

## Research Article

# A Doping Bits Based BP Decoding for Rateless Distributed Source Coding Applications

**N. M. Masoodhu Banu<sup>1</sup> and S. Sasikumar<sup>2</sup>**

<sup>1</sup> Department of Electronics and Communication Engineering, Sethu Institute of Technology, Virudhunagar, India

<sup>2</sup> Department of Electronics and Communication Engineering, R.M.D. Engineering College, Gummidipoondi Taluk, Thiruvallur District, Chennai, India

Correspondence should be addressed to N. M. Masoodhu Banu; banumobeen@yahoo.com

Received 15 September 2013; Revised 25 March 2014; Accepted 25 March 2014; Published 23 April 2014

Academic Editor: Sing Kiong Nguang

Copyright © 2014 N. M. Masoodhu Banu and S. Sasikumar. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A novel doping bits based belief propagation decoding algorithm, for rate-adaptive LDPC codes based on fixed bipartite graph code, is proposed. The proposed work modifies the decoding algorithm, by converting the puncturing nodes to regular source nodes and by following the encoding rule at the decoder. The transmitted doping bits in place of punctured bits, with the modified decoding algorithm at the decoder, feed all the punctured nodes with reliable log likelihood ratios. This enables the proposed decoding algorithm to recover all punctured nodes in the early iteration. The fast convergence leads to decoder complexity reduction while providing considerable improvement in performance.

## 1. Introduction

Conventional video coding technologies have been dominated by MPEG and H.26x family of standards, of which the most recent H.264/AVC [1] stands for the state of the art. They offer high quality, low bitrate video for demanding applications but at the cost of complex motion estimation and compensation for better rate-distortion performance. However, in some emerging applications such as low-power wireless video surveillance, wireless PC cameras, and mobile camera phone, low complexity encoding is required. Distributed source coding (DSC) [2, 3], a new paradigm in coding, which allows for sparingly low complexity encoding, is well matched for these applications as it shifts the encoding complexity to the decoder. Motion estimation [4] plays an important role in H.264 and MPEG4 video compression. It estimates the motion vector between the intra- and inter-frame and transmits only the motion vector for interframe, whereas the source and channel coded data are transmitted for intraframe. Motion vector estimation module is the one which increases the complexity of the encoder. Distributed video coding (DVC) based on DSC does not depend on

complex motion estimation. The core idea of DVC comes from some information theoretical results in 70s: Slepian-Wolf [5] and Wyner-Ziv [6] theorems. It states that if  $X$  and  $Y$  are two correlated sources,  $X$  can be encoded at a rate greater than or equal to its conditional entropy  $H(X | Y)$ . Though  $H(X | Y)$  is less than its entropy  $H(X)$ ,  $X$  can be decoded provided that  $Y$  was encoded at a rate  $H(Y)$  and is available at the decoder as side information. Wyner pointed out the possibility of using linear error correcting codes for Slepian-Wolf coding in [7]. As the practical DVC system needed high coding efficiency channel codes, the implementation was possible only after the invention of turbo and LDPC codes [8, 9] which reaches Shannon's capacity limit with longer length. Though the issue with high coding efficiency is ended with the invention of LDPC codes, the problem with the assumption of conditional entropy at the encoder remains. As the conditional entropy determines the rate of the code to be used for  $X$ , the transmitted syndrome bits based on presumption may or may not be sufficient to decode  $X$ . If the bit error rate (BER) at the decoder exceeds a given value, more syndromes are requested from the encoder. In this circumstance, the code designed should be incremental;

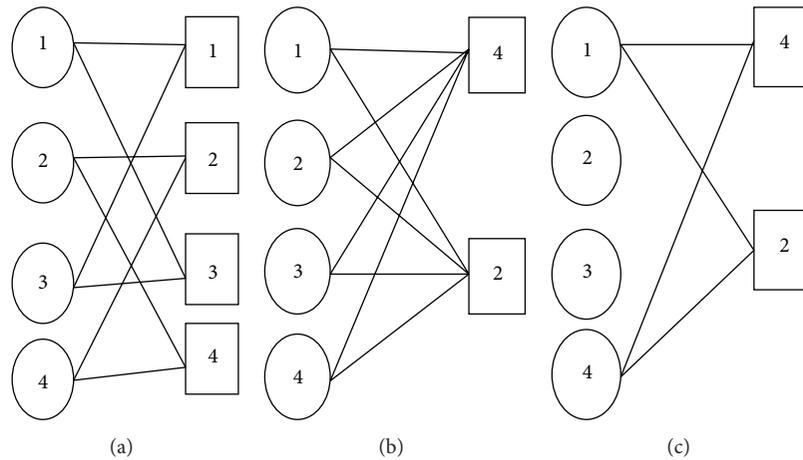


FIGURE 1: Decoding graphs if the encoder transmits (a) entire syndrome, (b) accumulated syndrome, and (c) even indexed syndrome bits.

that is, the encoder should not reencode the data. More precisely, the first bits are kept, and only additional bits are sent upon request and this is essentially done by rate-adaptive codes.

