Improved Rate-Adaptive Codes for Distributed Video Coding

Jeffrey J. Micallef, Reuben A. Farrugia and Carl J. Debono
Department of Communication and Computer Engineering
University of Malta
Msida, Malta
jeffrey.micallef@ieee.org, reuben.farrugia@um.edu.mt, c.debono@ieee.org

***Abstract*—Distributed Video Coding (DVC) is a coding paradigm which shifts the major computational intensive tasks from the encoder to the decoder. Temporal correlation is exploited at the decoder by predicting the Wyner-Ziv (WZ) frames from the adjacent key frames. Compression is then achieved by transmitting just the parity information required to correct the predicted frame and recover the original frame. This paper proposes an algorithm which identifies most of the unreliable bits in the predicted bit planes, by considering the discrepancies in the previously decoded bit plane. The design of the used Low Density Parity Check (LDPC) codes is then biased to provide better protection to the unreliable bits. Simulation results show that, for the same target quality, the proposed scheme can reduce the WZ bit rates by up to 7% compared to traditional schemes.**

I. INTRODUCTION

Traditional video coding schemes suppress spatial and temporal redundancies at the encoder to achieve high compression efficiencies. This contributes to asymmetric complexities in the system where the encoder is about 5 to 10 times more complex than the decoder [1]. Such encoders are therefore inappropriate for several emerging video capturing devices where computational resources are scarce. Distributed Video Coding (DVC) relies on Slepian-Wolf (SW) [2] and Wyner-Ziv (WZ) theorems [3] to minimize the complexities at the encoder. These theorems suggest that the coding efficiency should not be affected if the source statistics are explored at the decoder to reduce the complexity of the encoder.

The temporal correlation in DVC is explored at the decoder, where a prediction of the WZ frame (called the Side Information (SI)) is generated from the adjacent key frames using Motion Compensated Temporal Interpolation (MCTI) [4]. The dependency error between the original WZ frame and the predicted SI is then modeled as a virtual channel and only the syndrome information needed to correct SI is transmitted. Therefore, compression is heavily influenced by the coding efficiency of the channel coding scheme and by the quality of the SI. Turbo codes and convolution codes were initially employed in the Stanford [5] and in the PRISM [6] architectures respectively. Later, LDPC Accumulate (LDPCA) codes [7] were found to approach better the capacity of the virtual channel in DVC applications and soon replaced turbo codes in the state-of-the-art CODECs such as DISCOVER [8].

The design of Unequal error Protected LDPC codes, where some of the variables nodes have better convergence than others, was presented in [9], [10], considering an irregular profile for the variable nodes. In [11] we exploited the predictions of the dependency error and ensured that the edges receiving unreliable information can be distributed more uniformly across all the check nodes. This improved the convergence of the LDPCA code, giving a performance which is up to 13.3% closer to the SW bounds. However, the performance of these codes is dependent on the accuracy of the error predictions. This paper presents a DVC architecture which adopts an error localization mechanism to localize the unreliable bits in the predicted SI bit planes. This mechanism allows us to exploit the optimal performance of the codes we presented in [11] for DVC applications. The LDPCA codes presented in this paper vary from those in [11] as they consider an irregular degree distribution. This is used as in [9] to further improve the protection offered to the unreliable bits. Furthermore, in contrast with [11], the unreliable edges are uniformly distributed even at the lower rate sub-codes. Simulation results show that the proposed system can reduce the WZ bit-rates by up to 7% for the same target quality.

The paper is organized as follows: Section II discusses the adopted WZ video coding architecture. Section III gives an overview of the proposed error localization mechanism, while Section IV considers the construction of the proposed LDPCA code. The RD performance of the suggested scheme is then analyzed in Section V and Section VI concludes the paper.

