Reduced Decoder Complexity and Latency in Pixel-Domain Wyner-Ziv Video Coders

Marleen Morbee · Antoni Roca · Josep Prades-Nebot · Aleksandra Pižurica · Wilfried Philips

Abstract In some video coding applications, it is desirable to reduce the complexity of the video encoder at the expense of a more complex decoder. Wyner-Ziv (WZ) video coding is a new paradigm that aims to achieve this. To allocate a proper number of bits to each frame, most WZ video coding algorithms use a feedback channel, which allows the decoder to request additional bits when needed. However, due to these multiple bit requests, the complexity and the latency of WZ video decoders increase massively. To overcome these problems, in this paper we propose a rate allocation (RA) algorithm for pixel-domain WZ video coders. This algorithm estimates at the encoder the number of bits needed for the decoding of every frame while still keeping the encoder complexity low. Experimental results show that, by using our RA algorithm, the number of bit requests over the feedback channel - and hence, the decoder complexity and the latency - are significantly reduced. Meanwhile, a very near-to-optimal rate-distortion performance is maintained.

Marleen Morbee · Aleksandra Pižurica · Wilfried Philips
TELIN-IPI-IBBT, Ghent University, Sint-Pietersnieuwstraat 41, 9000 Ghent, Belgium
Tel.: +32 9 264 42 25
Fax: +32 9 264 42 95
E-mail: {marleen.morbee, aleksandra.pizurica, wilfried.philips}@telin.ugent.be

Antoni Roca · Josep Prades-Nebot
GTS-ITEAM, Universidad Politécnica de Valencia, Camino de Vera s/n, 46022 Valencia, Spain
Tel.: +34 96 387 93 02
Fax: +34 96 387 73 09
E-mail: anrope2@teleco.upv.es, jprades@dcom.upv.es

1 Introduction

Some video applications, e.g., wireless low-power surveillance, wireless PC cameras, multimedia sensor networks, and mobile camera phones, require low-complexity coders. Distributed video coding is a new paradigm that fulfills this requirement by performing intra-frame encoding and inter-frame decoding [18]. Hence, most of the computational load is moved from the encoder to the decoder, since in this case the distributed video *de*coders (and not the *en*coders) perform motion estimation and motion compensated interpolation. Two theorems from information theory, namely the Slepian-Wolf theorem [21] for lossless distributed source coding and the Wyner-Ziv (WZ) theorem [25] for lossy source coding with side information, suggest that such a system with intra-frame encoding and inter-frame decoding can come close to the efficiency of a traditional inter-frame encoding-decoding system.

The most common distributed video coders are WZ video coders implemented with error correcting codes such as syndrome codes [18, 26, 27], turbo codes [1–10, 13, 17, 19, 22, 23] and low-density parity-check (LDPC) codes [11, 14, 15, 24, 26, 27]. Some proposed coding schemes apply WZ coding to the pixel values of the video signal and are therefore called pixel-domain Wyner-Ziv (PDWZ) video coders [3–8, 10, 13, 17, 19, 22, 23]; other approaches exploit the statistical dependencies within a frame by applying an image transform and are categorized as transform-domain WZ video coders [2, 9, 11, 14, 18, 26, 27]. In this paper, we focus on the turbo code-based PDWZ video coding architecture, as it is well known in literature [4, 5, 7, 8, 13, 17, 23].

One of the most difficult tasks in WZ video coding is allocating a proper number of bits to encode each video frame. This is mainly because the encoder does not have access to the motion estimation information of the decoder and because small variations in the allocated number of bits can cause large changes in distortion. Most WZ video coders solve this problem by using a feedback channel (FBC), which allows the decoder to request additional bits from the encoder when needed. In this way an optimal rate is allocated; however, this solution has several drawbacks. Firstly, due to the multiple bit requests (and the corresponding multiple decodings) the computational complexity of the decoder increases significantly. In [6], it is shown that the overall workload in WZ video coding often exceeds that of conventional coders, such as H.264. Secondly, a latency is introduced since the use of the feedback channel and the bit requests implies a certain delay in the decoding of each frame [8].

To overcome these problems, in this paper, we propose a rate allocation (RA) algorithm for PDWZ video coders. This algorithm reduces the number of bit requests from the decoder over the feedback channel and simultaneously keeps the computational load of the encoder low. The final aim is to reduce the decoder complexity and the latency to a minimum, while maintaining very near-to-optimal rate-distortion (RD) performance. The proposed method is related to our previous work [17], where we studied a rate allocation algorithm for PDWZ video coders without feedback channel. In this paper, however, we focus on the PDWZ video coder with feedback channel. We utilize this feedback channel to improve the RA and to achieve very near-to-optimal RD performance while at the same time eliminating the main feedback channel inconveniences, i.e., its negative impact on latency and decoder complexity. Moreover, the PDWZ video coder used in this work has been improved compared to [17] in three respects. Firstly, in this work we take into account previously decoded bit planes (BPs) for the turbo decoding of each BP. Secondly, the estimation of the encoding rate in [17] was based on experimentally obtained performance graphs of the turbo codes, while in
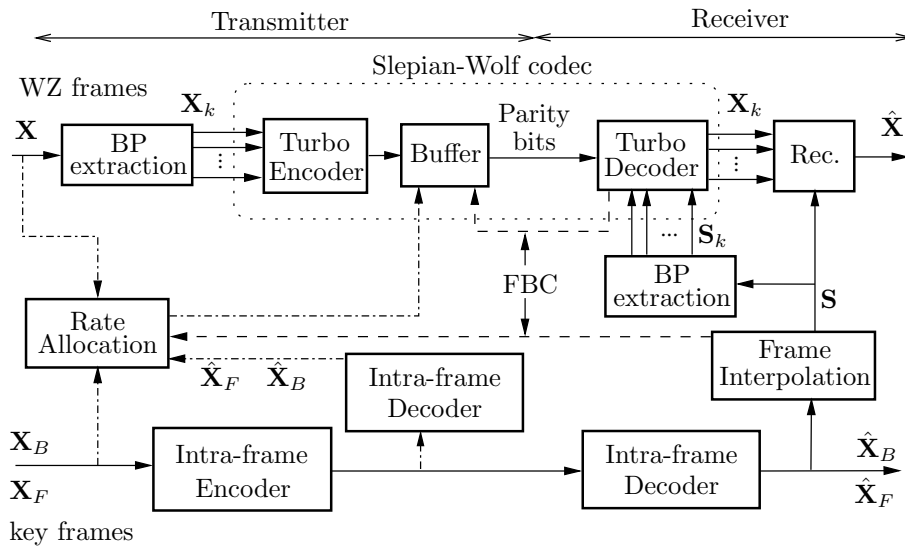
Fig. 1 General block diagram of a scalable PDWZ video coder.