The concept of rate-adaptive codes was originated in the channel coding literature. One of the key techniques for achieving rate adaptivity is by puncturing, but it is done at a relatively small performance loss [10]. Ha et al. [10] investigated the existence of a good puncturing pattern in LDPC codes for rate adaptivity, but the length of the code was large. They also proposed a systematic way to find the locations of punctured symbols in short length LDPC codes [11], which is rate adaptive and minimizes the performance loss due to the puncturing. Hyo et al. [12] proposed an algorithm to prevent the formation of a stopping set from the punctured variable nodes even when the amount of puncturing is quite large. In spite of the existing works on rate-adaptive channel codes, the design of rate-adaptive LDPC code for Slepian-Wolf coding is a challenging task due to wide compression rates needed to achieve the theoretical compression limit. It has been studied extensively by various authors; Varodayan et al. [13] proposed a rate-adaptive scheme by sending doping source bits in addition to the source bits in a deterministic manner. They also proposed a new construction of codes [14] called LDPC accumulated (LDPCA) codes to fit the Slepian-Wolf coding. LDPCA code set is designed in syndrome splitting method keeping the size of the stopping sets. However, it does not eliminate the short cycles carefully for each member code and results in low girth [15]. Rate-adaptive PEG codes have been suggested by [16] which increase the girth to reduce the cycles. The common shortcoming of these codes is that they handle the condition of poor side information at the decoder by sending more syndromes on request with variable bipartite graph for each rate change.

Figure 1 demonstrates how the bipartite graph varies with each rate change; Figure 1(a) shows the tanner graph for the base code, whose rate is one, and Figures 1(b) and 1(c) for 1/2 rate code. Figure 1(b) combines the nodes connected to odd indexed syndrome with the even indexed one to achieve 1/2

rate code. Figure 1(c) manipulates even indexed syndromes to achieve 1/2 rate code, which leaves with unconnected source nodes as shown. It does not favor decoding as the source nodes are already available as side information at the decoder. The varying structure of the tanner graph as shown in Figures 1(a), 1(b), and 1(c) needs a large memory management at the encoder. If the same LDPCA codes have to be used for Slepian-Wolf theoretical compression limit, leading to variable source nodes, zeros have to be padded to the actual length of information bits. Hence not only does that LDPCA need more complex memory management, but also it does not support rate adaptivity for any variation in block length. Skipping highly correlated colocated blocks in block based video coding leads to variable block length and hence source nodes. Hence the rate-adaptive codes for Slepian-Wolf coding have to serve two purposes: handle variable source nodes and check nodes length.

Chen et al. [17] showed that, under density evolution, each Slepian-Wolf coding problem is equivalent to a channel coding problem for a binary-input output-symmetric channel. Based on this any rate-adaptive channel codes design can be directly converted to Slepian-Wolf code design. Jiang et al. have addressed the problem of variable tanner graph and presented a solution based on fixed tanner graph in [18, 19] by using the analytical results on punctured LDPC codes in [17, 20]. They also showed that any arbitrarily small gap to capacity for the original code can be preserved after puncturing on the binary erasure channel (BEC) and it is also shown that punctured LDPC codes are as good as ordinary LDPC codes. The issue of variable tanner graph is solved in [18] by designing the code with extra (20–25%) puncturing source and check nodes, which may or may not carry the information. This method uses the standard BP decoding to recover the extended punctured parity nodes leading to slow convergence. The main contribution of this paper is to develop an algorithm for [18] to speed up the convergence process in decoding and at the same time to give improved BER performance. The work presented here gives a new construction method to ease the code design and modifies

the decoding algorithm. Thereby it improves some results from [18], where an earlier version of the algorithm was presented. The developed algorithm is called doping bits based fixed tanner graph algorithm, since it is based on sending doping bits instead of extended parity bits on additional syndrome request. PEG constructed LDPC codes are used in [21] as bit plane processing of DSC application requires short length codes. The iterative belief propagation (BP) decoder [22] with estimated side information and doping bits leads to performance improvement in terms of decoder complexity and BER.

The rest of the paper is organized as follows. Some important definitions and notations used throughout this paper are given in Section 2 and the novel concept of doping bits based decoding algorithm is introduced in Section 3. Simulation results are discussed in Section 4 and finally in Section 5 conclusion is presented.