II. WYNER-ZIV VIDEO CODING ARCHITECTURE

Fig. 1 illustrates the used Wyner-Ziv (WZ) video coding architecture. It is based on the IST Pixel Domain WZ codec in [4], with additional blocks introduced to predict the reliability of the SI bits and to order them in an effort to obtain the best performance from the LDPCA codes. Its operation is as follows: The incoming frame sequence is divided into key frames and WZ frames. The key frames are conventionally encoded with H.264/AVC Intra coding, while the WZ frames are divided into four quadrants and quantized into 2^M levels to produce the quantized symbol stream q. The *error localization* module adopts the methods discussed in Sec. III to localize the unreliable bits in each bit plane and generate an index sequence I. This index is used by the *interleaver* to place the unreliable bits in each quadrant at the beginning of the codeword. The interleaved sequence is then fed into the LDPCA encoder and the resulting syndrome bits are stored in a buffer to be transmitted in chucks upon decoder's request.

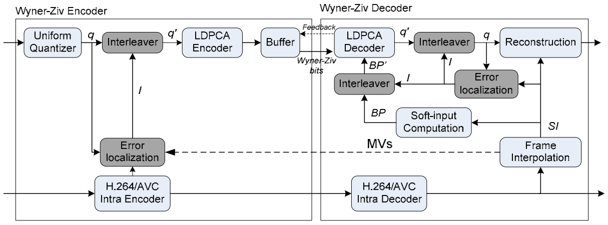
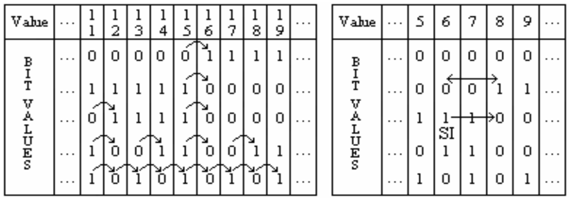


Figure 1. Wyner-Ziv video coding architecture.

At the decoder, the SI is first generated by performing MCTI [4] on the received key frames. The dependency error between the original WZ frame and the SI is then modeled as a Laplacian distribution [12] and used to convert the SI bits into bit-probabilities. Meanwhile, the *error localization* module re-generates the same index sequence I used at the encoder, and uses it to organize the order of the bit-probabilities to match that of the encoded bit stream, before they can be fed into the LDPCA decoder. Channel decoding is then performed using the received chunks of syndrome bits. If decoding fails, additional syndrome bits are requested through the feedback channel, until decoding is successful. The decoded bits are then placed back into their correct position and used to regenerate the original quantized stream q. The decoder then uses the SI together with the quantized symbols to obtain the best reconstruction of the original WZ frame.

III. ERROR LOCALIZATION MECHANISM



Value	…	11	12	13	14	15	16	17	18	19	…
BIT VALUES	…	0	0	0	0	0	1	1	1	1	…
	…	1	1	1	1	1	0	0	0	0	…
	…	0	1	1	1	1	0	0	0	0	…
	…	1	0	0	1	1	0	0	1	1	…
	…	1	0	1	0	1	0	1	0	1	…

(a)

Value	…	5	6	7	8	9	…
BIT VALUES	…	0	0	0	0	0	…
	…	0	0	0	1	1	…
	…	1	1 SI	1	0	0	…
	…	0	1	1	0	0	…
	…	1	0	1	0	1	…

(b)

Figure 2. Binary representation of integer value.