this paper we derive expressions for the encoding rate founded on information theory concepts. Thirdly, the estimation of the variance for the assumed Laplacian model of the correlation noise has been refined, since we dispose of a feedback channel which allows us to transmit to the encoder an additional decoder estimate of the variance.

The paper is organized as follows. In Section 2, we study the basics of PDWZ video coding. In Section 3, we study the decoder complexity and the latency, and we discuss how they are influenced by the number of bit requests. In Section 4, we describe the RA algorithm. Then, in Section 5, we experimentally study the RD performance of a PDWZ video coder with feedback channel that allocates bits with our RA algorithm, and we measure the reduction of the number of bit requests. Finally, the conclusions are presented in Section 6.

2 Pixel-domain Wyner-Ziv video coding

2.1 General scheme of a scalable PDWZ video coder

In WZ video coding, the frames are organized into key frames and WZ frames. The key frames are coded using a conventional intra-frame coder. The WZ frames are coded using the WZ paradigm, i.e., they are intra-frame encoded, but they are conditionally decoded using side information (Figure 1). In most WZ video coders, the odd frames are encoded as key frames, and the even frames are encoded as WZ frames [3,4]. Coding and decoding is done unsequentially in such a way that, before decoding the WZ frame $\mathbf{X}$, the preceding and succeeding key frames ($\mathbf{X}_\mathrm{B}$ and $\mathbf{X}_\mathrm{F}$) have already been transmitted and decoded. Thus, the receiver can obtain a good approximation $\mathbf{S}$ of $\mathbf{X}$ by interpolating its two closest decoded frames ($\hat{\mathbf{X}}_\mathrm{B}$ and $\hat{\mathbf{X}}_\mathrm{F}$). $\mathbf{S}$ is used as part of the side information to conditionally decode $\mathbf{X}$, as will be explained below.

The WZ video coders can be divided into two classes: the scalable coders [4, 8, 17], and the non-scalable coders [3]. The scalable coders have the advantages that the rate can be flexibly adapted and that the rate control is easier than in the non-scalable case. In this paper, we focus on the practical scalable PDWZ video coder depicted in Figure 1 [4, 8, 17]. In this scheme, we first extract the M BPs $\mathbf{X}_k$ $(1 \leq k \leq M)$ from the WZ frame $\mathbf{X}$. M is determined by the number of bits by which the pixel values of $\mathbf{X}$ are represented. Subsequently, only the m most significant BPs $\mathbf{X}_k$ $(1 \leq k \leq m, 1 \leq m \leq M)$ are encoded independently of each other by a Slepian-Wolf coder [21]. The other BPs $\mathbf{X}_k$ $(m + 1 \leq k \leq M)$ are not encoded and are simply discarded. That way, a certain amount of compression is achieved. The higher m, the higher the encoding rate, but the lower the distortion. The value of parameter m can be fixed along the sequence [4, 5, 10, 17, 22] or can be adaptively changed to fulfil the coding constraints [19]. The transmission and decoding of BPs is done in order of significance (the most significant BPs are transmitted and decoded first). The Slepian-Wolf coding is implemented with efficient channel codes that yield the parity bits of $\mathbf{X}_k$, which are *partially* transmitted over the channel thereby achieving compression. At the receiver side, the Slepian-Wolf decoder obtains the original BP $\mathbf{X}_k$ from the transmitted parity bits, the corresponding BP $\mathbf{S}_k$ extracted from the interpolated frame $\mathbf{S}$, and the previously decoded BPs $\{\mathbf{X}_1, \ldots, \mathbf{X}_{k-1}\}$. Note that $\mathbf{S}_k$ can be considered the result of transmitting $\mathbf{X}_k$ through a noisy *virtual channel*. The Slepian-Wolf decoder is a channel decoder that recovers $\mathbf{X}_k$ from its noisy version $\mathbf{S}_k$. Finally, the decoder obtains the reconstruction $\hat{X}$ of each pixel $X \in \mathbf{X}$ by using the decoded bits $X_k \in \mathbf{X}_k$ $(k = 1, \ldots, m)$ and the corresponding pixel S of the interpolated frame $\mathbf{S}$ through

$$\hat{X} = \begin{cases} X_{\mathrm{L}}, & S < X_{\mathrm{L}} \\ S, & X_{\mathrm{L}} \leq S \leq X_{\mathrm{R}} \\ X_{\mathrm{R}}, & S > X_{\mathrm{R}} \end{cases} \tag{1}$$

with

$$X_{\mathrm{L}} = \sum_{i=1}^{m} X_i 2^{8-i} \quad \text{and} \quad X_{\mathrm{R}} = X_{\mathrm{L}} + 2^{8-m} - 1. \tag{2}$$