## 2. Definitions and Notations

The notations and definitions used throughout the paper are introduced here. LDPC codes are represented by tanner graph [23] and consist of two sets of nodes connected by edges; variable nodes correspond to the  $N$  columns of the parity-check matrix  $H$  and the check nodes  $M$  represent the rows of  $H$ . There are two ways to represent the LDPC for DSC application: (1) by  $(N, M)$ , where  $N$  (input) is the number of information bits and  $M$  (output) is the number of syndromes; that is, only the syndromes are transmitted (2) by  $(K, N - K)$ , where  $K$  (input) is the number of information bits and  $N - K$  (output) is the number of parity bits; that is, only parity bits are transmitted. The work proposed here is based on syndrome method. Since  $N$  variable nodes are used for information bits, from now on it is called source nodes. Apart from  $N$  source nodes and  $M$  syndromes as in usual scheme, there are  $L$  numbers of extended source nodes called parity nodes and extended syndrome nodes for rate adaption. Figure 2 shows the tanner graph of system described in [18]. Finally  $M_{eq} = M + L$  represents the total syndromes and  $N_{eq} = N + L$  represents total source nodes.

Source nodes are combination of information bits  $S_1$  to  $S_6$  and parity bits  $P_1$  to  $P_3$ . Check nodes are combination of check nodes  $C_1$  to  $C_3$  and extended check nodes  $EC_1$  to  $EC_3$ . Extended check nodes are equal in number to parity nodes.

## 3. Doping Bits Based Rate-Adaptive LDPC Codes

**3.1. Algorithm.** LDPC codes in syndrome code form have been used effectively in fixed-rate distributed source coding due to its relatively more compression ratio than parity based scheme. The BP decoding is modified slightly in order to take into account the syndrome information [22]. LDPC codes used as Slepian-Wolf codes have to be rate adaptive in two ways, with respect to variation in source length and variation in the quality of side information estimation. Filling extra source node with zeros is one of the ways to adapt variable source length, but it results in more encoding complexity.

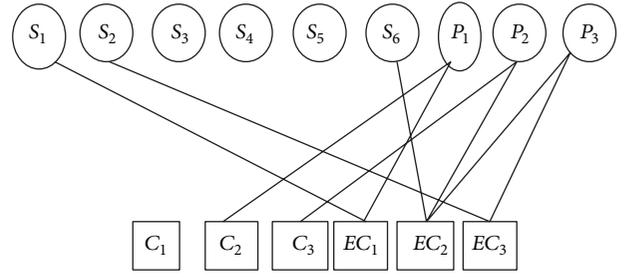


FIGURE 2: Code design by [18].

The design used in [18] gives solution for source length variation as well as variation in side information quality. Side information quality variation is handled by incremental syndromes. The encoding step includes

- (1) generation of parity sequence such that extended syndrome becomes zero;
- (2) generation of regular syndrome from the actual source using the generated parity sequence from step 1;
- (3) mixing a variable portion of the source bits with the generated parity sequence according to a predefined arithmetic operation;
- (4) in addition to this, the proposed scheme sends uncoded source bits called doping bits according to the desired compression rate. These bits are called doping bits for the reason explained later in the decoding section.

A different decoding method at the decoder facilitates these doping bits. Computation of parity sequence, that is,  $P_1$  to  $P_L$ , involves solving  $L$  linear equations associated with the corresponding check nodes. The basic idea is that, by way of modified decoding, the punctured parity node becomes reliable and assists the decoding process to converge fast in early iteration. The pattern of the doping bits over the source nodes is known to the decoder. There are two schemes to be explained: variable source length and fixed syndrome coding and fixed source length and variable syndrome coding. The supporting rate for Scheme A is  $M/N$  to  $M/N_{eq}$  and for Scheme B is  $M/N$  to  $M_{eq}/N_{eq}$ . The doping technique is applied when there is a requirement of additional syndrome from the decoder.

**3.1.1. Scheme A: Variable Source Length and Fixed Syndrome Coding.** This scheme comes into picture when the source statistics change to give variations in source length. The proposed scheme punctures the extended parity bits according to the compression ratio and estimates the punctured parity bits using the modified decoding scheme at the decoder, whereas the algorithm described in [18] does not do estimation. Estimation makes punctured node a reliable node and hence the LLR computed for these bits are highly reliable. The reliable punctured nodes based on modified decoding lead

to early convergence at the decoder. Additional  $s_{N+i}$  number of source is sent for variable source length coding as

$$e_i = P_i + s_{N+i}, \quad \text{where } i = 0 \text{ to } L. \quad (1)$$

**3.1.2. Scheme B: Variable Source Length and Variable Syndrome Coding.** This scheme is applied whenever the decoder requests for supplementary incremental syndrome due to poor side information estimation. For every additional syndrome requested, instead of sending the parity bits  $P_1$  to  $P_L$ , the encoder sends uncoded source bits to the decoder. This combined with the estimated  $P_1$  to  $P_L$  at the decoder improves the performance and also makes the decoding fast. In a nutshell the role of the rate-adaptive decoder is to recover the source  $X$  from the uncoded source bits and syndrome bits so far received from the encoder in conjunction with the block-candidate side information  $Y$  for any data length and syndromes. As this uncoded source enriches the performance, they are called doping bits.

