

# DISTRIBUTED COMPRESSION OF ZEROTREES OF WAVELET COEFFICIENTS

*Yige Wang, Shantanu Rane, Petros Boufounos and Anthony Vetro*

Mitsubishi Electric Research Laboratories, Cambridge, Massachusetts, U.S.A.

## ABSTRACT

A distributed coding algorithm is presented for compression of wavelet-transformed data. Data structures based on zerotrees are exploited for efficient compression of the significance map of wavelet coefficients. The coefficients are scanned in two stages, with a significance pass and refinement pass, similar to the SPIHT algorithm. The bits resulting from these passes are Slepian-Wolf coded using an LDPC syndrome code selected from a bank of available codes. A key realization is that, for each bitplane of the wavelet coefficients, the significance pass of the source data can be synchronized with that of the side information. This allows distributed compression of the significance pass. This is substantially different from previous mixed approaches in which the refinement pass was Slepian-Wolf coded, but the significance pass was coded independently. Rate-distortion results are presented for images from the ALOS AVNIR-2 multispectral dataset and compared against those obtained with SPIHT and JPEG2000.

**Index Terms**— Distributed Source Coding, Wavelets, LDPC codes, Zerotrees

## 1. INTRODUCTION

Distributed source coding is a compression framework in which correlated sources are encoded separately but decoded jointly. Unlike conventional compression algorithms, the correlation among the sources is exploited at the decoder only. The advantage is that separate encoding of the sources can be performed with low computational overhead and simpler circuitry. The framework rests on a seminal result by Slepian and Wolf, who proved that the compression performance of such a system is asymptotically identical to that of a conventional system in which the encoder exploits correlation among the sources [1]. Later, Wyner and Ziv derived bounds on the rate-distortion performance of a distributed codec, proving no loss with respect to conventional compression, for the special case of jointly Gaussian sources [2]. In the last five years, there has been a surge of interest in low-complexity video compression based on the distributed coding paradigm [3]. It has been shown that Wyner-Ziv coding has a significant compression advantage compared with schemes of comparable complexity, such as simple DCT-based intra coding or JPEG-like schemes. However wavelet-domain distributed coding schemes have shown rather modest compression gains over their non-distributed counterparts.

In this work, we develop an efficient distributed source coding algorithm for wavelet-transformed data. Our approach exploits correlations not only between the wavelet coefficients of correlated sources but also between their sparsity structure. Most wavelet-based distributed coding schemes today exploit correlation among the wavelet coefficients in much the same way as if they were DCT coefficients, i.e., they perform syndrome coding of the bitplanes of the source coefficients, and decode them using the bitplanes of

the side information coefficients [4, 5]. This does not fully exploit the sparsity structure of a wavelet decomposition, and places distributed source coding at a disadvantage with respect to advanced wavelet-based compression algorithms such as Embedded Zerotree Wavelet Coding (EZW) [6], Set Partitioning in Hierarchical Trees (SPIHT) [7] or JPEG2000 [8]. These algorithms exploit zerotrees of wavelet coefficients—data structures that can represent insignificant wavelet coefficients, or equivalently the significance map of wavelet coefficients, using very few bits. The present paper explores distributed source coding of these data structures, with the aim of outperforming wavelet-based image compression schemes.

Our aim is to develop an efficient low complexity algorithm for compression of multispectral images. Our work most closely resembles that of Cheung and Ortega on distributed coding of hyperspectral imagery [9]. In their work, distributed source coding is applied either directly to the wavelet bitplanes or to the refinement bits and sign bits in a wavelet decomposition. Compression of the significance map is accomplished using a non-distributed approach such as SPIHT. Our principal contribution is a distributed algorithm to compress the significance map. Encoding the significance map usually accounts for more than 50 % of the total bit budget in SPIHT or JPEG2000. By exploiting correlations among the significance maps of the source and the side information images, our approach reduces the overhead of the significance map, and increases compression efficiency.