2.2 Turbo code-based scalable PDWZ video coder

In this paper, the Slepian-Wolf coder is implemented with turbo codes (TC). The virtual channel is assumed to be symmetric and the symbols of the BPs are binary, so the virtual channel is modelled as a binary symmetric channel. To decode the kth transmitted BP $\mathbf{X}_k$ of a WZ frame $\mathbf{X}$, the turbo decoder needs to compute the error probability of each bit of the BP $\mathbf{S}_k$. The way to do this is related to the method proposed in [6]. In our previous work [17], only $\mathbf{S}$ was used as side information to obtain the error-free BP $\mathbf{X}_k$ from its parity bits. In this paper, however, apart from the received parity bits and the interpolated frame $\mathbf{S}$, we also take into account the information provided by the previously decoded BPs $\{\mathbf{X}_1, \ldots, \mathbf{X}_{k-1}\}$ of $\mathbf{X}$, as is done in [26, 27]. In order to efficiently combine all the available pieces of information for the computation of the error probability of each bit of the BP $\mathbf{S}_k$, we need to statistically model the correlation noise frame $\mathbf{U} = \mathbf{X} - \mathbf{S}$ [12]. As in [3, 4, 17], we assume that a pixel value $U \in \mathbf{U}$ follows a Laplacian distribution with a probability density function (pdf)

$$p(U) = \frac{\alpha}{2} e^{(-\alpha|U|)} \tag{3}$$

where $\alpha = \sqrt{2}/\sigma$ and σ is the standard deviation of the correlation noise frame $\mathbf{U}$. From the $k-1$ most significant bits $\{X_1, \ldots, X_{k-1}\}$ of $X \in \mathbf{X}$ that have already been transmitted and error-free decoded, the decoder knows that X lies in the quantization interval $[X_\mathrm{L}, X_\mathrm{R}]$ where X_L and X_R are as in (2) with $m = k-1$. Hence, the conditional pdf of X given S and $X_\mathrm{L} \leq X \leq X_\mathrm{R}$ is

$$p_\mathrm{dec}(X|S, X_\mathrm{L} \leq X \leq X_\mathrm{R}) = \begin{cases} \dfrac{\frac{\alpha}{2}e^{-\alpha|X-S|}}{\mathrm{P}(X_\mathrm{L} \leq X \leq X_\mathrm{R}|S)} & \text{if } X_\mathrm{L} \leq X \leq X_\mathrm{R} \\ \\ 0 & \text{otherwise} \end{cases} \qquad (4)$$

where the probability $\mathrm{P}(X_\mathrm{L} \leq X \leq X_\mathrm{R}|S)$ can be computed by integrating (3)

$$\mathrm{P}(X_\mathrm{L} \leq X \leq X_\mathrm{R}|S) = \int_{X_\mathrm{L}}^{X_\mathrm{R}} \frac{\alpha}{2} e^{-\alpha|X-S|}\, dX. \qquad (5)$$

To derive the error probability of the kth bit S_k of the pixel value S, we first observe that the decoded bit X_k will further shrink the quantization interval of X in such a way that

$$\begin{cases} X \in [X_\mathrm{L}, X_\mathrm{C}] & \text{if } X_k = 0 \\ X \in [X_\mathrm{C} + 1, X_\mathrm{R}] & \text{if } X_k = 1 \end{cases} \qquad (6)$$

where

$$X_\mathrm{C} = \left\lfloor \frac{X_\mathrm{L} + X_\mathrm{R}}{2} \right\rfloor \qquad (7)$$

with $\lfloor y \rfloor$ denoting the floor function that returns the highest integer less than or equal to y. For the pixel value X from which the bit X_k needs to be decoded, the values X_L, X_R, and X_C can be computed from the previously decoded bits $\{X_1, \ldots, X_{k-1}\}$ using (2) with $m = k-1$ and (7). The estimate $X_k = S_k$ is erroneous if $S_k = 0$ and $X \in [X_\mathrm{C} + 1, X_\mathrm{R}]$ or if $S_k = 1$ and $X \in [X_\mathrm{L}, X_\mathrm{C}]$. Hence, the error probability of the kth bit of S is estimated through

$$P_\mathrm{e}(S_k) = \begin{cases} \displaystyle\int_{X_\mathrm{C}+0.5}^{X_\mathrm{R}} p_\mathrm{dec}(X|S, X_\mathrm{L} \leq X \leq X_\mathrm{R})\, dX & \text{if } S_k = 0, \\ \\ \displaystyle\int_{X_\mathrm{L}}^{X_\mathrm{C}+0.5} p_\mathrm{dec}(X|S, X_\mathrm{L} \leq X \leq X_\mathrm{R})\, dX & \text{if } S_k = 1. \end{cases} \qquad (8)$$

Note that the integration intervals are extended by 0.5 in order to cover the whole interval $[X_\mathrm{L}, X_\mathrm{R}]$. For the first BP $\mathbf{X}_1$, no previous BPs have been transmitted and decoded and, consequently, $X_\mathrm{L} = 0$, $X_\mathrm{R} = 255$, and $X_\mathrm{C} = 127$ for all the pixels.

3 Decoder complexity and latency

In PDWZ video coders, the optimum rate R^* is the *minimum* rate necessary to losslessly decode the BPs $\mathbf{X}_k$ ($k = 1, \ldots, m$). The use of a rate higher than R^* does not lead to a reduction in distortion, but only to an unnecessary bit expense. On the other hand, encoding with a rate lower than R^* can cause the introduction of a large number of errors in the decoding of $\mathbf{X}_k$, which can greatly increase the distortion. This is because of the threshold effect of the channel codes used in WZ video coders.

