Modified Distribution of Correlation Noise for Improved Wyner-Ziv Video Coding Performance

Jeffrey J. Micallef[1], Reuben A. Farrugia[2], and Carl J. Debono[3]

Department of Communications and Computer Engineering,
University of Malta, Msida, MSD 2080, Malta.

[1] jeffrey.micallef@ieee.org
[2] reuben.farrugia@um.edu.mt
[3] c.debono@ieee.org

***Abstract*—Despite theorems predicting that Distributed Video Coding can achieve the same performance as traditional predictive video coding schemes, the coding efficiency of practical architectures is still far from these bounds. This is attributed to the poor Side Information (SI) estimated at the decoder and to the inability of the channel codes to recover the source at the Slepian-Wolf (SW) limits. This paper tackles the latter issue by recovering the SI bit-planes starting from the most unreliable bit of each coefficient. Most of the mismatch in SI is thus accumulated within the first decoded bit-planes, leaving the last bit-planes with very few or no mismatch. Low-Density Parity-Check Accumulate (LDPCA) codes can then benefit from such compact distribution of correlation noise since they offer a smaller percentage error, from the SW bounds, when mismatch is accumulated in few higher entropy bit-planes. Furthermore, with this setup, most of the last bit-planes can be recovered very effectively using just 8-bit or 16-bit Cyclic Redundancy Codes. Experimental results show that the proposed scheme can reduce the Wyner-Ziv bit-rates by up to 21% compared to the DISCOVER codec.**

I. Introduction

Mobile devices or miniature endoscopic cameras have very limited power and processing capabilities and this imposes a serious challenge for the implementation of traditional video coding schemes which employ very computationally demanding encoders [1]. Slepian-Wolf (SW) [2] and Wyner-Ziv (WZ) theorems [3] have shown that it is possible to shift the computational intensive tasks needed to explore the source statistics, from the encoder to the decoder, without affecting compression efficiency. Distributed Video Coding (DVC) exploits such theorems to create a new coding paradigm that can facilitate the implementation of lightweight encoders, targeted for such applications. Surveillance systems and multi-view video coding where multiple cameras are connected to a central decoding system, may also benefit from such coding paradigm since it also provides cheaper encoders.

Opposed to the traditional video coding schemes, where the source statistics are explored at the encoder, the DVC paradigm exploits temporal correlation at the decoder and predicts the WZ frames using Motion Compensated Temporal Interpolation (MCTI) between the adjacent key frames. These predictions, known as the Side Information (SI), are then used to aid compression by allowing the decoder to recover the WZ bit-planes using only a sub-set of parity bits needed to correct the predictions in SI. The efficiency of a DVC system is thus dependent on the quality of the SI and the ability of channel codes to correct SI at rates close to the SW bounds.

Early DVC architectures considered syndrome decoding techniques for channel coding [4]. Yet, these soon evolved to include more sophisticated codes, like Turbo-codes [5] or Low-Density Parity-Check (LDPC) codes [6], which use the statistics of the underlying channel to improve performance [7]. Using dynamic noise modelling [8], the previously decoded bit-planes [9]-[10], or reversing the order of the bit-planes [11] can all improve the accuracy of channel statistics and help to approach the SW rates with either code. Yet, the LDPC and LDPC Accumulate (LDPCA) codes [12] were found to perform better than Turbo-Codes and soon replaced them in state-of-the-art DVC codecs [13]-[14]. The authors in [15] improved the performance of rate-adaptive LDPC codes by optimizing the degree distribution over a certain entropy range, whilst [16] used a greedy girth maximization algorithm to maintain the largest cycles as the check nodes are merged. Conversely, the authors in [17] skewed most of the correlation noise to the beginning of the codeword and used Unequal error Protected LDPCA codes [18] to better protect the variable nodes receiving unreliable SI information and approach the SW bounds.

The work in this paper suggests a model which can anticipate the reliability of the soft-input values and uses it to order the bits of each SI coefficient according to their probability of mismatch, starting from the most unreliable bits. This allows most of the correlation noise to be accumulated within the first bit-planes, so that the dependency error can be corrected more efficiently when using LDPCA codes. These codes offer a lower percentage error from the SW bound when mismatch is accumulated in few higher entropy bit-planes, rather than spread across several bit-planes as in traditional DVC schemes [12]. Moreover, most of the last bit-planes are left with very few (generally 1-bit or 2-bit errors) or no mismatch at all and these can be recovered at rates very close to the SW bounds using just 8-bit or 16-bit Cyclic Redundancy Codes (CRC). Simulation results show that the proposed method can reduce the WZ bit-rates by up to 21%

MMSP'13, Sept. 30-Oct. 2, 2013, Pula (Sardinia), Italy.
978-1-4799-0125-8/13/$31.00 ©2013 IEEE.