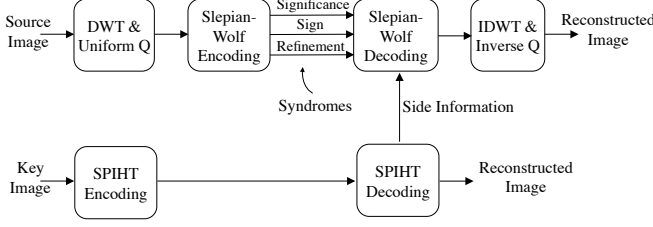
The remainder of this paper is organized as follows: Section 2 describes the proposed framework for distributed compression of the significance map, refinement bits and sign bits in a spatial orientation tree. In Section 3, we evaluate the algorithm on multispectral images, comparing its performance with a scheme in which each spectral band is independently compressed using SPIHT and JPEG2000.

## 2. DISTRIBUTED CODING OF WAVELET ZEROTREES

This section describes a wavelet-domain distributed source coding scheme, shown in the Fig. 1. We adopt an asymmetric distributed coding framework. The source and side information are 2-D images of equal size and that the same number of levels of wavelet decomposition are used for the source and side information images.

### 2.1. Side Information

The side information—which can be a previous video frame, or one of the image channels in a multispectral image—is independently compressed and sent to the decoder to aid in side information decoding. To compress the side information image, we adopt the well-known SPIHT algorithm. A desired number of 2-D wavelet decompositions are applied to the image, followed by uniform quantization. Proceeding from the most significant to the least significant bitplane, SPIHT efficiently communicates the significance map of the wavelet coefficients, along with their signs. The coefficient magnitudes are encoded in the significance map and in a separate refinement pass.



**Fig. 1.** Distributed coding scheme exploiting spatial orientation trees of wavelet coefficients. The significance map syndromes are decoded first followed by the sign syndromes and finally the refinement syndromes.

SPIHT was chosen to code the side information image, known as the key image in Wyner-Ziv coding, for two reasons: Firstly, SPIHT represents the significance map, refinement bitplanes and sign bits in a bit stream that is nearly incompressible, i.e., an entropy coder can only improve the compression ratio by a small amount. For low complexity encoding, the entropy coder can therefore be omitted. Secondly, to perform distributed coding, a SPIHT bit stream can be extracted from the source image, and Slepian-Wolf coding can be performed using an LDPC syndrome code. Owing to synchronization requirements at the decoder, the distributed version of SPIHT differs from the original SPIHT, as described below.

## 2.2. Distributed Encoding based on SPIHT

After the 2-D wavelet transform and the uniform quantization is applied to the source image, we construct a spatial orientation tree from a wavelet decomposition. Each node of this tree corresponds to a wavelet coefficient identified by its spatial coordinate, as  $c_{i,j}$ . Each node has either zero direct descendants in the next finer wavelet decomposition level or four direct descendants arranged in a  $2 \times 2$  array of adjacent wavelet coefficients. Descendants of a node which are not direct descendants are henceforth referred to as indirect descendants. Now, just like in SPIHT, define the following sets:

- $\mathcal{O}(i, j)$  is the set of coordinates of direct descendants of node  $(i, j)$ . This set is either empty or has 4 elements.
- $\mathcal{D}(i, j)$  is the set of all descendants of node  $(i, j)$
- $\mathcal{L}(i, j) = \mathcal{D}(i, j) - \mathcal{O}(i, j)$

As in SPIHT, declare three sets LIP (List of Insignificant Pixels), LSP (List of Significant Pixels), and LIS (List of Insignificant Sets). Different from SPIHT, declare a fourth set called TLIS (Temporary List of Insignificant Sets) and a fifth set called ALIS (Auxiliary List of Insignificant Sets). All five sets are initialized and used in Algorithm 1. Lastly define a significance function of a coordinate set  $\mathcal{A}$  for bitplane  $n$  as

$$S_n(\mathcal{A}) = \begin{cases} 1 & \text{if } \max_{(i,j) \in \mathcal{A}} |c_{i,j}| \geq 2^n \\ 0 & \text{otherwise} \end{cases}$$

Note that, each iteration of Algorithm 1 results in a different number of bits signalling the significance map. Further, the number of sign bits and refinement bits depend on the significance map. Thus, in each iteration, a Slepian-Wolf code with a different rate is necessary. In practice, syndromes derived from LDPC codes [10] of different rates are used as Slepian-Wolf codes.