In DVC, the difference between the original WZ pixel values and the corresponding SI is expected to follow a Laplacian distribution [4]. Given that the differences is generally very small, it is expected that most of the time a set of consecutive low significant bits will be flipped up till a certain bit plane level, as shown in Fig. 2(a). The proposed error localization mechanism exploits this observation to localize the unreliable bits in the predicted bit plane. Fig. 3 considers an illustrative example of this method. In this figure, the 3rd, 4th and 5th bit planes of a WZ frame were XORed with the corresponding bit planes of the SI and hence the grey dots represent the locations where the two bit planes differ. It can be seen that the discrepancies present in the 4th bit plane (Fig. 3(b)) are a subset of the discrepancies at the lower level (3rd) bit plane shown in Fig. 3(a), and consecutively a superset of the discrepancies at the next higher level (5th) bit plane shown in Fig. 3(c). Hence, most of the unreliable bits in the 5th bit plane can be located by considering the discrepancies in the 4th bit plane and filter out the locations where bit flipping stops at the 4th bit plane without affecting the 5th one. This can be determined by searching for the value nearest to the SI for which the 4th bit differs from that of the SI and consider this value as a better predictor of the WZ value. If the 5th bit of this value differs from that of the SI, it is assumed that bit flipping affects the 5th bit plane as well and that pixel is interleaved to the beginning of the codeword. This algorithm works well even when the difference between the WZ value and the SI is greater than one and is more accurate to predict the unreliable bits in the higher order bit planes.

To predict most of the unreliable bits at the 4th bit plane the discrepancies at the 3rd bit plane (grey pixels in Fig. 3(a)) are scanned one by one. For the pixels marked by the red circle, it is observed that the SI is 6 and the corresponding WZ value is 9, hence according to Fig. 2(b) the value 8 should be considered as the predictor of the WZ value. Since the 4th bit of this value differs from that of the SI, it is expected that the 4th bit plane has an unreliable bit at these locations and from Fig. 3(b) it is clear that these locations do in fact differ at the 4th bit plane as well. When localizing the unreliable bits at the 5th bit plane, the same pixels are considered again; however, this time the 5th bit matches that of the SI (see Fig. 3(b)) and hence it is assumed that the 5th bit plane does not contain unreliable SI bits at these locations, which is a correct assumption according to Fig. 3(c). Fig. 3(d)-(f) show the discrepancies occurring when the pixels in each quadrant are interleaved with the proposed index sequence. Once again the grey pixels represent the position where the two bit planes differ, showing that most of the unreliable bits can be effectively interleaved to the beginning of the codeword (at the top of each quadrant). For the first 100 frames of the *Hall Monitor* sequence, this mechanism can detect about 91%, 94% and 98% of the discrepancies in the 3rd, 4th and 5th bit plane.

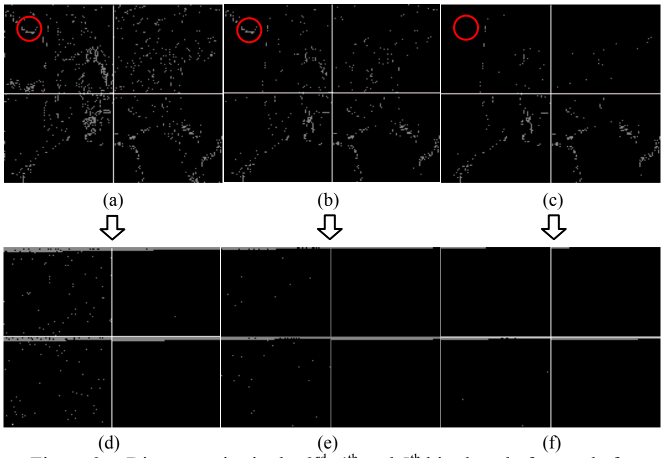


Figure 3. Discrepancies in the 3rd, 4th and 5th bit plane before and after interleaving.

While, this localization mechanism performs very well when locating the unreliable bits for the 3rd and higher significant bit planes, it fails to predict the unreliable bits at the 2nd bit plane from the discrepancies at the 1st bit plane. For this reason, it is proposed that encoding starts from the 2nd bit plane using random interleaving and then proceed with higher significant bit planes using the proposed interleaving scheme. The least significant bit plane is then encoded at the end considering the discrepancies in 2nd bit plane to locate a subset of the unreliable bits at 1st bit plane.