A common RA solution adopted in WZ video coders is the use of a feedback channel and a rate-compatible punctured turbo code [20]. In this configuration, the turbo encoder generates all the parity bits for the BPs to be encoded, saves these bits in a buffer (see Figure 1), and divides them into parity bit sets. The size of a parity bit set is N/T_{punc}, where T_{punc} is the puncturing period of the rate-compatible punctured turbo code and N is the number of pixels in each frame. To determine the adequate number of parity bit sets to send for a certain BP $\mathbf{X}_k$, the encoder first transmits one parity bit set from the buffer. Then, if the decoder detects that the residual error probability Q_k is above a threshold t [17], it requests an additional parity bit set from the buffer through the feedback channel. This transmission-request process is repeated until $Q_k < t$. If we denote by K_k the number of transmitted parity bit sets, then the encoding rate R_k for BP $\mathbf{X}_k$ is

$$R_k = r \ K_k \ \frac{N}{T_{\mathrm{punc}}}, \tag{9}$$

with r being the frame rate of the video.

This solution has several drawbacks. Firstly, the transmission-request process increases the decoder complexity drastically since multiple parity bit decodings have to be performed for each BP of the WZ frame. More specifically, when we denote by $O_{\mathrm{dec},k}$ the number of operations needed for the turbo decoding of the kth BP, then the number of operations O_{dec} for the decoding of a WZ frame is [6]

$$O_{\mathrm{dec}} = \sum_{k=1}^{m} O_{\mathrm{dec},k} = \sum_{k=1}^{m} 2P_{\mathrm{TC}}(W_k + 1), \tag{10}$$

where W_k is the number of bit requests for the decoding of the kth BP and P_{TC} is a variable combining the parameters of the rate-compatible punctured turbo code [6]. In our setup, P_{TC} is fixed for all the decodings and is independent of W_k, so the decoder complexity depends on the number of bit requests needed for the decoding of the BPs through the factor $\sum_{k=1}^{m}(W_k+1)$. W_k is determined mainly by the correlation between the interpolated frame $\mathbf{S}$ and the WZ frame $\mathbf{X}$ for the kth BP; this correlation is usually high for the most significant BP and decreases for less significant BPs.

Secondly, the feedback channel increases the coding latency [8]. In fact, after sending a parity bit set, the encoder has to wait for an answer from the decoder before it can send more bits. In this work, we assume the network to be perfect, so the delay introduced by networking effects on the transmission of the bits is set to 0. Then, the round trip delay per bit request depends on the time needed for one turbo decoding. Consequently, when we denote by $L_{\mathrm{dec},k}$ the latency for the decoding of the kth BP, the total latency for the decoding of a WZ frame can be expressed as

$$L_{\mathrm{dec}} = \sum_{k=1}^{m} L_{\mathrm{dec},k} = \sum_{k=1}^{m} \frac{2P_{\mathrm{TC}}(W_k + 1)}{v} \tag{11}$$

where W_k and P_{TC} are the same as in (10) and v is the processor speed (in operations/s). In our setup, P_{TC} and v are fixed for all the decodings and are independent of W_k, so the total latency depends on the number of bit requests needed for the decoding of the BPs through the factor $\sum_{k=1}^{m}(W_k+1)$.

As shown by (10) and (11), both the decoder complexity and the latency can be reduced by minimizing the number of bit requests, or more specifically, by reducing

the factor $\sum_{k=1}^{m}(W_k+1)$. Note that this factor yields a relative reduction of decoder complexity and latency, which is independent of the specific implementation parameters of the coder, such as P_{TC} and v. In the following section, we propose a novel rate allocation algorithm for PDWZ video coders with feedback channel, which provides an estimate of the optimal number of parity bit sets that have to be transmitted, thereby reducing the number of bit requests to a minimum.

4 The proposed rate allocation algorithm

The main idea of the proposed method is to estimate at the encoder side, for each BP of the WZ frames, the optimal (i.e. the minimal required) number of parity bits. By allocating the estimated optimal number of parity bits at the encoder, the number of bit requests over the feedback channel will be reduced. The proposed approach attempts to avoid overestimation of the optimal number of parity bits. This is an important aspect because if more bits than needed are allocated, there will not be a decrease in distortion but only an unnecessary bit expense (as explained in Section 3).

As every BP of a WZ frame $\mathbf{X}$ is separately encoded, a different rate R_k must be allocated to each BP $\mathbf{X}_k$. A lower bound of the appropriate encoding rate R_k is estimated based on the adopted Laplacian correlation model (3) and the entropy of $\mathbf{X}_k$ conditional on the interpolated frame $\mathbf{S}$ and the previously decoded BPs $\{\mathbf{X}_1, \ldots, \mathbf{X}_{k-1}\}$. Hence, our algorithm consists of two steps. Firstly, we make an estimate $\hat{\sigma}^2$ of the parameter σ^2 of the Laplacian model (Section 4.1). Secondly, for each BP $\mathbf{X}_k$, we use $\hat{\sigma}^2$ to estimate a lower bound of the encoding rate R_k for BP $\mathbf{X}_k$ by means of the conditional entropy (Section 4.2). In the following, we explain both steps of our RA algorithm in more detail.