### Algorithm 1 SW-ENCODE

- 1: **Initialization:**  $n = \lfloor \log_2(\max_{(i,j)} |c_{i,j}|) \rfloor$ . LSP =  $\emptyset$ . LIP contains coordinates  $(i, j)$  of all coefficients in the lowest frequency subband, i.e., roots of the zero tree. LIS contains the coordinates in LIP which have descendants in the higher frequency subbands. Initialize TLIS = LIS and ALIS =  $\emptyset$ .
- 2: **Sorting Pass:** (Significance Map and Sign Bits)
- 3: **for all**  $(i, j) \in \text{LIP}$  **do**
- 4:     Copy  $S_n(i, j)$  to a buffer  $U$ .
- 5:     Transmit the syndrome of  $U$  and clear  $U$ .
- 6: **for all**  $(i, j) \in \text{LIP}$  **do**
- 7:     **if**  $S_n(i, j) = 1$  **then**
- 8:         Move  $(i, j)$  from LIP to LSP.
- 9:         Copy the sign of  $c_{i,j}$  to a buffer  $V$ .
- 10: **for each**  $(i, j) \in \text{TLIS}$  **do**
- 11:     **if**  $(i, j)$  has direct descendant(s) **then**
- 12:         Copy  $S_n(\mathcal{D}(i, j))$  to the buffer  $U$
- 13:     **else if**  $(i, j)$  has indirect descendant(s) **then**
- 14:         Copy  $S_n(\mathcal{L}(i, j))$  to the buffer  $U$
- 15:     Transmit the syndrome of  $U$  and clear  $U$
- 16: **for each**  $(i, j) \in \text{TLIS}$  **do**
- 17:     **if**  $(i, j)$  has direct descendant(s) **and**  $S_n(\mathcal{D}(i, j)) = 1$  **then**
- 18:         **for each**  $(k, l) \in \mathcal{O}(i, j)$  **do**
- 19:             Copy  $S_n(k, l)$  to a buffer  $T$
- 20:         Transmit the syndrome of  $T$  and clear  $T$
- 21:     **for each**  $(i, j) \in \text{TLIS}$  **do**
- 22:         **if**  $(i, j)$  has direct descendant(s) **and**  $S_n(\mathcal{D}(i, j)) = 1$  **then**
- 23:             **for each**  $(k, l) \in \mathcal{O}(i, j)$  **do**
- 24:                 **if**  $S_n(k, l) = 1$  **then**
- 25:                     Add  $(k, l)$  to LSP
- 26:                     Append the sign of  $c_{k,l}$  to  $V$
- 27:                 **else if**  $S_n(k, l) = 0$  **then**
- 28:                     Append  $(k, l)$  to LIP
- 29:             **if**  $\mathcal{L}(i, j) \neq \emptyset$  **then**
- 30:                 Append  $(i, j)$  to ALIS as a coefficient with no direct descendants
- 31:             **else**
- 32:                 Remove  $(i, j)$  from LIS and TLIS
- 33:         **if**  $(i, j)$  has indirect descendant(s) **and**  $S_n(\mathcal{L}(i, j)) = 1$  **then**
- 34:             Add each  $(k, l) \in \mathcal{O}(i, j)$  to ALIS as a coefficient with direct descendants
- 35:             Remove  $(i, j)$  from LIS and TLIS.
- 36: **if** ALIS  $\neq \emptyset$  **then**
- 37:     Append ALIS to LIS.
- 38:     Set TLIS = ALIS, ALIS =  $\emptyset$ , and go to Step 10
- 39:     Transmit the syndrome of  $V$  and clear  $V$
- 40: **Refinement Pass:**
- 41:     For each  $(i, j) \in \text{LSP}$  except those coordinates that entered the LSP in the current sorting pass, copy the  $n^{\text{th}}$  MSB of  $|c_{i,j}|$  to a buffer  $Y$ .
- 42:     Transmit the syndrome of  $Y$  and clear  $Y$
- 43:      $n \leftarrow n - 1$ , and go to Step 2

The sequence of operations used to populate the LIS, LIP and LSP and extract bits from these sets in the Slepian Wolf encoding algorithm is different from that used in the original SPIHT algorithm [7]. This is necessary in order to enable side information de-

coding: Unlike the SPIHT algorithm, in which the bit stream is generated at the encoder bit by bit, and parsed at the decoder in the same sequence, our algorithm needs to maintain causality and synchronization for *blocks* of syndromes. Therefore, for every bit-plane, bits derived from the LIS are compressed first, followed by bits derived from the LIS, followed by bits derived from the LSP. Specifically, for bits derived from the LIS, coefficients with direct descendants are encoded separately from those with indirect descendants.

