LDPCA Code Construction for Slepian-Wolf Coding

Jeffrey J. Micallef, Reuben A. Farrugia, and Carl J. Debono

***Abstract*—Error correcting codes used for Distributed Source Coding (DSC) generally assume a random distribution of errors. However, in certain DSC applications, prediction of the error distribution is possible and thus this assumption fails, resulting in a sub-optimal performance. This letter considers the construction of rate-adaptive Low-Density Parity-Check (LDPC) codes where the edges of the variable nodes receiving unreliable information are distributed evenly among all the check nodes. Simulation results show that the proposed codes can reduce the gap to the theoretical bounds by up to 56% compared to traditional codes.**

***Index Terms*—Distributed source coding, rate-adaptive LDPC codes, Slepian-Wolf theorem.**

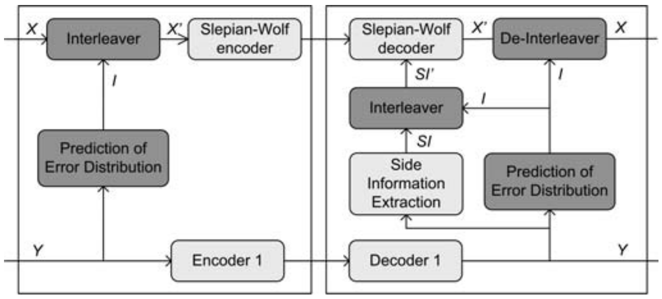


Fig. 1. Proposed architecture used for asymmetric DSC.

I. Introduction

In asymmetric Distributed Source Coding (DSC), two correlated sources, X and Y are encoded independently and then jointly decoded to achieve compression. Source Y is compressed with traditional coding techniques and used at the decoder to estimate source X. According to Slepian-Wolf (SW) theorem [1], if this estimate is used as a Side Information SI to aid decoding, the original source X can be losslessly recovered at rates $R \geq \mathrm{H}(X|SI)$ even if SI is not known at the encoder. Wyner [2] proposed that such compression can be achieved using channel coding techniques, where source X is compressed into its syndrome representation S, and SI is used to choose the most likely value of X from the other coset elements represented by S. Various implementations of the system using robust error correcting codes proved the validity of this approach [3]–[5], with LDPC Accumulate (LDPCA) codes considered in [5] performing best. These codes consist of an LDPC syndrome-former followed by an accumulator, where the check nodes of the LDPC code are split to form a stronger code each time decoding fails. Later on, in [6], these codes were improved by considering check node merging techniques, where the highest rate code serves as the base code and the lower rate codes are obtained by merging check nodes.

Our previous work shows that these codes can provide only a sub-optimal solution for certain DSC applications, where the dependency error can be predicted to a certain extent [7]. Such predictions were exploited to ensure that the edges of the variable nodes receiving unreliable information can be biased so that they are distributed evenly among all the check nodes at the base code, significantly improving the coding efficiency when the error prediction is accurate. Biasing of edges was also considered in [8], where the connections of the inner Luby Transform (LT) codes for Raptor codes were biased instead. This letter extends on our previous work and considers a biased code construction where the unreliable information is distributed uniformly even among the lower rate check nodes. Furthermore, it considers an irregular degree distribution to further improve Unequal error Protection (UeP) and protect the unreliable variable nodes more than the rest. Results show that the proposed code construction can reduce the gap to the SW bounds by up to 56% compared to the traditional codes in [6] and up to 50% compared to our previous codes in [7].

This letter is organized as follows: Section II and Section III summarize the DSC architecture that is being adopted and the features that must be considered during the construction of the proposed LDPCA codes. Section IV analyzes the performance of the proposed codes, while Section V concludes the letter.