**3.1.3. Decoding.** The key to the proposed solution is the modified decoding algorithm which follows the encoding steps. It differs from [18] by the way of parity nodes computation and BP decoding for the doping bits at the decoder. Once the parity nodes are used to find the extended zero syndromes, it is used to find the other nonzero syndromes in combination with the other source symbols. The encoding part is elucidated first in order to appreciate the decoding. For simplicity (8, 4) code is assumed where the punctured nodes are 20%, that is, 2 numbers. The new code dimension is (10, 6). That is, 2 syndrome nodes must be made zeros. The Tanner graph is drawn partially in Figure 3, in order to highlight only the zero syndrome nodes computation. In Figure 3,  $EC_1$  and  $EC_2$  are zero syndrome nodes and  $P_1$  and  $P_2$  are parity nodes. The equation for the zero syndrome nodes at the encoder is given by

$$\begin{aligned} EC_1 &= P_1 \oplus S_6, \\ EC_2 &= P_2 \oplus S_3. \end{aligned} \quad (2)$$

At the encoder  $P_1$  and  $P_2$  are computed such that  $EC_1$  and  $EC_2$  are zeros, with  $S_6$  and  $S_3$  are known. The same can be reversed at the decoder as the core idea of DSC coding is that the side information is correlated with the source being encoded. Hence at the decoder the equation to find the parity bits is given by

$$\begin{aligned} P_1 &= EC_1 \oplus S_6, \\ P_2 &= EC_2 \oplus S_3. \end{aligned} \quad (3)$$

As punctured nodes are computed to have value, it is not fed with zero as LLR. This modification would leverage the punctured/parity node's reliability and hence the decoding performance. This transforms the intentional puncturing with heterogeneous parallel channel (BSC/BEC) parameters [20] to homogeneous channel. The strengthened LLR values using the modified algorithm, at the decoder, lead to fast convergence.

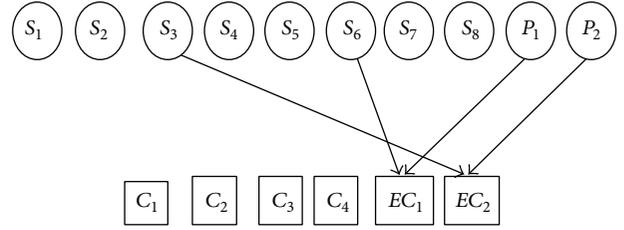


FIGURE 3: Illustrating modified decoding with punctured nodes estimation.

Eliminating the punctured parity nodes as described above leaves space to send some other bits in place of extended syndromes and these bits are called doping bits. As the doping bits are 100% reliable source bits, some modification can be done in BP decoding; that is, the message from doping bits node to check nodes is unaltered and always will be

$$p_s^{\text{out}} = 1 - D, \quad (4)$$

where  $p_s^{\text{out}}$ , the output, is message from source node to check nodes and  $D$  is the doping bit [13]. This in turn not only converges fast, but also reduces the computational complexity.

**3.2. Code Design.** Exclusive properties of the code design needed by system in [18] on comparison with the standard LDPC code are that (1) the extended source nodes and extended check nodes should be placed at the end in order to allow for puncturing within binary erasure channel; (2) in addition to the extended source nodes, other source nodes connected to the extended check nodes should have one to one connection as shown in Figure 3. Here the parity bits are directly connected to extended check nodes and only few other source nodes are connected to make the computation easier. If  $EC_2$  had been connected to two parity nodes as in Figure 2, then equation for  $EC_2$  would change to

$$\begin{aligned} EC_1 &= P_1 \oplus S_1, \\ EC_2 &= P_2 \oplus P_3 \oplus S_6, \\ EC_3 &= P_3 \oplus S_2. \end{aligned} \quad (5)$$

$P_1$ ,  $P_2$ , and  $P_3$  are computed in a way that,  $EC_1$ ,  $EC_2$ , and  $EC_3$  are zeros in each input cycle. Solving these equations will be computationally intensive, if the above three equations are not independent. It is very hard to place both extended source nodes and syndrome nodes at the end with independent check node equation for a specified BER parameter. As the scheme is based on fixed tanner graph, extended syndrome nodes can be placed anywhere with the only restriction of having independent check node equations. Selecting extended syndromes in this way opens too many choices, which is otherwise not possible with direct design. The position of the extended syndrome nodes has to be known by the decoder in this case. This is not very complicated,

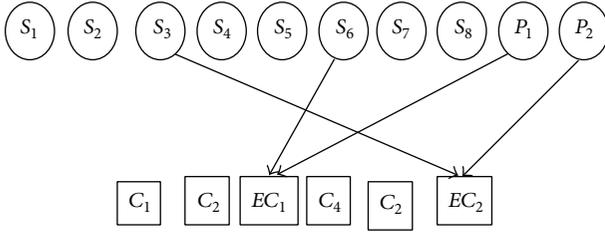


FIGURE 4: Proposed construction.

as it is one time process. The new construction is shown in Figure 4.