### 2.3. Side Information Decoding based on SPIHT

The decoding algorithm uses the same sequence of steps as Algorithm 1, with the following 2 caveats: Firstly, the operations are being performed on the side information image. Secondly, the steps which say “Transmit the syndrome of  $U$  and clear  $U$ ” are replaced by “Using the received syndromes, correct the errors in  $U$ , replace the corresponding side information bits with the corrected bits and clear  $U$ ”. It is in this error correction step that the correlation between the source and the side information is exploited. When the correlation is high, the number of syndromes required for error correction is significantly smaller than the size of  $U$  (or  $V$ ,  $Y$  and  $T$  as the case may be). After all steps are carried out with the above modifications to Algorithm 1, the side information bit stream is transformed into a reconstructed error-free SPIHT bit stream for the source image, which can then be decoded in the usual way. Note that, since the above algorithm is based on SPIHT, it is possible to decode bitplane-by-bitplane and stop whenever the desired image quality is obtained; extra syndromes may be neglected.

### 2.4. Predictors for Significance, Sign and Refinement Bits

The pseudocode of Algorithm 1 leaves out one detail, which is the distribution of the source given the side information. This distribution determines the initial log-likelihood ratios used by the LDPC belief propagation decoder. While, decoding the significance map and the sign bits, it is assumed that the bits are i.i.d Bernoulli- $p$ , where  $p$  is determined experimentally, by measurements on similar data sets. A different model might achieve better Slepian-Wolf coding performance, and this investigation is left for future work.

The refinement pass bits are decoded after the significance map and sign bits for a given bitplane have already been decoded. Thus, side information decoding of the refinement bits is exactly like side information decoding used in many distributed video coding schemes, i.e., conditioned on the value of the wavelet coefficients of the side information, the corresponding values of the wavelet coefficients of the source are distributed according to a Laplace distribution. For 2D images with  $d$  levels of decomposition, there are  $3d+1$  subbands, and a different Laplacian parameter is used for each subband. The values of the Laplacian parameters are determined by training over a large number of similar datasets.

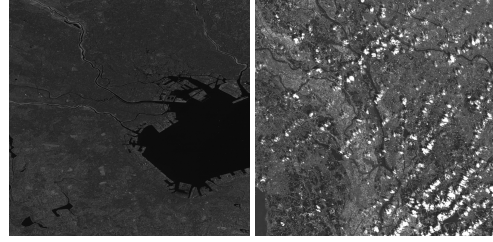
## 3. EXPERIMENTAL EVALUATION

The algorithm described above is valid for a variety of data sources. For concreteness, our experiments are carried out on real raw multispectral data acquired by the Advanced Land Observation Satellite (ALOS) using the Advanced Visible and Near Infra-red Radiometer (AVNIR-2). More details and directions to acquire these datasets are available online<sup>1</sup>. Specifications of the images used in this work are given in Table 1 and two of the images are shown in Fig. 2. The

<sup>1</sup>[http://www.alos-restec.jp/index\\_e.html](http://www.alos-restec.jp/index_e.html)

Quantity	Value
Image Size	4096 × 4096 pixels per channel, 8 bits/pixel raw
Channels (Wavelengths)	Blue: 0.42-0.50 $\mu\text{m}$ , Green: 0.52-0.60 $\mu\text{m}$ Red: 0.61-0.69 $\mu\text{m}$ , IR: 0.76-0.89 $\mu\text{m}$
Spatial Resolution	10m at Nadir
Swath Width	70km at Nadir

**Table 1.** Specifications of multispectral data used in experiments.



**Fig. 2.** ALOS-AVNIR-2 multispectral datasets 1 (left) and 2 (right).

data was intentionally chosen to contain various topographical features such as vegetation, ocean, coastline; city structures; weather conditions such as atmospheric haze and clouds. Table 2 reports the correlation coefficients among the pixels of the image channels in data sets 1 and 2. There is significant correlation among adjacent spectral bands (e.g. Blue and Green, or Green and Red) and reduced correlation as the frequency range corresponding to the different bands increases. Low correlation of all channels with the IR channel is partly attributed to the large separation of the IR spectrum from the end of the Red spectrum and the larger bandwidth of the IR sensor compared to the other sensors.