II. Features for LDPCA Code Construction

The DSC architecture used in this work is illustrated in Fig. 1 and was presented in [7]. It introduces a *prediction of error distribution* module at both the encoder and decoder to predict the reliability of SI bits. This information is then used to place the unreliable bits at the beginning of the codeword to be encoded. This architecture is suitable for DSC applications where the dependency error is predictable, such as in Sensor Networks, where each sensor node encodes its data with respect to the correlated information of the entire network [9] and in Distributed Video Coding, where the reliability of SI can be predicted from the previously decoded bit planes [10] or from the difference between the adjacent key frames [11].

The designed DSC architecture assumes that most of the unreliable bits can be interleaved at the beginning of the codeword, while the remaining undetected errors are distributed randomly across the rest of the codeword. The code construction procedure can therefore distinguish between: i) the *unreliable variable nodes* at the beginning of the codeword and ii) the remaining *reliable variable nodes* which receive more reliable bits. In [7], the performance was improved by allowing each check node at the base code to connect to a

Manuscript received June 9, 2011. The associate editor coordinating the review of this letter and approving it for publication was V. Stankovic.

The research work is partially funded by STEPS-Malta and by European Funds–ESF 1.25.

The authors are with the Department of Communications and Computer Engineering, University of Malta (e-mail: jeffrey.micallef@ieee.org, reuben.farrugia@um.edu.mt, c.debono@ieee.org).

Digital Object Identifier 10.1109/LCOMM.2011.080811.111232

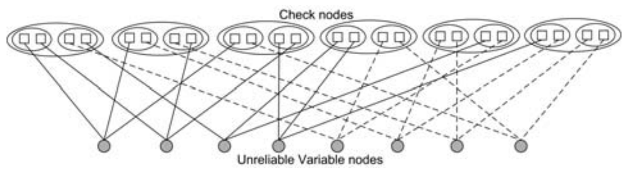


Fig. 2. LDPCA code considering unreliable variable nodes.

limited number of edges coming from the unreliable variable nodes as illustrated in Fig. 2. This figure considers only the edges of the unreliable variable nodes, whereas the other check node edges are randomly connected to the remaining variable nodes. The ovals at the top represent the way the check nodes are accumulated to form the lower rate sub-graphs.

It can be noticed that each check node receives only one unreliable edge at a time. However, this uniformity is not maintained for the lower rate sub-codes. For example, if only the first four variable nodes are in error, the first and third check nodes of the lowest rate graph (outer ovals) receive a lot of unreliable information compared to the rest. Decoding with the lowest rate graph might therefore fail, because the extrinsic information given by these check nodes fails to converge correctly. Yet, all the check nodes, even those receiving reliable edges, are split requesting unnecessary syndrome bits which degrades compression efficiency. Hence, the unreliable edges are sub-optimally arranged across the available check nodes.

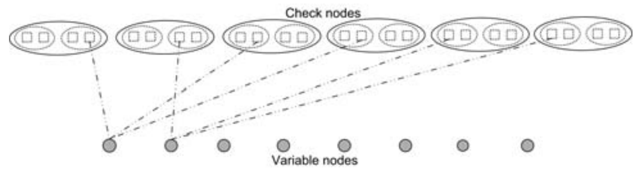
III. Proposed LDPCA Code Construction

The prediction of the dependency error is used to ensure that, as the errors accumulate at the beginning of the codeword, the unreliable edges are distributed more uniformly across all the lower rate check nodes. As in [6], the highest rate graph is considered as the base code and is built by considering one variable node at a time, starting from the leftmost unreliable nodes. The neighboring check nodes are selected as follows:

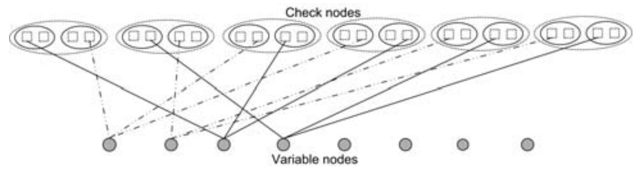
Step 1: Initially all the check nodes are grouped into different puncturing periods Δ, each representing the check nodes that are merged to form the lowest rate check nodes, as shown by the outer ovals in Fig. 3(a). The edges from the leftmost variable nodes are then distributed across these groups, considering a random check node within each group.