Selection of the number of codes has been explained with an example by taking a QCIF video. The total number of blocks with the block size of  $4 \times 4$  for a QCIF ( $176 \times 144$ ) video is 1584. The number of source node when processed in bit planes without compression will be 1584. Additional example is  $512 \times 512$  hyperspectral image [24] which also can be DSC coded due to the high correlation between neighboring bands.

In skip mode the length of the source node changes, due to skipping blocks which are highly correlated with the colocated blocks in the previous frame. This can be further explained with the help of mother-daughter video, taking  $Y$  component of 3rd and 4th frames and hyperspectral image of 28th and 29th bands. As shown in Figures 5(c) and 5(d), since the grandma sequence is a slow motion video, all the blocks are almost the same and the PSNR metric between them is more than 45 db. The difference between the frames is only due to the camera lighting condition. Skipping these blocks for coding reduces the source bits for LDPC coding. Changing the PSNR of the skip blocks for a specified rate distortion performance also will change the number of source bits for coding. Another factor contributing to variable source node is quantized DCT coefficients which can be discarded due to its low magnitude. Apart from the above, variations in correlation with every adjacent frame of a video lead to variable source length. Due to the similarity of the decoding algorithm, the degree distributions optimized for LDPC channel codes (e.g., using [25]) can be applied directly to the proposed rate-adaptive codes.

## 4. Results and Discussion

The performance of doping bits based rateless code is analyzed on the basis of storage complexity, encoder-decoder complexity, and BER performance. The experimental setup parameters are as follows.

- (i) The channel is modeled as multiple BSC channel in order to incorporate the precise correlation between each bit plane independently [26].
- (ii) The channel code for the proposed system has been designed for BSC channel with 20% punctured or extra source nodes using density evolution. The threshold cross over probability of the resulting code is 0.12.

TABLE 1: Comparison of number of codes required for the proposed scheme with other codes.

Code dimension for other rate-adaptive codes	Code dimension for the proposed scheme
(500, 600)	(600, 350)
(600, 700)	(720, 420)
(700, 800)	(864, 504)
(800, 900)	(1036, 605)
(900, 1000)	(1244, 726)
(1000, 1100)	(1493, 871)
(1100, 1200)	—
(1200, 1300)	—
(1300, 1400)	—
(1400, 1500)	—
(1500, 1600)	—

- (iii) The construction used for the code design is standard and PEG codes.
- (iv) The virtual channel correlation was modeled as BSC/BEC when punctured nodes are involved and BSC channel when there is an estimation of punctured nodes.
- (v) Maximum source length considered is 1584 and the minimum one considered is 500 as explained in Section 3.
- (vi) A set of 5 codes has been designed with base code rate  $1/2$ , within the range 500–1584, as in Table 1.

**4.1. Storage Complexity.** There is no difference between the number of codes used in the proposed scheme and reference [18], but, to appreciate the advantage of the proposed scheme with the rate-adaptive codes which do only syndrome adaption, Table 1 compares the number of codes. The number of codes with other rate-adaptive codes for a source length variation from 500 to 1584 is 11 codes, whereas it is only 6 codes for the proposed algorithm. Also the resolution step is 100 for regular codes and 1 for the proposed algorithm.

**4.2. Complexity of Encoder and Decoder.** Encoder complexity is given by  $S = x * H$  where  $S$  is syndrome,  $x$  is information bits, and  $H$  is parity check matrix. Let  $j$  and  $k$  be the source and check nodes degrees, respectively; it means that  $k$  number of source nodes is connected to each check nodes. As syndromes are nothing, but check nodes value, each syndrome computation needs  $k$  number of multiplications (MULs) and  $k - 1$  xor additions. There is no encoder complexity saving as such, but the design methodology always results in one-to-one relationship between extended parity nodes and extended syndrome nodes. This gain is achieved by interleaved extended syndrome nodes.

The figures, given in Tables 2 and 3, illustrate the number of iterations required for (1800 and 1050) rate-adaptive code, where the number of extended data and syndromes is 300. Decoder complexity reduction is achieved by modified



FIGURE 5: (a) and (b) 28th and 29th bands of hyperspectral image. (c) and (d) Y component of 3rd and 4th frames of grandma video.

TABLE 2: Number of iterations required for Scheme A.

Number of punctured parity nodes	Number of extended data	Iteration needed by reference [18]	Iteration needed by modified decoding
0	300	10	10
150	150	30	25
300	0	50	38

TABLE 3: Number of iterations required for Scheme B.

Number of punctured parity nodes	Number of extended syndromes or number of doping bits	Iteration needed by reference [18]	Iteration needed by doping bits along with modified decoding
0	300	10	4
150	150	21	10
300	0	50	38

decoding. For Scheme A, the gain comes by making the punctured nodes as nodes with certainty. For Scheme B, the gain is due to two factors: conversion of punctured nodes to nodes with certainty and high confidence LLR values of doping bits. Accordingly the improvement for item 2 in Table 2 is only due to modified decoding and hence a moderate improvement of only 16.66%, whereas the improvement for item 2 in Table 3 is due to two factors. They are modified decoding for 50% of the nodes and 50% of doping bits in place of extended syndromes.