4.1 Estimation of σ^2

The true value of σ^2 can only be obtained by combining information that is only available at the encoder (the original frame $\mathbf{X}$) and information that is only available at the decoder (the interpolated frame $\mathbf{S}$). The encoder could obtain $\mathbf{S}$ by motion compensated interpolation, but, of course, that would heavily increase the encoder complexity which is undesirable in WZ video coding. Thus, neither the decoder, nor the encoder can obtain the true value of σ^2. In [12], the authors propose to estimate the variance by interchanging information between encoder and decoder. That way, the coder can estimate the variance with high precision, but at the expense of a massive increase in feedback channel communication and the consequent overhead and delay. In our approach, however, we would like to keep the feedback channel overhead and delay as low as possible. Therefore, our idea is to estimate σ^2 separately at the encoder (Section 4.1.1) and at the decoder (Section 4.1.2) side. In a next step we then combine the two estimates via the feedback channel (Section 4.1.3). More specifically, we will transmit for each frame an estimate $\hat{\sigma}_{\text{dec}}^2$ of σ^2 made at the decoder to the encoder through the feedback channel, so that both estimates are available at the encoder side. Note that transmitting $\hat{\sigma}_{\text{dec}}^2$ introduces an overhead and a round trip delay. This overhead, however, is negligible and the latency is well compensated for by the reduction of the number of bit requests. By combining $\hat{\sigma}_{\text{dec}}^2$ with the estimate $\hat{\sigma}_{\text{enc}}^2$ at the encoder, the risk of overestimating σ^2 is reduced.

4.1.1 Encoder estimate $\hat{\sigma}^2_{\text{enc}}$

The estimation at the encoder should be very simple in order to avoid significantly increasing the encoder complexity. We adopt the approach of [17], but we take the coding of the key frames into account. Then, $\hat{\sigma}^2_{\text{enc}}$ is the mean squared error (MSE) between the current WZ frame and the average of the two closest decoded key frames:

$$\hat{\sigma}^2_{\text{enc}} = \frac{1}{N} \sum_{(v,w)\in\mathbf{X}} \left(\mathbf{X}(v,w) - \frac{\hat{\mathbf{X}}_{\text{B}}(v,w) + \hat{\mathbf{X}}_{\text{F}}(v,w)}{2}\right)^2 \qquad (12)$$

with N denoting the number of pixels in each frame. The decoded key frames are obtained by the intra-frame decoding unit at the encoder site (see Figure 1).

4.1.2 Decoder estimate $\hat{\sigma}^2_{\text{dec}}$

At the decoder, motion compensated interpolation is performed on a block-basis in order to generate the interpolated frame $\mathbf{S}$ [4]. During the interpolation process of a block of the frame $\mathbf{X}$, the best matching blocks in $\hat{\mathbf{X}}_{\text{B}}$ and $\hat{\mathbf{X}}_{\text{F}}$ are searched using a minimum MSE criterion. Assuming linear motion between $\hat{\mathbf{X}}_{\text{B}}$ and $\hat{\mathbf{X}}_{\text{F}}$, the pixel values of both frames contribute equally to the interpolated pixels that constitute the frame $\mathbf{S}$. Then, the estimate of the variance between the original frame $\mathbf{X}$ and the interpolated frame $\mathbf{S}$ is [10]:

$$\hat{\sigma}^2_{\text{dec}} = \frac{1}{4}\frac{1}{N} \sum_{(v,w)\in\mathbf{S}} \left(\hat{\mathbf{X}}_{\text{B}}(v - dv, w - dw) - \hat{\mathbf{X}}_{\text{F}}(v + dv, w + dw)\right)^2 \qquad (13)$$

where (v,w) corresponds to the pixel location in $\mathbf{S}$ and $(2dv, 2dw)$ denotes the motion vector between the corresponding pixels $(v - dv, w - dw)$ and $(v + dv, w + dw)$ in $\hat{\mathbf{X}}_{\text{B}}$ and $\hat{\mathbf{X}}_{\text{F}}$, respectively.

4.1.3 Combining $\hat{\sigma}^2_{\text{enc}}$ and $\hat{\sigma}^2_{\text{dec}}$

Since we want to avoid overestimating the optimal number of parity bits, we need to avoid overestimating σ^2. Therefore, we propose

$$\hat{\sigma}^2 = \min(\hat{\sigma}^2_{\text{enc}}, \ \hat{\sigma}^2_{\text{dec}}). \qquad (14)$$

Experimental results on ten test sequences show that, in 97% of the cases, $\hat{\sigma}^2 \leq \sigma^2$, which is exactly what is required for our purpose.

4.2 Estimation of the encoding rates $\{R_k\}$

The estimation of the encoding rates R_k for the BPs $\mathbf{X}_k$ is related to the algorithm described in Section 2.2 to estimate the error probabilities of the bits of the BPs $\mathbf{S}_k$ (extracted from $\mathbf{S}$) at the decoder. Note that at the encoder, we know all the BPs of frame $\mathbf{X}$ but not the corresponding interpolated frame $\mathbf{S}$; at the decoder, however, we know $\mathbf{S}$ but only the previously decoded BPs of $\mathbf{X}$. More specifically, to estimate the required number of bits to encode a bit X_k of the kth BP $\mathbf{X}_k$, we observe that when encoding the kth bit of a pixel $X \in \mathbf{X}$ the most significant $k-1$ bits of this pixel X have

already been decoded without errors. Hence, the decoder is aware of $\{X_1, \ldots, X_{k-1}\}$ and the corresponding pixel S of the interpolated frame $\mathbf{S}$. Consequently, the minimum number of bits $B(X_k)$ to encode a bit X_k of BP $\mathbf{X}_k$ is the entropy of X_k conditional on S and the previously decoded bits $\{X_1, \ldots, X_{k-1}\}$:

$$B(X_k) = H(X_k | S, X_1, \ldots, X_{k-1}) \tag{15}$$

Applying the chain rule, we derive

$$B(X_k) = H(X_1, \ldots, X_k | S) - \sum_{i=1}^{k-1} B(X_i) \tag{16}$$

and further,

$$B(X_k) = \sum_{s=0}^{255} f_S(s) H(X_1, \ldots, X_k | S = s) - \sum_{i=1}^{k-1} B(X_i) \tag{17}$$