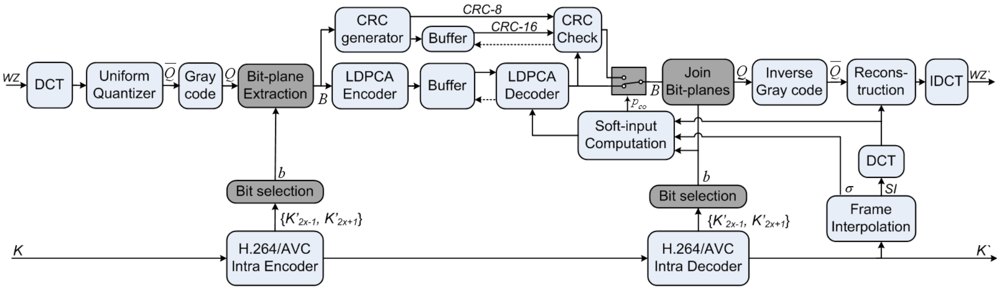


Fig. 1. Proposed Wyner-Ziv video coding architecture

compared to the DISCOVER codec, with minimal increase in the encoder's complexity.

The WZ video coding architecture adopted in this work is outlined in Section 2. Section 3 briefly reviews how to anticipate the reliability of the soft-input values, whilst Section 4 shows how to determine the order of coefficient bits. CRC codes which are suitable to correct bit-planes with a very small number of bit-errors are then discussed in Section 5. Section 6 presents the experimental results, whereas the final comments and conclusions are provided in Section 7.

II. Wyner-Ziv Video Coding Framework

Fig. 1 shows the transform domain Wyner-Ziv video coding architecture considered in this work, which is built around the DISCOVER codec [14] with the introduced modules depicted in a darker shade of grey. For a GOP size of 2, the even frames are encoded using WZ techniques and denoted as $\mathbf{WZ}_{2x}$. Conversely, the odd frames are encoded as key frames using the H.264/AVC Intra coding scheme and denoted as $\mathbf{K}_{2x-1}$ and $\mathbf{K}_{2x+1}$ for the backwards and forwards key frames of the WZ frame $\mathbf{WZ}_{2x}$. The WZ frames will first undergo a block-based Discrete Cosine Transform (DCT) and the DCT coefficients are organized into bands having coefficients of the same frequency. Each band is then uniformly quantized into 2^M levels, using a dead-zone quantizer which can adapt with the dynamic range of the band [19], and the quantized symbols $\bar{\mathbf{Q}}$ are mapped into gray code representation $\mathbf{Q}$. The *Bit selection* module considers the DCT coefficients of the reconstructed key frames $\mathbf{K}'_{2x-1}$ and $\mathbf{K}'_{2x+1}$, as described in Section 4, and for each quantized symbol $\mathbf{Q}$ it determines the next bit-level b ($\mathbf{Q}^{(b)}$) for which the decoder may produce the most unreliable soft-input predictions. The $\mathbf{Q}^{(b)}$ bits for all the coefficients within the band are then gathered together within the *Bit-plane Extraction* module to form the next bit-plane $\mathbf{B}$ to be encoded. This bit-plane is fed into an *LDPCA encoder* [12] and a *CRC generator* module. The latter calculates both an 8-bit and a 16-bit CRC. The 8-bit CRC is sent directly to the decoder, while the 16-bit CRC and the LDPCA syndrome bits are stored in a buffer and sent upon request.

The decoder reconstructs the adjacent key frames $\mathbf{K'}_{2x-1}$ and $\mathbf{K'}_{2x+1}$, using H.264/AVC Intra decoding technique and utilizes them to predict the SI via MCTI techniques [8]. These frames are also used to determine which bit-levels b from the quantized symbol stream $\mathbf{Q}$ are being encoded within the next bit-plane $\mathbf{B}$, allowing the decoder to maintain synchronization. The correlation noise between the original WZ frame and the SI is subsequently modeled at coefficient level, using the difference between the forward and backward motion compensated frames as in [20]. The resulting Laplacian parameters σ and all the previously decoded bit-planes are then used to generate the appropriate soft-input values for the next bit-plane $\mathbf{B}$ [9] and fed into the LDPCA decoder and into the CRC Check module. The decoder calculates the cross-over probability p_{co}, for the current bit-plane and uses it to determine the appropriate decoding strategy, as discussed in Section 5. This varies from using the traditional LDPCA decoding scheme for a high probability of mismatch to using a 16-bit CRC code which is more efficient when recovering bit-planes with few or no bit-errors. The recovered bit-planes are then joined together to form the quantized symbol stream $\mathbf{Q}$ which undergoes inverse gray code mapping. The resulting symbols $\bar{\mathbf{Q}}$ are used together with the SI, to obtain the best reconstruction of the original WZ coefficients [21]. An Inverse DCT is finally applied to get the WZ frame back into the pixel domain.