At the decoder, the discrepancies in the previously decoded bit plane are determined by XORing the bit plane of the SI with the bit plane obtained after successful decoding. Meanwhile, the encoder computes the XOR operator between the previously encoded bit plane and the corresponding bit plane of the SI to obtain the same discrepancies. A fast and low-complexity reproduction of the SI can be obtained by allowing the decoder to transmit the Motion Vectors (MVs) estimated during the generation of the SI as in [13]. These MVs, which can be compressed efficiently using arithmetic coding, to allow the encoder to reconstruct the SI from the adjacent key frames, enabling it to generate the same index *I* on both sides of the codec.

IV. Proposed LDPCA Code Construction

The error localization mechanism discussed in the previous section was adopted by the DVC architecture to identify and place most of the unreliable bits at the beginning of the codeword. A method similar to our previous work in [11] can therefore be employed in this paper. This method divides the variable nodes into two sets: i) the unreliable variable nodes at the beginning of the codeword and ii) the remaining reliable variable nodes. The LDPCA base code is then biased such that each check node receives at most one edge connected to an unreliable variable node, as shown in Fig. 4. The results presented in [11] have shown that the uniformity of unreliable edges significantly improves the convergence and the error correcting capability of the LDPCA codes when the performance of the error localization mechanism is satisfactory.

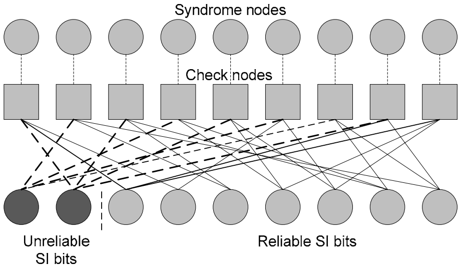


Figure 4. Considered edge biasing.

The LDPCA codes considered in this paper extend on our previous work by using an irregular degree distribution for the variable nodes. In this case, the highest degree variable nodes are assigned to the unreliable variable nodes since these are known to converge faster and to provide a higher level of protection than the lower degree variable nodes [9]. In addition, the proposed code construction also ensures that the unreliable edges are distributed uniformly among all the available check nodes even at the lower rate sub-codes. This is achieved by assuming that an ideal error localization mechanism can incrementally accumulate the unreliable SI bits at the beginning of the codeword. Following this assumption, the proposed LDPCA code construction requires that the variable nodes of the highest rate graph are considered one by one, starting from the first unreliable variable nodes. Meanwhile, the check nodes are grouped together so that each group represents the check nodes that are merged to form the lowest rate sub-graph. The edges of the leftmost variable nodes are then distributed evenly across these groups, considering only one check node within each group. When all the groups are considered once, each group is split in half so as to represent the check nodes of a higher rate graph and the next edges are distributed evenly across the newly created groups. Splitting of check node groups and distribution of edges across the new groups is repeated until all the highest rate check nodes are considered only once, so as to obtain the same base code in [11]. The rest of the graph is then built as in [7] where graph conditioning techniques [14] are considered to build the highest rate base code, while 4-lengthed cycles affecting 2-degree variable nodes and loss of edges are avoided for all lower rate codes.

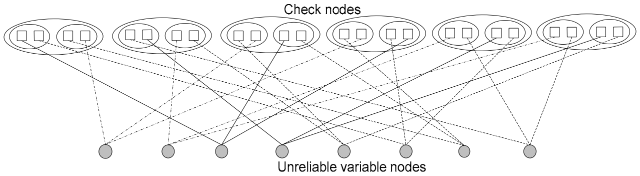


Figure 5. Proposed LDPCA code construction.

An example of such code construction is illustrated in Fig. 5. This figure considers only the edges connected to the first set of unreliable variable nodes, while the other check node edges are assumed to be randomly connected to the remaining variable nodes. The ovals at the top represent the way the check nodes are merged together to obtain the check nodes of the lower rate sub-codes. It can be observed that, when only the first two bits are in error, the check nodes of the lowest rate (other ovals) graph receive one unreliable edge each. Meanwhile, when the first four bits are in error, the outer ovals receive two unreliable edges each, which are split such that each of the inner ovals receives only one unreliable edge. This uniformity is also maintained when all the eight bits are in error and helps improving the convergence of the lower rate sub-codes, such that most of the errors can be corrected with less syndrome information.