The rate-distortion performance of the proposed algorithm was compared against independent SPIHT and JPEG2000 coding of the channels. In all schemes, 3 levels of 2-D DWT are applied, using the reversible 5/3 wavelet transform. First, we compared the number of syndrome bits generated by distributed source coding with those generated by independent SPIHT coding. The percentage of the bit rate saved is plotted in Fig. 3 as a function of the number of bit planes used. The savings are computed for significance, refinement and sign bits generated by the distributed source coding scheme summed over the green, red and IR channels, where the blue channel is treated as side information. The plots show that, as there is high correlation in the most significant bits, the bit rate savings are large at high compression ratios, and diminish as finer details are added.

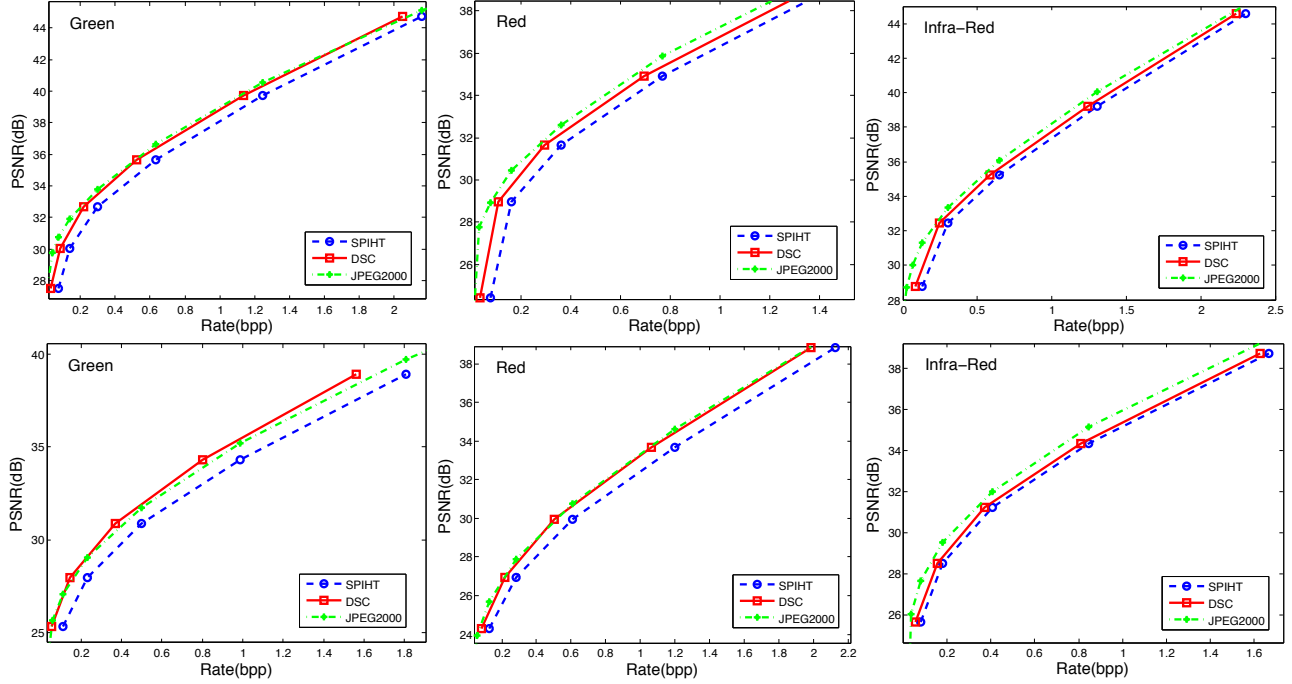
The PSNR metric is used to measure the quality of the decoded channels. The rate vs. PSNR tradeoff is plotted in Fig. 4 for Datasets 1 (top row) and 2 (bottom row). The performance is better than independent SPIHT coding and approaches that of independent JPEG2000 coding<sup>2</sup>. Additionally, the proposed algorithm has lower encoding complexity because its syndrome coding module is much simpler than the arithmetic coding module in JPEG2000.