Step 2: When all these groups are considered once, each group is divided in half as shown in Fig. 3(b), such that they will now represent the check nodes of a higher rate graph. The edges of the next variable nodes are then connected to a check node found within the newly created groups, considering each group only once and distributing them uniformly between the groups available within the different puncturing period (outer oval). A higher preference is given to the check nodes found within the groups that are split first, such that the first syndrome bits can be used to improve the error correcting capability of the variable nodes on the left (which are more likely to be in error). Whilst selecting the best choice according to these preferences, it is also ensured that the edges do not form 4-lengthed cycles consisting of 2-degree variable nodes and cause no loss of edges at any lower rate sub-code.

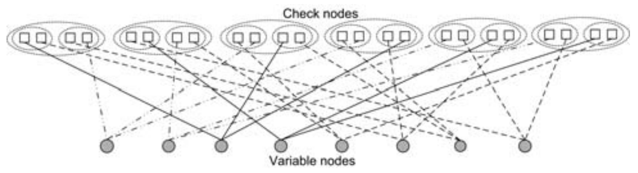
Step 3: The next variable nodes are repeatedly connected as in step 2, where the check node groups are split every time



(a) Distribution of unreliable edges across the lowest rate sub-code graph.



(b) Distribution of unreliable edges across a higher rate graph.



(c) LDPCA codes with proposed edge biasing.

Fig. 3. Proposed LDPCA code construction.

there are no more empty groups to consider. This procedure continues until all the highest rate check nodes are considered once, so as to obtain the same base code in [7] (See Fig. 3(c)).

The rest of the graph, which deals with more reliable variable nodes, is then built as proposed in [6]; i.e. considering a graph conditioning algorithm for the highest rate base code, whilst avoiding loss of edges and harmful structures (such as 4-lengthed cycles affecting 2-degree variable nodes) at all the lower rate sub-code graphs. The graph conditioning algorithm in [12] was preferred since this considers the connectivity of the cycles beside their length, hence avoiding longer cycles with poor graph connectivity. The proposed code construction assigns the highest possible degrees to the unreliable variable nodes and distributes the lower degrees across the remaining variable nodes, to improve the level of protection offered to the unreliable variable nodes compared to the rest [13].

This code construction ensures that the unreliable edges are distributed evenly across the lower rate check node even when only some of the first variable nodes are in error. For example, when the code in Fig. 3(c) receives only the first two bits in error, the check nodes at the lowest rate (other ovals) receive one unreliable edge each. Meanwhile, if the first four bits are in error, the outer ovals receive two unreliable edges each, which are divided such that each of the inner ovals receives one unreliable edge. The same occurs when all the first eight bits are in error. Furthermore, since most of the check nodes receive the same amount of unreliable information, the error correcting capability of the different parts of the graph needs to be improved at the same time. This avoids transmission of unnecessary syndrome bits and improves compression.

IV. Experimental Results

An LDPCA code with a degree distribution of $\lambda(x)$=0.3x+0.4x^2+0.3x^3 and a codeword length of 396 bits is constructed as discussed in Section III. Source Y' is considered as a random sequence of 396 binary symbols,