$$= -\sum_{s=0}^{255} f_S(s) \sum_{x_1=0}^{1} \ldots \sum_{x_k=0}^{1} \mathrm{P}(X_1 = x_1, \ldots, X_k = x_k | S = s) \quad \ldots$$

$$\ldots \quad \log_2 \mathrm{P}(X_1 = x_1, \ldots, X_k = x_k | S = s) - \sum_{i=1}^{k-1} B(X_i) \tag{18}$$

where $f_S(s)$ is the probability mass function (pmf) of $S \in \mathbf{S}$. As the interpolated frame $\mathbf{S}$ is not available at the encoder, we use instead of $f_S(s)$ the pmf of X, $f_X(x)$, since both pmfs can be considered very similar. By s and x we denoted the possible outcomes of X and S which are $\in \{0, \ldots, 255\}$ and by $x_1, \ldots, x_k$ we denoted the possible outcomes of $X_1, \ldots, X_k$ which are $\in \{0, 1\}$. In practice, we estimate $f_X(x)$ through the histogram of the WZ frame $\mathbf{X}$. $\mathrm{P}(X_1, \ldots, X_k | S)$ can be computed from the assumed Laplacian pdf of U (3) with the estimated parameter $\hat{\sigma}^2$ (14). More concrete,

$$\mathrm{P}(X_1, \ldots, X_k | S) = \mathrm{P}(X_\mathrm{L} \leq X \leq X_\mathrm{R} | S) \tag{19}$$

where $\mathrm{P}(X_\mathrm{L} \leq X \leq X_\mathrm{R} | S)$ is as in (5) with X_L and X_R as in (2) (with $m = k$). By using (18), we can now compute $B(X_k)$ by calculating recursively $B(X_i)$ ($i = 1, \ldots, k-1$) starting from $i = 1$.

Finally, the minimum encoding rate R_k for BP $\mathbf{X}_k$ is

$$R_k = r \, N \, B(X_k) \tag{20}$$

with r being the frame rate of the video and N being the number of pixels in each frame.

By using (9), the number of parity bit sets to be transmitted K_k is estimated through

$$K_k = \left\lceil \frac{R_k}{N / T_{\mathrm{punc}}} \right\rceil + 1 \tag{21}$$

where T_{punc} is the puncturing period of the rate-compatible punctured turbo code and $\lceil y \rceil$ denotes the ceiling function that returns the smallest integer not less than y. The last term of the sum is a rate margin which is applied to compensate for the sub-optimality of the adopted turbo code.

5 Experimental results and discussion

In this section, we first experimentally study the RD performance of a PDWZ video coder with feedback channel that allocates bits with our RA algorithm (RA-PDWZ video coder) and compare it to the same PDWZ video coder with feedback channel that does not use our RA algorithm (optimal rate allocation). Subsequently, we show how our RA algorithm reduces the number of bit requests from the decoder over the feedback channel.

The PDWZ video coder used in the experiments, first decomposes each WZ frame into its 8 BPs. Then, the m most significant BPs are separately encoded by using a rate-compatible punctured turbo code; the other BPs are thrown away. In our experiments, m is chosen to be $\in \{0, \ldots, 3\}$. The turbo coder is composed of two identical constituent convolutional encoders of rate $1/2$ with generator polynomials $(1, 33/31)$ in octal form. The puncturing period was set to 32 which allowed our RA algorithm to allocate parity bit multiples of $N/32$ bits to each BP, where N is the number of pixels in each frame. The key frames were intra-coded using H.263+ [16] with quantization parameter QP. We used the H.263+ software implementation of the University of British Columbia (UBC) (Version 3). The interpolated frames were generated at the decoder with the interpolation tools described in [4]. The threshold t for Q_k was set to 10^{-3}. Note that this PDWZ video coder shares its main characteristics with the coders proposed in [4, 5, 8, 13, 23]. Hence, the efficiency of our coder is expected to be close to that of the coders of [4, 5, 8, 13, 23]. The authors have verified that this holds for the test sequences used later in this section for which the coding efficiency of [4, 5, 8, 13, 23] has been published. As an illustration, we have plotted the rate-distortion performance of our coder and the coder of [13] for the first 100 frames of the test video sequence *Foreman* in Figure 2. The resolution of the sequence is QCIF (176×144 pixels/frame) and the frame rate is 30 frames/s. In this experiment, only the rate and distortion of the luminance of the WZ frames is considered. The WZ frame rate is 15 frames/s. To obtain the data shown in Figure 2, both coders used the same quantization parameters. In particular, the key frames are losslessly coded and each RD point corresponds to a fixed number of bit planes m sent for the encoding of the WZ frames ($m = 1, \ldots, 4$). Hence, Figure 2 provides a fair comparison between the coding efficiency of both coders. For the other sequences, the plots look similar. For the sake of conciseness, they are not shown here. For a comparison of the performance of our PDWZ video coder with existing conventional coding schemes we refer the reader to [4, 5, 13].

In Figure 3, we present a visual result of our PDWZ video coder. In this figure, we show for WZ frame $\mathbf{X}$ number 70 of the sequence *Foreman* (QCIF, 30 frames/s) its two adjacent decoded key frames $\hat{\mathbf{X}}_B$ and $\hat{\mathbf{X}}_F$ (coded using H.263+ with $QP = 10$), the interpolated frame $\mathbf{S}$ after motion estimation and motion compensated interpolation at the decoder, and the final reconstructed WZ frame $\hat{\mathbf{X}}$ when 2 BPs are transmitted (or in other words $m = 2$). Below the decoded frames the PSNR of the frame and the number of bits dedicated to the encoding of the frame are indicated. We observe that the quality of the decoded WZ frame is better than the quality of its adjacent decoded key frames, while the number of bits used for the encoding of the WZ frame is lower. For a detailed study of the partition of quality between key frames and WZ frames, we refer the reader to our previous work [19].

