieve the entire rate region for slepian-wolf coding," *EURASIP Signal Processing Journal, Special Issue on Distributed Source Coding*, vol. 86, no. 11, 2006.
- [9] N. Gehrig and P. L. Dragotti, "Symmetric and asymmetric slepian-wolf codes with systematic and non-systematic linear codes," *IEEE Communication Letters*, vol. 9, no. 1, 2005.
- [10] M. Sartipi and F. Fekri, "Distributed source coding in wireless sensor networks using ldpc coding: The entire slepian-wolf rate region," in *Proc. of the Wireless Communication Networking Conference (WCNC 2005)*, New Orleans, USA, 2005.
- [11] D. Schonberg, "Practical distributed source coding and its application to the compression of encrypted data," Ph.D. dissertation, University of California, Berkeley, 2007.
- [12] J. Garcia-Frias and Y. Zhao, "Compression of correlated binary sources using turbo codes," *IEEE Communication Letters*, vol. 5, no. 10, pp. 417–419, 2001.
- [13] S. S. Pradhan and K. Ramchandran, "Generalized coset codes for distributed binning," *IEEE Transactions on Information Theory*, vol. 51, no. 10, pp. 3457–3474, 2005.
- [14] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low density parity check codes," *IEEE Transaction on Information Theory*, no. 2, pp. 619–637, Feb. 2001.