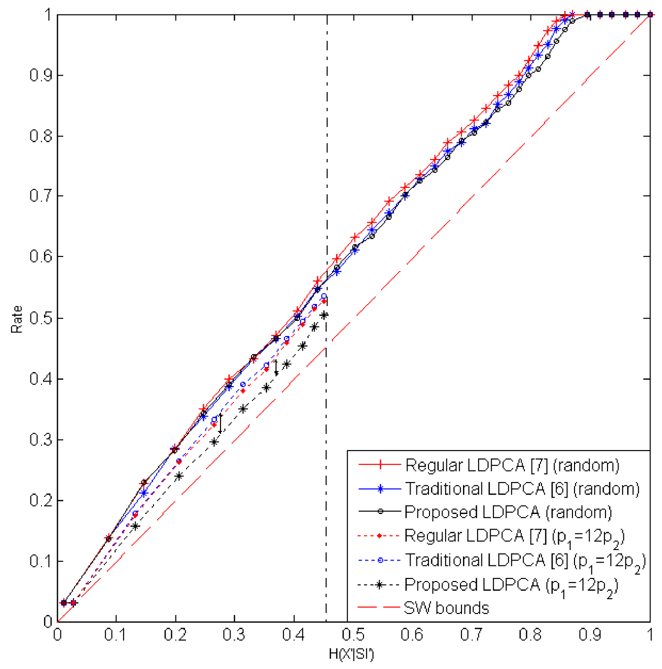


Fig. 4. Performance obtained with different LDPCA codes.

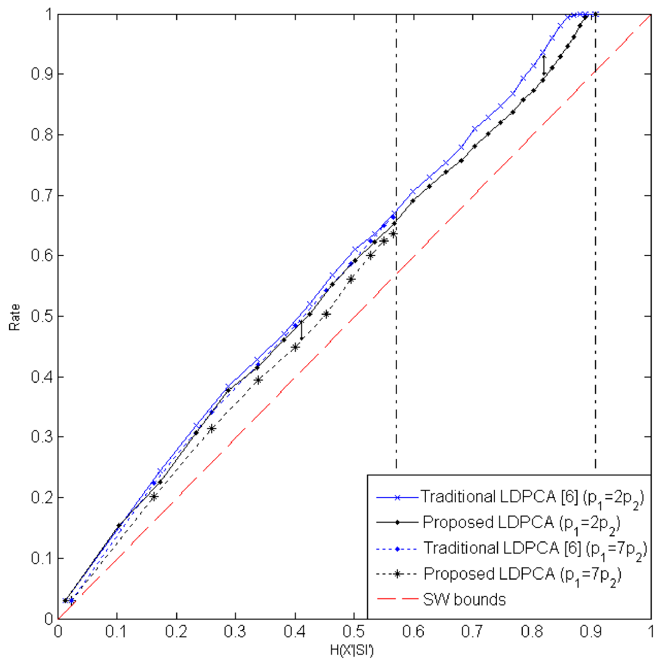


Fig. 5. Performance obtained with different LDPCA codes.

whilst the corresponding SI' is obtained from Y' by flipping some bits according to a randomly generated error pattern. The error pattern is generated such that the first 99 bits have a higher probability of error at the beginning of the codeword, with an average probability of error p_1, whilst the rest of the codeword has an average probability of error p_2 with $p_1 > p_2$.

Sequence SI' is then converted into soft-input information according to the distribution of errors employed, and fed into the LDPCA decoder together with a subset of the syndrome information generated for Y'. The decoder can determine whether the source bits have been decoded successfully or not and tries to correct SI' for a maximum of 50 iterations before requesting other syndrome information. Decoding is assumed to be successful after receiving all the syndrome information, since the source bits can be recovered using linear algebra [5].

The performance of the proposed code, taking the average rate of 200 trials for every entropy point, is shown in Fig. 4. The conditional entropy H($X'|SI'$) was calculated by considering the appropriate distribution of errors for the two different parts of the codeword. Fig. 4 studies the performance of the code for the two extremes: i) when error prediction fails such that errors are randomly distributed across the whole codeword (p_1=p_2) and ii) when the interleaver arranges the codeword such that p_1=$12p_2$, i.e. 80% of the errors are interleaved to affect the first part of the codeword. For comparison purposes, the performance of the traditional codes found in [6] and that of our previous codes in [7] were also considered for the same conditions and plotted in Fig. 4. The vertical line indicates the maximum entropy considered in each case, due to the a priori knowledge (error ratio p_1:p_2) assumed by the *prediction of error distribution* module (error pattern generator).