<sup>2</sup>Compression of red given green, and IR given red showed no significant improvement in performance. Compression using multiple previously decoded channels as side information remains to be explored.

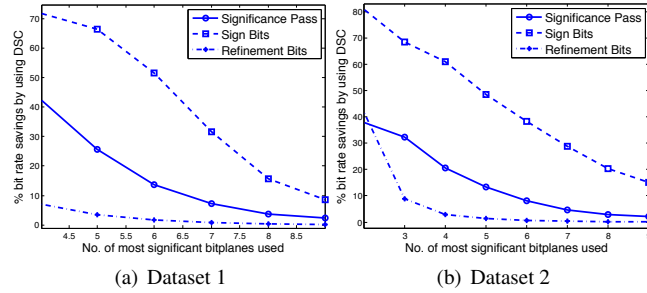
	G	R	IR
B	.904	.853	.654
G		.932	.595
R			.418

	G	R	IR
B	.949	.933	.567
G		.957	.612
R			.464

**Table 2.** Raw pixel domain inter-channel correlation coefficient for Data Set 1 (left) and 2 (right).



**Fig. 4.** Top Row: Dataset 1; Bottom Row: Dataset 2; Rate-distortion performance for distributed source coding is significantly better than independent SPIHT coding for all cases. In some cases, it approaches the R-D tradeoff of independent JPEG2000 coding at much lower complexity. In the distributed scheme, the blue channel is independently coded with SPIHT and is used as side information.



**Fig. 3.** Distributed source coding results in bit rate savings compared to independent coding with SPIHT. For fair comparison, the rate consumed by independent SPIHT coding of the blue channel is also included in the distributed coding bit rates.

#### 4. CONCLUSIONS

We presented a wavelet-domain distributed source coding algorithm based on SPIHT, in which wavelet zerotrees are described using a significance map, sign bits and refinement bits. Each of these three components is Slepian-Wolf-compressed using LDPC syndrome codes of appropriate length. Our experiments on multispectral images indicate that distributed coding of the significance map provides significant rate savings at high compression ratios, or equivalently for most significant bitplanes of wavelet coefficients. Several aspects of this algorithm invite further study; we are interested in developing side information-based prediction models for such schemes, evaluating the usage of multiple decoded images as side information, and optimally combining the zerotree-based approach with conventional distributed coding on bitplanes of transform coefficients.

#### 5. REFERENCES

- [1] D. Slepian and J. K. Wolf, "Noiseless Coding of Correlated Information Sources," *IEEE Trans. Information Theory*, pp. 471–480, July 1973.
- [2] A. D. Wyner and J. Ziv, "The Rate-Distortion Function for Source Coding with Side Information at the Decoder," *IEEE Trans. Information Theory*, vol. 22, pp. 1–10, Jan. 1976.
- [3] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," *Proceedings of the IEEE, Special Issue on Advances in Video Coding and Delivery*, vol. 93, no. 1, pp. 71–83, Jan. 2005.
- [4] X. Guo, Y. Lu, F. Wu, and W. Gao, "Distributed video coding using wavelets," in *Proc. IEEE International Conference on Circuits and Systems (ISCAS)*, Kos Island, Greece, sep 2006, pp. 5427–5430.
- [5] S. Li, S. Fang, and Z. Li, "Wyner-Ziv video coding for low bitrate using SPIHT algorithm," in *IEEE Workshop on Signal Processing Systems (SiPS)*, Shanghai, China, oct 2007, pp. 341–345.
- [6] J. M. Shapiro, "Embedded Image Coding using Zerotrees of Wavelet Coefficients," *IEEE Trans. Image Processing*, vol. 41, no. 12, pp. 3445–3462, Dec. 1993.
- [7] A. Said and W. Pearlman, "A New Fast and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 6, no. 3, pp. 243–250, June 1996.
- [8] D. Taubman and M. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards and Practice*, Kluwer Academic Publishers, 2002.
- [9] N-M. Cheung and A. Ortega, "An Efficient and Highly Parallel Hyperspectral Imagery Compression Scheme based on Distributed Source Coding," in *Proc. Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, Nov. 2006.
- [10] R. G. Gallager, "Low-Density Parity-Check Codes," *M.I.T. Press*, 1963.