To assess the efficiency of our RA algorithm, we encoded several test sequences (QCIF, 30 frames/s) with the described RA-PDWZ video coder. For the plots, we only include the rate and distortion of the luminance of the WZ frames. The WZ frame
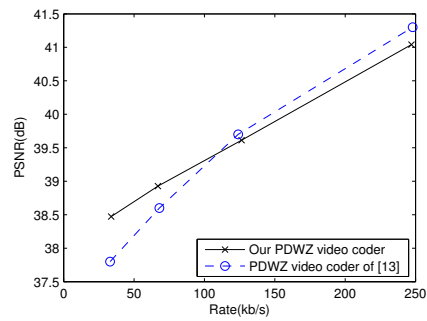
Fig. 2 Comparison between the rate-distortion performance of the PDWZ video coder of [13] and our PDWZ video coder for the first 100 frames of the Foreman sequence (QCIF, 30 frames/s). The key frames are losslessly coded.
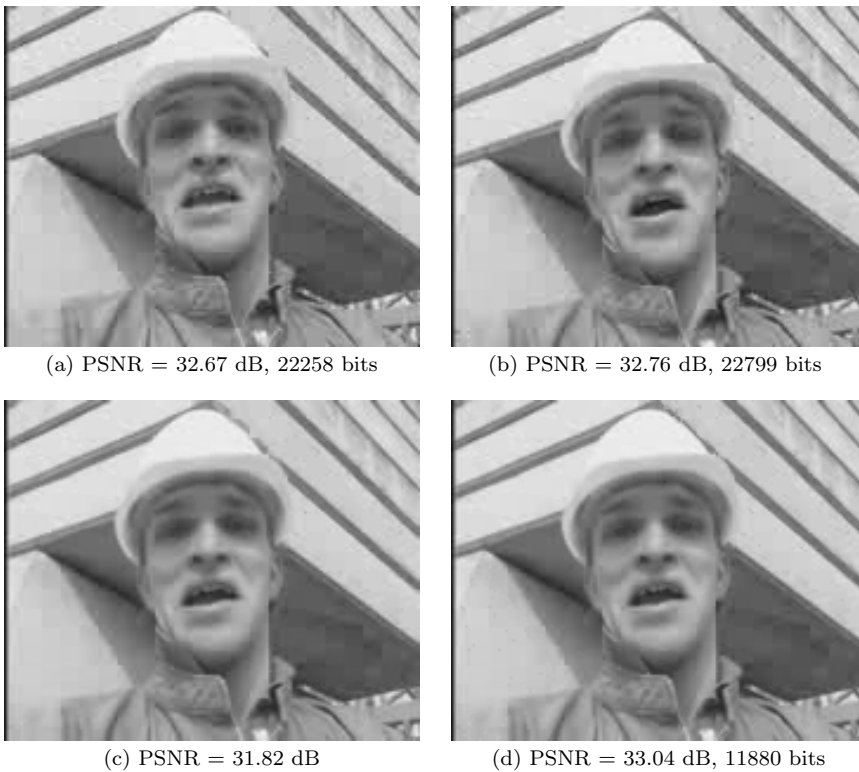