III. Reliability of the Soft-Input Values

Let $P(\mathbf{Q}^{(b)}=0|\mathbf{SI}, \mathbf{Q}^{(b\text{-}1)},\ldots,\mathbf{Q}^{(0)})$ and $P(\mathbf{Q}^{(b)}=1|\mathbf{SI}, \mathbf{Q}^{(b\text{-}1)},\ldots,\mathbf{Q}^{(0)})$ denote the conditional bit-probabilities for the b^{th} bit of symbol $\mathbf{Q}$, given the side information SI and the previously decoded bits $\mathbf{Q}^{(b\text{-}1)},\ldots, \mathbf{Q}^{(0)}$. Furthermore, let the coefficient at position $\mathbf{p} = (p_x, p_y)$ in the SI frame be denoted by $\mathbf{SI}_{2\text{x}}(\mathbf{p})$, whilst that of the WZ frame denoted by $\mathbf{WZ}_{2x}(\mathbf{p})$. The conditional bit-probabilities are calculated by modelling a Laplacian distribution around the coefficient value of SI ($\mathbf{SI}_{2x}(\mathbf{p})$) and then add up the relevant areas under the distribution, as indicated by the previously decoded bit-planes [9]. Fig. 2 illustrates the procedure used to estimate the soft-input values for the 3rd bit-plane (b = 2) when the SI has a value of 123 and the first two bit-planes are both decoded as 1. The bit-probabilities, shown by the shaded regions, are then used to calculate the Log-Likelihood Ratios (LLR) values according to:

$$LLR = \log\left(\frac{P\left(Q^{(b)} = 0 \mid SI, Q^{(b-1)}, \ldots, Q^{(0)}\right)}{P\left(Q^{(b)} = 1 \mid SI, Q^{(b-1)}, \ldots, Q^{(0)}\right)}\right) \qquad (1)$$

The sign of the LLR values, denoted as $\mathbf{S}^{(b)}$, represents the predicted bit-value and the magnitude represents the confidence in such decision.

Gray-code representation has been considered to improve the correlation between the WZ data and its SI [22]. The grey and the white regions in Fig. 2 represent the intervals for which the coefficient values have a corresponding **Q** index with the same bit-value at a given bit-plane level *b*. In general, it is noticed that the LLR predictions, calculated at the decoder, are incorrect when the SI coefficients $\mathbf{SI}_{2x}(\mathbf{p})$ and the WZ coefficients $\mathbf{WZ}_{2x}(\mathbf{p})$ are within adjacent intervals. For example, if in Fig. 2 the WZ coefficient $\mathbf{WZ}_{2x}(\mathbf{p})$ has a value of 118, $P(\mathbf{Q}^{(b)}=1|\mathbf{SI},\mathbf{Q}^{(b-1)},\ldots,\mathbf{Q}^{(0)})$ for the 3rd bit-plane (b=2) is greater than $P(\mathbf{Q}^{(b)}=0|\mathbf{SI},\mathbf{Q}^{(b-1)},\ldots,\mathbf{Q}^{(0)})$, and the LLR value is incorrectly predicted as having a bit-value of 1 despite the low correlation noise. This is always observed when $\mathbf{SI}_{2x}(\mathbf{p})$ and $\mathbf{WZ}_{2x}(\mathbf{p})$ are within adjacent intervals and this observation is used to anticipate the reliability of the soft-input values.

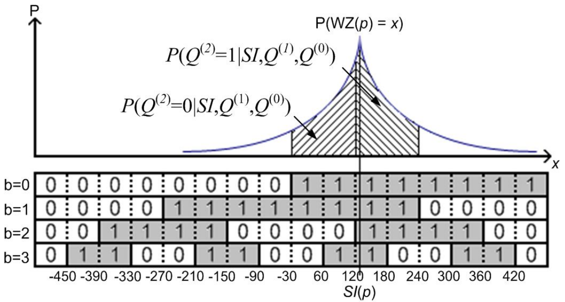

P
P(WZ(p) = x)
$P(Q^{(2)}=1|SI,Q^{(1)},Q^{(0)})$
$P(Q^{(2)}=0|SI,Q^{(1)},Q^{(0)})$
x
b=0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1
b=1 0 0 0 0 1 1 1 1 1 1 1 1 0 0 0 0
b=2 0 0 1 1 1 1 0 0 0 0 1 1 1 1 0 0
b=3 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1 0
-450 -390 -330 -270 -210 -150 -90 -30 60 120 180 240 300 360 420
SI(p)

Fig. 2. Calculating soft-input values for the 3rd bit-plane

For the DC band, the coefficients are quantized using a uniform quantizer assuming the range $[0, 2^{11})$, hence the bth bit-plane of the quantized symbols **Q** maintains the same bit-value within intervals $I_i^{(b)}$, whose endpoints (denoted by $B_i^{(b)}$) are set by:

$$B_i^{(b)} = 2^{11-b}(i-\tfrac{1}{2}) \text{ for } i = 0 \ldots 2^b \tag{2}$$

Thus an interval *i*, within the set of intervals $I_i^{(b)}$ for bit-plane *b*, has the range $[B_i^{(b)}, B_{i+1}^{(b)})$ with the intervals falling outside the coefficient range $[0, 2^{11})$ truncated to such range.