V. Experimental Results

The performance of the proposed coding scheme is analyzed using the coding architecture described in Sec II. Fig. 6 and Fig. 7 show the performance obtained when encoding the first 100 frames of the *Foreman* and *Hall Monitor* sequences, considering only the Luminance component of the WZ (even) frames. Both sequences have a QCIF resolution and were encoded with a GOP size of 2, at a WZ frame rate of 15 frames per second. The different RD points were obtained by considering the quantization levels $2^M \in \{4, 8, 16, 32\}$ for the WZ frames and a Quantization Parameter $QP \in \{34, 30, 26, 21\}$ and $QP \in \{31, 28, 24, 20\}$ for the corresponding key frames of the *Foreman* and *Hall Monitor* sequences respectively. These QPs were chosen such that the average quality (Peak Signal-to-Noise Ratio (PSNR)) of the key frames is the same as the average quality of the WZ frames. Each plot considers the performance that is obtained with 1) the *traditional coding scheme* where the bit planes are randomly interleaved and encoded with the traditional LDPCA codes in [7] and 2) the *proposed coding scheme* where the bit planes are ordered according to the index I given by the error localization mechanism and encoded with the proposed LDPCA codes. Both LDPCA codes have a codeword length of 6336 bits and a degree distribution of $\lambda(x) = 0.3x+0.4x^2+0.3x^3$ with a rate multiple of 66.

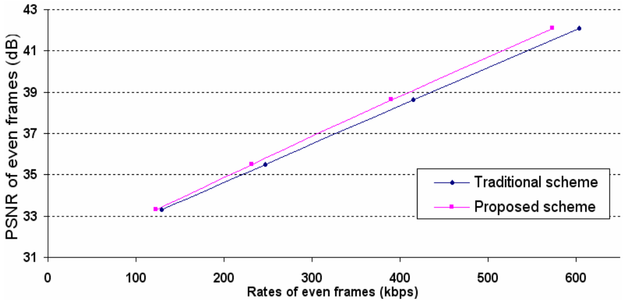


Figure 6. Rate-Distortion performance for the *Foreman* sequence.

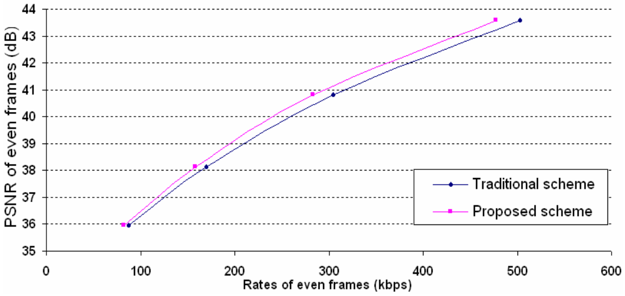


Figure 7. Rate-Distortion performance for the *Hall Monitor* sequence.

The two coding schemes resulted in the same PSNR values since the same algorithms were considered for the generation of the SI and for the reconstruction process. However, with the introduction of the error localization mechanism, most of the unreliable bits were effectively placed at the beginning of the codeword leading to an improvement in the coding efficiency, when using the proposed codes. It was observed that the rates required to decode every WZ frame, were always smaller than those required by the traditional scheme and there were no cases where the LDPCA decoder failed to converge correctly. The WZ bit-rates were reduced, shifting the RD curves to the left for the same target quality. Table I summarizes the percentage reduction in WZ bit rates, compared to the traditional coding scheme, showing that the WZ bit-rates can be reduced by up to 7% for the same video quality.

TABLE I. PERCENTAGE REDUCTION IN WYNER-ZIV RATES.