**4.3. Bit Error Rate Performance.** In this section the performance of Scheme A and Scheme B is compared for various rates and with that of rate-adaptive LDPC codes of [18] for distributed source coding of two binary correlated sources. Simulation parameters are listed in Table 4. The virtual correlation channel is modeled as BSC due to bit plane coding. The variable conditional entropy was simulated by adding noise with various cross over probabilities with the motion compensated frame 1.

TABLE 4: Simulation parameters for rate compatibility.

Base code rate	(1500, 750)
Derived code rate for use as rate-adaptive codes	(1800, 1050) ( $L = 20\%$ of $N$ ) = 300; $N = 1500$ ; $M = 750$ . $N_{eq} = N + L = 1800$ . $M_{eq} = M + L = 1050$ .
Source $X$	Bit planes of frame 2 of grandma video sequence.
Source $Y$	Bit planes of frame 1 of grandma video sequence.
Side information	Motion compensated Frame1 at the decoder.
Rate considered for Scheme A	(1) $750/1500 = 0.5$ Punctured nodes = 300.
Rate = $M/N_i$ where $i = 0$ to 300	(2) $750/1600 = 0.46875$ Punctured nodes = 200. Extended source nodes = 100.
Rate considered for Scheme B	(1) $1000/1500 = 0.666$ ; Punctured nodes = 50 Doping nodes = 250 (extended syndromes) Extended source nodes = 0.
Rate = $M/N$ to $M_{eq}/N_{eq}$	(2) $900/1600 = 0.5625$ Punctured nodes = 50 Doping nodes = 150 (extended syndromes) Extended source nodes = 100

Figure 6 shows the BER performance comparison under variable source length and fixed syndrome (Scheme A). The rates are modified by changing the number of source nodes, keeping the syndromes constant. The performance improvement in low rate codes, due to less number of punctured nodes, is almost evenhanded by the less number of zero syndromes. Rate 0.5 and 0.4685 codes have only a small performance gap for both the proposed and reference [18] schemes and this too goes unseen at high correlation as punctured nodes converge faster with high correlated side information. However, comparatively, the higher gain of the proposed scheme is due to the estimation of punctured nodes at the decoder; that is, the proposed scheme reaches BER of  $10^{-5}$  at 0.06 cross over probability, whereas reference [18] scheme reaches only at 0.025 cross over probability.

Figure 7 shows the BER performance comparison under variable source length and syndrome (Scheme B). The rates are modified by changing the number of syndrome nodes and source length. High rate codes perform better as the doping bits dole out as requested extra syndromes in addition to modified decoding. The performance gap remains increasing as the number of doping bits increases.

Performance of the proposed Scheme A and Scheme B is compared with that of reference [18] in Figure 8. The rate 0.46875 code was used for comparison. Scheme A with estimated punctured node performs better than that of [18].

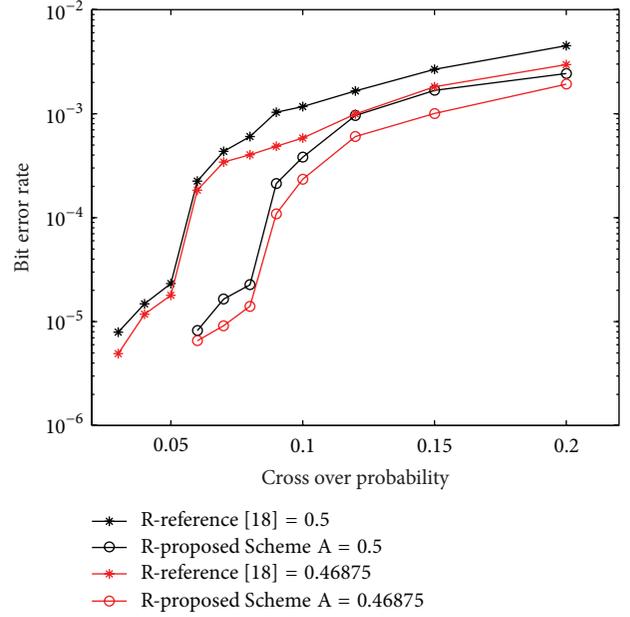


FIGURE 6: Scheme A: comparison of BER with cross over probability for rate adoption.