Conversely, for the AC coefficient bands, the quantizer adapts with the dynamic range of the coefficients [19] and thus the size of the intervals varies from band to band. For an AC band with a coefficient range [-*MaxCoeff*, *MaxCoeff*), the Quantizer Step-size *QS* is set to:

$$QS = \left\lceil \frac{2 \times MaxCoeff}{2^L} \right\rceil \tag{3}$$

where 2^L represents the number of quantization levels for the band. The quantized coefficient symbols **Q**, are first incremented by 2^{L-1} to obtain indices in the range $[0, 2^L)$, before mapping them to Gray-codes. As seen in Fig. 2, the bth bit-plane will thus change its bit-value at intervals of width:

$$W_b = 2^{L-b} \times QS \tag{4}$$

except for the first and last interval which are half W_b long, and for the middle interval, which is *QS* longer due to the use of double zero interval. As seen in [13], the smallest coefficient value represented with this quantizer is:

$$MinVal = -(2^{L-1}+1) \times QS \tag{5}$$

For the AC coefficients the endpoints $B_i^{(b)}$, for the set of intervals $I_i^{(b)}$ at the bth bit-plane, are given by:

$$B_i^{(b)} = \begin{cases} MinVal + W_b(i-\frac{1}{2}) & ,i \le \left\lceil \frac{2^b}{2} \right\rceil \\ MinVal + W_b(i-\frac{1}{2}) + QS & ,i > \left\lceil \frac{2^b}{2} \right\rceil \end{cases} \tag{6}$$

The reliability of the soft-input values predicted at the decoder can be anticipated by considering that the LLR values derived for the WZ coefficient $\mathbf{WZ}_{2x}(\mathbf{p})$, or $\mathbf{SI}_{2x}(\mathbf{p})$, found near the endpoints $B_i^{(b)}$, are more likely to be incorrect. This is because a very small variation in light intensity can cause the corresponding SI or WZ coefficient to fall within an adjacent $I_i^{(b)}$ interval. Conversely, the coefficients which are further away from such endpoints (e.g. at the middle of the interval) have a higher probability of being predicted correctly. If $\mathbf{WZ}_{2x}(\mathbf{p})$ coefficient in Fig. 2 has a value of 240, the LLR predictions for the 3rd bit-plane are correct for all values of $\mathbf{SI}_{2x}(\mathbf{p})$ between 120 and 360, since these would still fall in the same $I^{(b)}$ interval as the WZ coefficient. Moreover, it is very unlikely for the SI to fall outside this range given the dependency error follows a Laplacian distribution.

Fig. 3 illustrates the probability of mismatch between the bth bit of the quantized *WZ* symbols **Q** i.e. $\mathbf{Q}^{(b)}$, and corresponding bits predicted by the sign of the LLR values $\mathbf{S}^{(b)}$, when considering the first AC coefficient of all the frames in the *Foreman* sequence. In general, these coefficients have a dynamic range $\mathbf{WZ}_{2x}(\mathbf{p}) \in [-486, 486)$ and are quantized at 2^L=32-levels, yet the statistics were collected over the region $WZ \in [-210, 180]$ which provided enough sample points to observe a clear statistical correlation. It can be seen that the probability of mismatch ($P(\mathbf{S}^{(b)} \neq \mathbf{Q}^{(b)})$) for bit-plane *b* is small at the centre of the $I^{(b)}$ intervals and peaks at the endpoints $B_i^{(b)}$.

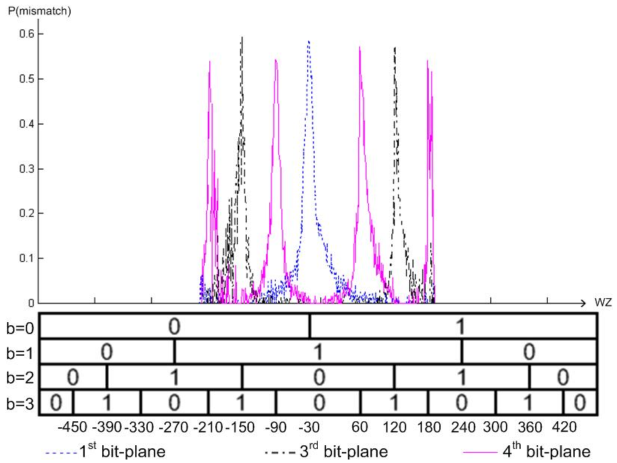


Fig. 3. Probability of mismatch for different WZ coefficient values

IV. Accumulating Unreliable SI Bits

The coefficients values of the adjacent key frames $\mathbf{K'}_{2x-1}(\mathbf{p})$ and $\mathbf{K'}_{2x+1}(\mathbf{p})$, are highly correlated to that of the WZ frame $\mathbf{WZ}_{2x}(\mathbf{p})$ and to the SI, especially for regions of low motion. Hence, given that neither the WZ frame, nor the SI frame, are available at the both sides of the codec to ensure synchronization, the average DCT coefficient of the adjacent frames:

$$K_{avg}(p)=\frac{K'_{2x-1}(p)+K'_{2x+1}(p)}{2} \tag{7}$$

can be used to predict the position of $\mathbf{WZ}_{2x}(\mathbf{p})$ and $\mathbf{SI}_{2x}(\mathbf{p})$ coefficients relative to the $B_i^{(b)}$ boundaries of the b^{th} bit-plane [23]. This consequently serves as a metric to anticipate the probability of mismatch $P(\mathbf{S}^{(b)}\neq\mathbf{Q}^{(b)})$ for each coefficient bit and to control the distribution of mismatch amongst bit-planes. Furthermore, since both the adjacent key frames are readily available at both sides of the codec, synchronization is easily maintained.