No of levels	4	8	16	32
Foreman	4.94%	6.28%	6.04%	5.01%
Hall Monitor	5.48%	6.70%	7.01%	5.13%

VI. CONCLUSION

This paper introduced an error localization mechanism that can be adopted in DVC architectures to locate and place most of the unreliable bits at the beginning of the codeword. The construction of the LDPCA codes exploited this knowledge to distribute the unreliable edges as uniformly as possible among all the available check nodes. Simulation results showed that the proposed coding scheme can significantly improve the coding efficiency of the WZ video codec, reducing the WZ bit rates by up to 7% for the same target quality. Future work will consider the design of error localization mechanisms which do not require the transmission of the MVs to maintain synchronization, hence avoiding the use of the feedback channel.

ACKNOWLEDGMENT

The research work is partially funded by the STEPS-Malta. This scholarship is partly financed by the European Union - European Social Fund (ESF 1.25).

REFERENCES

[1] T. Wiegand, G. Sullivan, G. Bjøntegaard and A. Luthra, “Overview of the H.264/AVC video coding standard,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, No. 7, pp. 560-576, July 2003.

[2] J. Slepian and J. Wolf, “Noiseless coding of correlated information sources,” *IEEE Trans. Inform. Theory*, vol. 19, No. 4, pp. 471-480, July 1973.

[3] A. Wyner and J. Ziv, “The rate-distortion function for source coding with side information at the decoder,” *IEEE Trans. Inform. Theory*, vol. 22, No. 1, pp. 1-10, January 1976.

[4] J. Ascenso, C. Brites and F. Pereira, “Improving frame interpolation with spatial motion smoothing for pixel domain distributed video coding,” in *Proc. Eur. Assoc. Signal Image Process. Conf. Speech Image Process. Multimedia Commun. Services*, pp. 1–6, Jun/Jul. 2005.

[5] A. Aaron, R. Zhang and B. Girod, “Wyner-ziv coding of motion video,” in *Proc. Asilomar Conf. Signals, Syst. and Comput.*, pp. 240-244, 2002.

[6] R. Puri and K. Ramchandran, “PRISM: A new robust video coding architecture based on distributed compression principles,” in *Proc. Allerton Conf. Commun. Control and Comput.*, Oct. 2002.

[7] J. Ascenso, C. Brites, and F. Pereira, “Design and performance of a novel low-density parity-check code for distributed video voding,” in *Proc. Int. Conf. Image Process.*, pp. 1116-1119, Oct. 2008.

[8] X. Artigas, J. Ascenso, M. Dalai, S. Klomp, D. Kubasov, and M. Ouaret, “The DISCOVER codec: architecture, techniques and evaluation,” in *Proc. Int. Picture Coding Sym.*, Nov. 2007.

[9] M. Luby, M. Mitzenmacher, A. Shokrollahi, and D. Spielman, “Improved low-density parity-check codes using irregular graphs,” *IEEE Trans. Inform. Theory*, vol. 47, pp. 585–598, Feb. 2001.

[10] N. Rahnavard and F. Fekri, “Unequal error protection using low-density parity-check codes,” in *Proc. Int. Sym. Inform. Theory*, pp. 449, Jun/Jul. 2004.

[11] J. J. Micallef, R. A. Farrugia and C. J. Debono, “Low-density parity-check codes for asymmetric distributed source coding,” in *Proc. Int. Conf. Info. Theory Info. Security,* pp. 985-988, Dec. 2010.

[12] C. Brites, J. Ascenso and F. Pereira, “Studying temporal correlation noise modeling for pixel based wyner-ziv video coding,” in *Proc Int. Conf. Image Process.,* pp. 273-276, Oct. 2006.

[13] K. Min, S. Park and D.Sim, “Distributed video coding based on adaptive slice size using received motion vectors,” in *Proc. Int. Picture Coding Sym.*, pp. 262-265, Dec. 2010.

[14] T. Tian, C. Jones, J.D. Villasenor, and R.D. Wesel, “Selective avoidance of cycles in irregular LDPC code construction,” IEEE *Trans. on Commun.*, Vol. 52, No. 8, pp. 1242-1247, 2004.