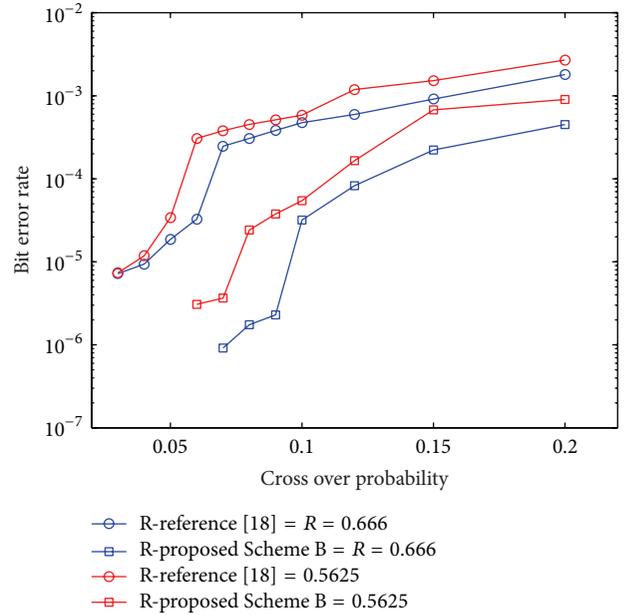


FIGURE 7: Scheme B: comparison of BER with cross over probability for rate adoption.

Due to additional syndrome the rate becomes 0.5625 for Scheme B and unsurprisingly it shows better performance than any other schemes because of doping bits. Figure 9 compares the performance of LDPC codes constructed by standard algorithm and PEG construction algorithm, as expected PEG based codes outperforms standard one due to high girth.

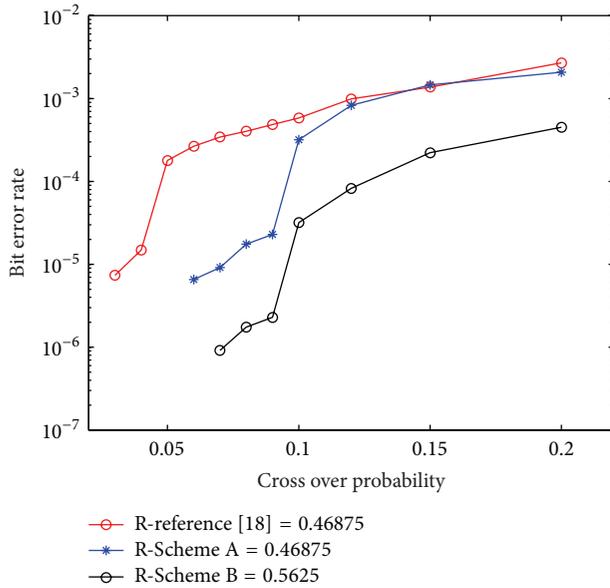


FIGURE 8: Performance comparison of Schemes A and B.

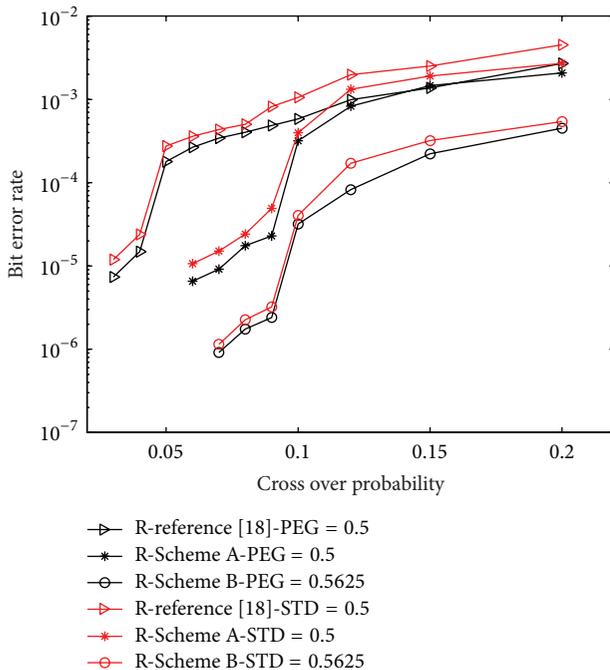


FIGURE 9: Performance comparison of PEG and standard codes.

## 5. Conclusion

Modified decoding algorithm presented in this paper converts the punctured nodes to regular source nodes. The conversion changes the original BSC/BEC parallel channel to BSC/BSC homogeneous channel and hence improves decoding performance. Doping bits in place of additional syndromes as a result of estimated punctured node play a crucial role in decoding complexity reduction and BER performance. An experimental result also shows that

the proposed scheme effortlessly integrates the skip block based compression technique with DSC coding.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## References