The work in this paper proposes to encode each coefficient starting from the bit which is expected to have the most unreliable soft-input value. Following the observations discussed in Section 3, this can be achieved by encoding the bit-levels at which $\mathbf{K}_{\mathbf{avg}}(\mathbf{p})$ is closer to the $B_i^{(b)}$ boundaries first. For $\mathbf{K}_{\mathbf{avg}}(\mathbf{p})$ = -20, as in Fig. 4, the bit-planes are thus encoded in the order $b\in\{0,4,3,2,1\}$.

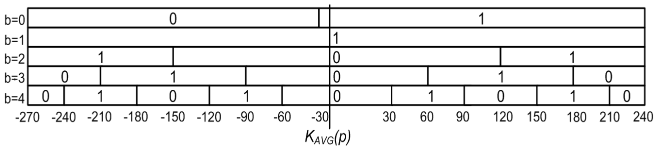
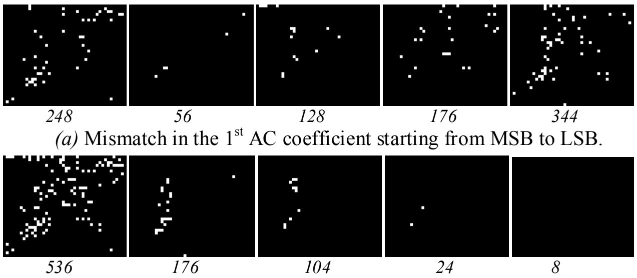


Fig. 4. Determining the encoding order of the bits



(a) Mismatch in the 1st AC coefficient starting from MSB to LSB.

(b) Accumulation of mismatch within the first bit-planes.

Fig. 5. Mismatch in *SI* (a) before and (b) after sorting the coefficient bits

Fig. 5 illustrates the discrepancies $\mathbf{S}^{(b)}\neq\mathbf{Q}^{(b)}$ within all the bit-planes of the first AC coefficient of the 15th frame in the *Hall Monitor* sequence (*15 Hz*), with the number of WZ bits required to correct such mismatch listed underneath. Fig. 5(a) considers the mismatch when the bit-planes of SI are decoded from the Most Significant Bit-planes (MSB) to the Least Significant Bit-plane (LSB), whilst Fig. 5(b) shows the mismatch when the bits of each coefficient are encoded in the proposed order. It is clear that the proposed scheme can accumulate the mismatch in the SI within the first bit-planes, leaving very few mismatches within the other bit-planes. The reason for such an arrangement is twofold; First of all, the LDPCA codes offer a smaller percentage error from the SW bounds at higher entropy [12] and hence they operate more efficiently when the mismatch in SI is accumulated in few higher entropy bit-planes, like the first three bit-planes in Fig. 5(b), rather than dispersed across a set of lower entropy bit-planes as in Fig. 5(a). For very high motion or low quantized videos, as the first bit-planes accumulate most of the unreliable bits, the SI and WZ bit-planes become almost independent but the LDPCA can still operate very close to the bounds since they are limited by a maximum rate of 1. Furthermore, with this arrangement, most of the last decoded bit-planes are left with very few (usually 1-bit and 2-bit errors) or no mismatch at all and can be decoded very efficiently using only 8 or 16-bit CRC codes, as seen in the next section.

For different values of $\mathbf{K}_{\mathbf{avg}}(\mathbf{p})$ in the range [*-MaxCoeff, MaxCoeff*), the order of the bits changes at the middle of every quantization interval *QS,* defined in (3). The encoder and decoder can thus employ a list of pre-calculated permutations, one for every half of the quantization interval, and use the index:

$$A(p)=2^{L}+floor\left(\frac{K_{avg}(p)}{2\cdot QS}\right) \tag{8}$$

to select the correct bit-order to be used from the lookup table, according to the corresponding $\mathbf{K}_{\mathbf{avg}}(\mathbf{p})$. Hence, the proposed algorithm requires only the use of some additional lookup tables, plus an integer division and addition for each coefficient, allowing the bit-planes to be ordered with minimal (O(*n*)) additional computational complexities.