Fig. 4 illustrates that when the prediction of error distribution fails, the performance of the code converges to that of the traditional LDPCA codes, which is still better than that of the regular LDPCA code in [7] for the same conditions. However, as the interleaver skews the distribution of errors to be higher at the beginning of the codeword, the rates are reduced by up to 0.038 bit/bit compared to the traditional codes in [6] and by up to 0.030 bit/bit compared to our previous regular codes in [7]. These were calculated at a conditional entropy H($X'|SI'$) of 0.27 bit/bit and 0.37 bit/bit respectively and represent a reduction in the gap to the SW bound of up to 56% and 50% compared to previous gaps. On average, the gap to the SW bound was reduced by about 50% and 45%. Fig. 5 shows the performance of the proposed LDPCA code at p_1=$4.5p_2$ (40% detection rate) and at p_1=$7p_2$ (70% detection rate), where the rates were reduced by up to 0.046 bit/bit (at H($X'|SI'$) of 0.82 bit/bit) and 0.037 bit/bit (at H($X'|SI'$) of 0.41 bit/bit) respectively, reducing the gap by up to 45% and 38%. The average gap reduction obtained is of about 35% and 20%.

The gain in performance depends on the error ratio p_1:p_2 obtained after interleaving the codewords according to the predictions made by the *prediction of error distribution* module. A similar gain in performance was also observed at other probability ratios p_1:p_2 (where $p_1 > p_2$) and even at longer codeword lengths. Furthermore, the performance of the codes never goes below that of the traditional codes in [6].

V. Conclusion

This letter considered the construction of rate-adaptive LDPC codes which exploit the prediction of dependency errors to distribute the unreliable edges evenly among all the check nodes. The performance of these codes is similar to that of the traditionally constructed codes when error prediction fails. However, their performance incrementally improves as more accurate error predictions are achieved, reducing the gap to the SW bounds by up to 56% compared to the traditional codes.

REFERENCES

[1] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inf. Theory*, vol. 19, pp. 471–480, 1973.

[2] A. Wyner, "Recent results in the Shannon theory," *IEEE Trans. Inf. Theory*, vol. 20, pp. 2–10, 1974.

[3] A. Aaron and B. Girod, "Compression with side information using Turbo codes," *DCC*, 2002.

[4] A. Liveris, Z. Xiong, and C. Georghiades, "Compression of binary sources with side information at the decoder using LDPC codes," *IEEE Commun. Lett.*, vol. 6, pp. 440–442, 2002.

[5] D. Varodayan, A. Aaron, and B. Girod, "Rate-adaptive codes for distributed source coding," *Signal Process.*, vol. 86, pp. 3123–3130, 2006.

[6] J. Ascenso, C. Brites, and F. Pereira, "Design and performance of a novel low-density parity-check code for distributed video coding," *ICIP*, 2008.

[7] J. J. Micallef, R. A. Farrugia, and C. J. Debono, "Low-density parity-check codes for asymmetric distributed source coding," *ICITIS*, 2010.

[8] Q. Xu, V. Stankovic, and Z. Xiong, "Distributed joint source-channel coding of video using Raptor codes," *DCC*, 2005.

[9] K. Yuen, B. Li, and B. Liang, "Distributed data gathering in multi-sink sensor networks with correlated sources," *IFIP Networking*, 2006.

[10] J. J. Micallef, R. A. Farrugia, and C. J. Debono, "Improved Wyner-Ziv video coding efficiency using bit plane prediction," *ICIP*, 2011.

[11] R. Puri and K. Ramchandran, "PRISM: a new robust video coding architecture based on distributed compression principles," *40th Allerton Conf. Commun., Control, and Comput.*, 2002.

[12] T. Tian, C. Jones, J. D. Villasenor, and R. D. Wesel, "Selective avoidance of cycles in irregular LDPC code construction," *IEEE Trans. Commun.*, vol. 52, pp. 1242–1247, 2004.

[13] N. Rahnavard and F. Fekri, "Unequal error protection using low-density parity-check codes," *ISIT*, 2004.