- [1] J. Ostermann, J. Bormans, P. List et al., "Video coding with H.264/AVC: tools, performance, and complexity," *IEEE Circuits and Systems Magazine*, vol. 4, no. 1, pp. 7–28, 2004.
- [2] S. S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): design and construction," *IEEE Transactions on Information Theory*, vol. 49, no. 3, pp. 626–643, 2003.
- [3] A. Aaron, R. Zhang, and B. Girod, "Wyner-Ziv coding of motion video," in *Proceedings of the 36th Asilomar Conference on Signals Systems and Computers*, pp. 240–244, Pacific Grove, Calif, USA, November 2002.
- [4] N. Brady, "MPEG-4 standardized methods for the compression of arbitrarily shaped video objects," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 8, pp. 1170–1189, 1999.
- [5] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Transactions on Information Theory*, vol. 19, no. 4, pp. 471–480, 1973.
- [6] A. D. Wyner and J. Ziv, "Rate-distortion function for source coding with side information at the decoder," *IEEE Transactions on Information Theory*, vol. 22, no. 1, pp. 1–10, 1976.
- [7] A. D. Wyner, "Recent results in the shannon theory," *IEEE Transactions on Information Theory*, vol. 20, no. 1, pp. 2–10, 1974.
- [8] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Transactions on Information Theory*, vol. 45, no. 2, pp. 399–431, 1999.
- [9] S.-Y. Chung, G. David Forney Jr., T. J. Richardson, and R. Urbanke, "On the design of low-density parity-check codes within 0.0045 dB of the Shannon limit," *IEEE Communications Letters*, vol. 5, no. 2, pp. 58–60, 2001.
- [10] J. Ha, J. Kim, and S. W. McLaughlin, "Rate-compatible puncturing of low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 50, no. 11, pp. 2824–2836, 2004.
- [11] J. Ha, J. Kim, D. Klinc, and S. W. McLaughlin, "Rate-compatible punctured low-density parity-check codes with short block lengths," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 728–738, 2006.
- [12] Y. P. Hyo, W. K. Jae, S. Kwang, and C. W. Keum, "Efficient puncturing method for rate-compatible low-density parity-check codes," *IEEE Transactions on Wireless Communications*, vol. 6, no. 11, pp. 3914–3919, 2007.
- [13] D. Varodayan, Y.-C. Lin, and B. Girod, "Adaptive distributed source coding," *IEEE Transactions on Image Processing*, vol. 21, no. 5, pp. 2630–2640, 2012.
- [14] D. Varodayan, A. Aaron, and B. Girod, "Rate-adaptive distributed source coding using low-density parity-check codes," in *Proceedings of the 39th Asilomar Conference on Signals, Systems and Computers*, pp. 1203–1207, Pacific Grove, Calif, USA, November 2005.

- [15] J. Fan, Y. Xiao, and K. Kim, "Design LDPC codes without cycles of Length 4 and 6," *Journal of Electrical and Computer Engineering*, vol. 2008, Article ID 354137, 5 pages, 2008.
- [16] M. Jang, J. W. Kang, and S.-H. Kim, "A design of rate-adaptive LDPC codes for distributed source coding using PEG algorithm," in *Proceedings of the IEEE Military Communications Conference (MILCOM '10)*, pp. 277–282, San Jose, Calif, USA, November 2010.
- [17] J. Chen, D. He, and A. Jagmohan, "Slepian-Wolf code design via source-channel correspondence," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT '06)*, pp. 2433–2437, Seattle, Wash, USA, July 2006.
- [18] J. Jiang, D. He, and A. Jagmohan, "Rateless Slepian-Wolf coding based on rate adaptive low-density-parity-check codes," in *Proceedings of the IEEE International Symposium on Information Theory (ISIT '07)*, pp. 1316–1320, Nice, France, June 2007.
- [19] D.-K. He, A. Jagmohan, L. Lu, and V. Sheinin, "Wyner-Ziv video compression using rateless LDPC codes," in *Visual Communications and Image Processing*, Proceedings of the SPIE 6822, San Jose, Calif, USA, January 2008.
- [20] H. Pishro-Nik and F. Fekri, "Results on punctured low-density parity-check codes and improved iterative decoding techniques," *IEEE Transactions on Information Theory*, vol. 53, no. 2, pp. 599–614, 2007.
- [21] X.-Y. Hu, E. Eleftheriou, and D. M. Arnold, "Regular and irregular progressive edge-growth tanner graphs," *IEEE Transactions on Information Theory*, vol. 51, no. 1, pp. 386–398, 2005.
- [22] A. D. Liveris, Z. Xiong, and C. N. Georghiades, "Compression of binary sources with side information at the decoder using LDPC codes," *IEEE Communications Letters*, vol. 6, no. 10, pp. 440–442, 2002.
- [23] R. M. Tanner, "Recursive approach to low complexity codes," *IEEE Transactions on Information Theory*, vol. 27, no. 5, pp. 533–547, 1981.
- [24] "Hyperspectral Images of natural scenes 2004," [http://personalpages.manchester.ac.uk/staff/david.foster/Hyperspectral\\_images\\_of\\_natural\\_scenes\\_04.html](http://personalpages.manchester.ac.uk/staff/david.foster/Hyperspectral_images_of_natural_scenes_04.html).
- [25] "LTHC: LdpcOpt," 2001, <http://ipgdemos.epfl.ch/ldpcOpt>.
- [26] M. Vaezi and F. Labeau, "Improved modeling of the correlation between continuous-valued sources in LDPC-based DSC," in *Proceedings of IEEE 46th Asilomar Conference on Signal, System and Computers*, pp. 1659–1663, Pacific Grove, Calif, USA, November 2012.



**Hindawi**

Submit your manuscripts at  
<http://www.hindawi.com>