V. Wyner-Ziv Coding at Low Entropy

Before trying to decode any bit-plane, the decoder requests an 8-bit CRC code from the encoder to ensure that all bit-planes are reconstructed correctly. It then considers the Laplacian model, used for the computation of the soft-input values, to estimate the average cross-over probability p_{co} for the bit-plane [24] and uses it together with CRC-8 to decide the best decoding strategy, as follows:

Case 1: When $p_{co} > T_1$, the decoder assumes a lot of mismatch in the initial predictions given by the SI and corrects them by requesting LDPCA syndrome information from the encoder, as in any traditional DVC architecture.

Case 2: When $T_1 > p_{co} > T_2$, the decoder assumes that the SI bit-plane experiences 2-bit, 1-bit, or no-bit errors at all. A 16-bit CRC code is thus used to try and correct up to 2-bit errors [25]-[28]. These codes perform closer to the bounds, compared to the LDPCA codes which need a minimum rate of 2/66 (i.e. 48-bits) to avoid loss of edges at lower rates [12]. A 16-bit CRC with a generator polynomial:

$$G(x)=x^{16}+x^{15}+x^{13}+x^{10}+x^{9}+x^{8}+x^{7}+x^{6}+x^{4}+x+1 \tag{9}$$

proposed in [29], was adopted to ensure the largest hamming distance between the set of codewords sharing the same syndrome information. The decoder considers a look-up table containing the CRC syndromes for every possible 1-bit or 2-bit error pattern. This is used to identify the possible mismatch in the SI by comparing the syndrome of the initial SI bit-plane with the syndromes stored in the look-up table. The polynomial in (9) generates a unique syndrome for every possible 1-bit error pattern, which can be uniquely identified and corrected. Yet, it shares the same syndrome information for a number of 2-bit error patterns. The soft-input values and the CRC-8 are then considered to single out the actual error pattern. The decoder selects the error pattern which can generate a valid bit-plane, according to CRC-8 code, and has a very low confidence value for at least one of the 2-bit positions. The confidence value should be smaller than half the average LLR for that bit-plane. If no error pattern in the table satisfies both the above criteria, the decoder assumes that the initial assumptions of 1-bit or 2-bit errors were not correct and requests LDPCA syndrome information to recover the WZ bit-planes.

Case 3: When $p_{co} < T_2$, the decoder assumes that the WZ bit-planes can be successfully predicted by the SI bit-plane. If CRC-8 syndrome is satisfied, the bit-plane of SI is thus used for re-construction without requesting further parity information. Conversely, if CRC is not satisfied the decoder requests CRC-16 and later LDPCA syndrome bits from the encoder until the bit-plane is recovered successfully.

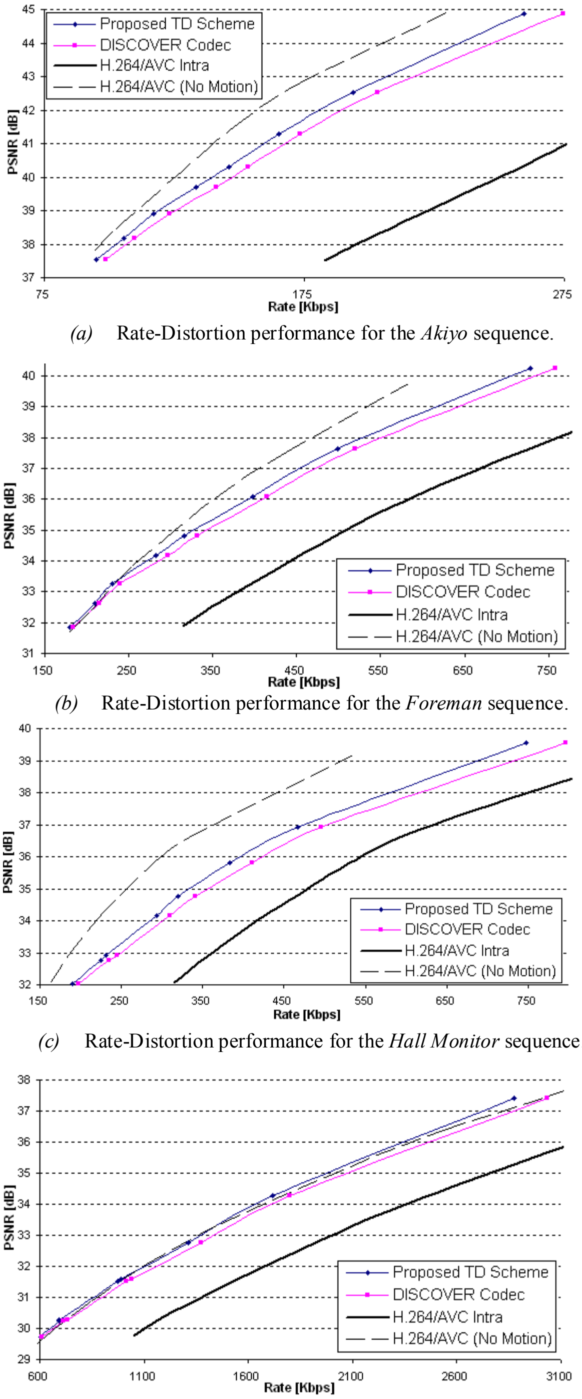
The thresholds T_1 and T_2 were set by considering that the bit-plane must have an average bit-error probability which is less than 3-bit errors for T_1 and less than 1-bit error for T_2. The thresholds were thus assigned the value:

$$T_1 = \frac{2.5}{1584} \text{ and } T_2 = \frac{0.5}{1584} \qquad (10)$$

Using this decoding strategy, the total WZ bit-rate required to correct the bit-planes in Fig. 5(b) is 11% less than that required for the bit-planes in Fig. 5(a). Nevertheless, Fig. 5 considers one of the highest motion frames in *Hall Monitor* sequence, for which a poor SI is generated. Better predictions are usually obtained for other frames, especially low-motion or heavily quantized frames, where all mismatch is confined to the first one or two bit-planes, obtaining better results.

VI. Experimental Results

The *Akiyo* (15 Hz) and *Foreman* (30 Hz) having QCIF resolution, and the *Hall Monitor* (15 Hz) and *Coastguard* (30Hz) sequences with CIF resolution, were compressed using the Wyner-Ziv video coding architecture illustrated in Fig. 1. Here, the bits of each coefficient are ordered according to the reliability of the SI predictions, as discussed in Section 4, and decoded using the scheme in Section 5. The average Rate-Distortion (R-D) performance, obtained for the luminance component of the WZ frames and key-frames, was then plotted in Fig. 6, together with R-D performance of the DISCOVER codec [13], where the coefficient bits are recovered using LDPCA codes, starting from the MSB to the LSB. For fair comparisons and to show that gains are not due to gray-code representations, the DISCOVER codec was set to consider gray code representation and both setups consider a GOP size of 2 as is common in DVC literature [20]-[24]. The DCT coefficients of the WZ frames were quantized using the eight 4×4 quantization matrices considered for the DISCOVER project [13]. Conversely, the key frames were compressed using H.264/AVC Intra coding scheme, with the quantization parameter set to maintain the same average quality through the whole video sequence. For comparison purposes, the plots in Fig. 6 also include the Rate-Distortion performance obtained when all the frames of the sequences are encoded with H.264/AVC Intra or H.264/AVC No-motion coding schemes, since only these techniques have a comparable low encoding complexity.



(a) Rate-Distortion performance for the *Akiyo* sequence.

(b) Rate-Distortion performance for the *Foreman* sequence.

(c) Rate-Distortion performance for the *Hall Monitor* sequence.

(d) Rate-Distortion performance for the *Coastguard* sequence

Fig. 6. Rate-Distortion performance for the tested sequences

The results demonstrate that the WZ bit-planes can be recovered more efficiently when the coefficient bits are ordered according to the reliability of the soft-input values, and corrected using the strategy in Section 5, rather than grouped according to their significance as done in traditional codecs. Table I summarizes the overall percentage reduction in WZ bit-rates obtained at each R-D point corresponding to the quantization matrices Q_x in [14]. It shows that the WZ bit-rates can be reduced by up to 21% compared to the DISCOVER codec, whilst the quality of the reconstructed video is left unaltered since both schemes consider the same WZ indices **Q** and the same SI generation. Better gains are observed for low-motion sequences (*Akiyo*), where the predictions of the adjacent key frames are more precise. These allow the mismatch in the SI to be corrected by the first bit-plane, whilst the other bit-planes are recovered using just CRC codes. Nevertheless, this scheme is still valid for high motion sequences (*Foreman*) where the bit-rates are reduced by up to 10 %.

TABLE I
THE OVERALL % REDUCTION IN WZ BIT-RATES

	Hall monitor	*Foreman*	*Coastguard*	*Akiyo*
Q_1	12.8 %	9.2 %	14.8 %	19.4 %
Q_2	11.0 %	8.1 %	12.8 %	16.5 %
Q_3	12.6 %	10.5 %	18.0 %	21.6 %
Q_4	11.1 %	10.2 %	11.3 %	19.4 %
Q_5	13.9 %	10.1 %	12.2 %	21.3 %
Q_6	12.2 %	8.0 %	8.7 %	20.8 %
Q_7	11.0 %	7.4 %	8.1 %	16.4 %
Q_8	9.2 %	7.3 %	7.2 %	13.1 %

VII. CONCLUSION

This paper has considered an algorithm which can anticipate the reliability of the soft-input values predicted at the decoder. This information is used to ensure that the SI coefficients are encoded starting from the most unreliable bits. With the new arrangements, the mismatch in SI is accumulated within the same bit-planes and is corrected more efficiently with LDPCA codes. Moreover, the last bit-planes, which are frequently affected by few or no-bit errors, are recovered efficiently using just 8-bit or 16-bit CRC codes. Experimental results demonstrate that the proposed approach can reduce the WZ bit-rates by up to 21% compared to the DISCOVER codec, with a minimal increase in complexity.

ACKNOWLEDGMENT

This research work was partially funded by the Strategic Educational Pathways Scholarship Scheme (STEPS-Malta) and by 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. on CSVT*, vol. 13, no. 7, pp. 560-576, Jul. 2003.

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

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

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

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

[6] A. D. Liveris, Z. Xiong, and C. N. Georghiades, “Compression of binary sources with side information at the decoder using LDPC codes,” *IEEE Commun. Lett.*, vol. 6, 2002, pp. 440-442, Oct. 2002.

[7] R. P. Westerlaken, S. Borchert, R. K. Gunnewiek, and R. L. Lagendijk, “Analyzing symbol and bit-plane-based LDPC in distributed video coding,” in *Proc. of ICIP,* Sept. 2007.

[8] J. Ascenso, C. Brites, and F. Pereira, “Improving frame interpolation with spatial motion smoothing for pixel domain distributed video coding”, in *Proc. EURASIP Conf. on Speech and Image Processing, Multimedia Commun. and Services*, Jul. 2005.

[9] S. Cheng, and Z. Xiong, “Successive refinement for the Wyner-Ziv problem and layered code design,” in *Proc. DCC*, Mar. 2004.

[10] J. J. Micallef, R. A. Farrugia, and C. J. Debono, “Improved Wyner-Ziv video coding efficiency using bit plane prediction,” in *Proc. of ICIP*, Sept. 2011.

[11] Y. Vatis, S. Klomp, and J. Ostermann, “Inverse bit plane decoding order for turbo code based distributed video coding,” in *Proc. of ICIP,* Sept. 2007.

[12] D. Varodayan, A. Aaron, and B. Girod, “Rate-adaptive distributed source coding using low-density parity-check codes,” in *Proc. Asilomar Conf. on Signals, Syst. and Comput.*, Oct. 2005.

[13] X. Artigas, J. Ascenso, M. Dalai, S. Klomp, D. Kubasov, and M. Ouaret, “The DISCOVER codec: architecture, techniques and evaluation,” in *Proc. of PCS*, Nov. 2007.

[14] DISCOVER Project [Online]. Available: www.discoverdvc.org.

[15] F. Cen, “Design of degree distributions for LDPCA codes,” *IEEE Commun. Lett.*, vol. 13, no. 7, pp. 525-527, Jul. 2009.

[16] M. Jang, J. W. Kang, and S. Kim, “A Design of Rate-Adaptive LDPC Codes for Distributed Source Coding Using PEG Algorithm,” in *Proc. of MILCOM*, Nov. 2010.

[17] J. J. Micallef, R. A. Farrugia, and C. J. Debono, “Improved rate-adaptive codes for distributed video coding,” in *Proc. of VCIP*, Nov. 2011.

[18] J. J. Micallef, R. A. Farrugia, and C. J. Debono, “LDPCA code construction for Slepian-Wolf coding,” *IEEE Commun. Lett.*, vol. 15, no. 10, pp. 1100-1103, Oct. 2011.

[19] C. Brites, J. Ascenso, and F. Pereira, “Improving transform domain Wyner-Ziv video coding performance,” in *Proc. of ICASSP*, May 2006.

[20] C. Brites, and F. Pereira, “Correlation noise modeling for efficient pixel and transform domain Wyner-Ziv video coding,” *IEEE Trans. on CSVT*, vol. 18, no. 9, pp. 1177-1190, Jul. 2008.

[21] D. Kubasov, J. Nayak and C. Guillemot, “Optimal reconstruction in Wyner-Ziv video coding with multiple side information,” in *Proc. of MMSP*, Oct. 2007.

[22] C. Brites, and F. Pereira, “An efficient encoder rate control solution for transform domain Wyner–Ziv video coding,” *IEEE Trans. on CSVT*, vol. 21, no. 9, pp. 1278–1292, Sept. 2011.

[23] J. J. Micallef, R. A. Farrugia, and C. J. Debono, “Reducing correlation noise in Wyner-Ziv video coding,” in *Proc. of EUSIPCO*, Aug. 2012.

[24] D. Kubasov, K. Lajnef, and C. Guillemot, “A Hybrid encoder/Decoder Rate Control for a Wyner–Ziv Video Codec with a Feedback Channel,” in *Proc. of MMSP*, Oct. 2007.

[25] ITU-T Rec. G.7041. Generic Framing Procedure (GFP), 2003.

[26] B. Lin, “Correcting single-bit errors with CRC8 in ATM cell headers,” Computing Platform Division, Freescale Semiconductor, East Kilbride, Scotland, Doc. AN2918, Rev. 0, Jun. 2005.

[27] P. Yun, G. Ning, and D. Zaiwang, “CRC look-up table optimization for single-bit error Correction,” *Tsinghua Science and Technology,* vol. 12, no. 5, pp. 620-623, Oct. 2007.

[28] S. Babaie, A. K. Zadeh, S. H. Es-hagi, and N. J. Navimipour, “Double bits error correction using CRC method,” in *Proc. of SKG,* Oct.2009.

[29] P. Koopman, and T. Chakravarty, “Cyclic redundancy code (CRC) polynomial selection for embedded networks,” in *Proc. of DSN*, Jun. 2004.