(a) PSNR = 32.67 dB, 22258 bits

(b) PSNR = 32.76 dB, 22799 bits

(c) PSNR = 31.82 dB

(d) PSNR = 33.04 dB, 11880 bits

Fig. 3 For WZ frame $\mathbf{X}$ number 70 of the sequence *Foreman*: (a) its preceding decoded key frame $\hat{\mathbf{X}}_{\mathrm{B}}$, (b) its succeeding decoded key frame $\hat{\mathbf{X}}_{\mathrm{F}}$, (c) the interpolated frame $\mathbf{S}$ after motion estimation and motion compensated interpolation at the decoder and (d) the reconstructed WZ frame $\hat{\mathbf{X}}$ for $m = 2$. The key frames are intra-coded with H.263+ ($QP = 10$).
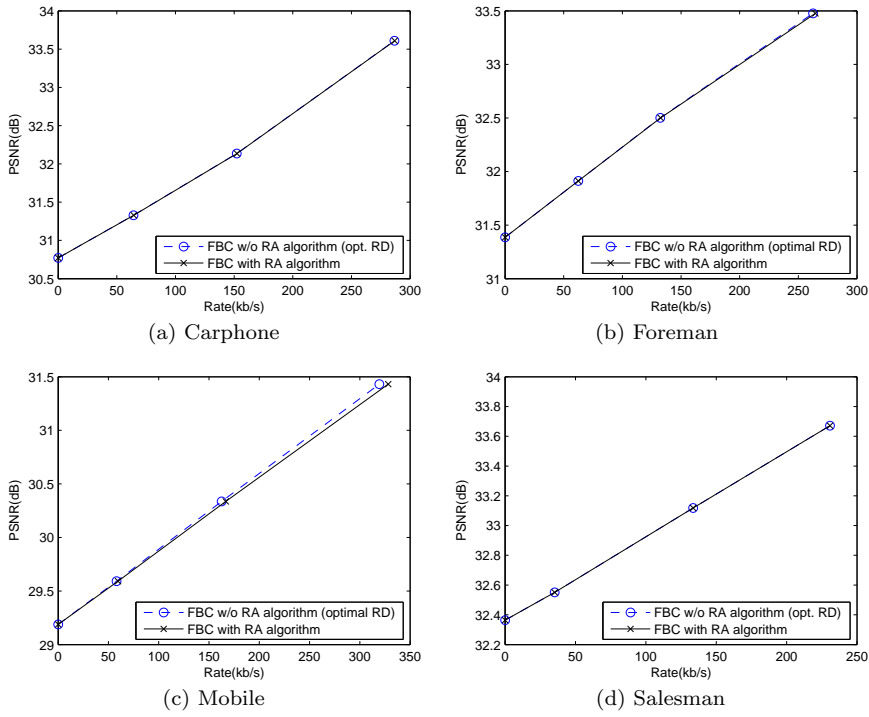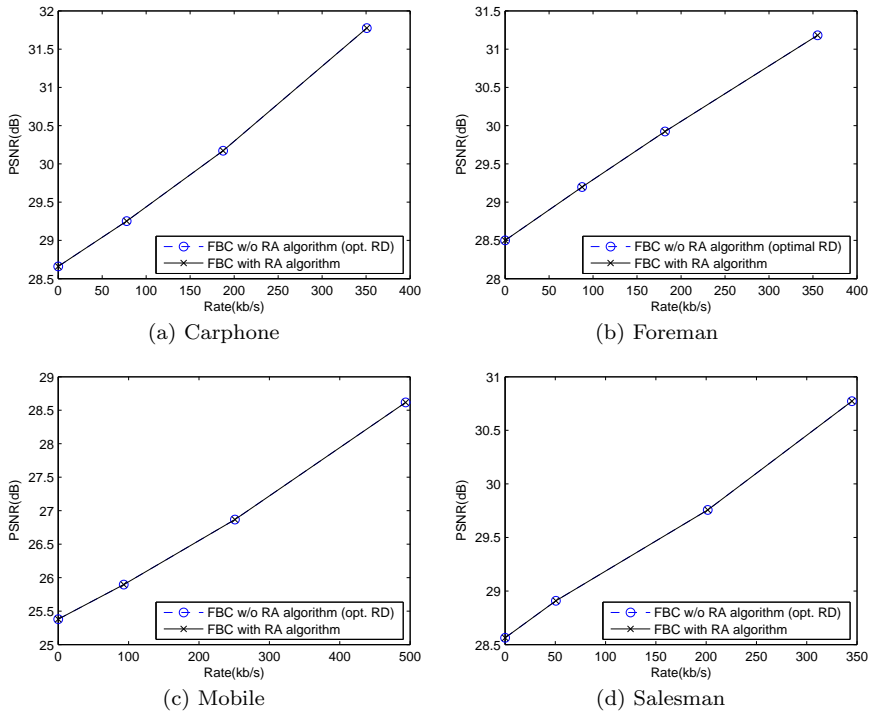
Fig. 4 RD performance of the RA-PDWZ video coder for the sequences (a) *Carphone*, (b) *Foreman*, (c) *Mobile* and (d) *Salesman*. Also the RD performance for the case of optimal rate allocation is shown. The key frames are intra-coded with H.263+ $(QP = 10)$.

Video sequence	$W_{k,\mathrm{opt}}$			$W_{k,\mathrm{RA}}$			f
	BP 1	BP 2	BP 3	BP 1	BP 2	BP 3	
Akiyo	1.49	3.00	7.00	0.49	1.04	4.03	1.69
Carphone	4.41	6.41	10.32	1.65	2.28	3.73	2.26
Coast	2.70	4.72	10.67	0.28	1.17	2.72	2.94
Container	8.07	1.14	6.31	3.76	0.14	3.47	1.79
Foreman	4.25	4.88	9.97	1.25	1.23	1.70	3.08
Hall	3.25	2.81	6.40	0.75	0.93	2.97	2.02
Mobile	3.89	7.80	12.20	0.02	0.16	0.17	8.04
Mother&Daughter	4.41	2.84	7.39	1.08	1.51	4.32	1.78
Salesman	1.97	7.28	7.19	0.92	4.49	3.82	1.59
Tennis	12.20	2.64	6.07	3.13	1.05	1.97	2.61

Table 1 Comparison of the average number of bit requests for the encoding of the kth BP $(k = 1, \ldots, 3)$ between a PDWZ video coder *with* $(W_{k,\mathrm{RA}})$ and *without* $(W_{k,\mathrm{opt}})$ our RA algorithm. f is the bit request reduction ratio (see (22)). The key frames are intra-coded with H.263+ $(QP = 10)$.

Fig. 5 RD performance of the RA-PDWZ video coder for the sequences (a) *Carphone*, (b) *Foreman*, (c) *Mobile* and (d) *Salesman*. Also the RD performance for the case of optimal rate allocation is shown. The key frames are intra-coded with H.263+ ($QP = 20$).

Video sequence	$W_{k,\mathrm{opt}}$			$W_{k,\mathrm{RA}}$			f
	BP 1	BP 2	BP 3	BP 1	BP 2	BP 3	
Akiyo	3.00	5.99	10.27	2.00	3.86	7.11	1.39
Carphone	5.55	8.23	12.74	2.65	3.90	5.85	1.92
Coast	4.34	7.57	15.68	1.61	3.57	7.16	1.99
Container	11.52	2.63	10.72	6.91	1.63	7.82	1.44
Foreman	6.36	6.94	13.60	3.28	3.07	4.83	2.11
Hall	5.95	4.64	10.57	3.31	2.71	6.94	1.51
Mobile	6.84	12.30	19.42	2.09	2.89	4.75	3.27
Mother&Daughter	7.13	4.52	10.82	3.73	3.18	7.77	1.44
Salesman	3.26	11.72	11.06	2.22	8.84	7.56	1.34
Tennis	14.64	3.94	9.10	5.18	2.30	4.70	2.02

Table 2 Comparison of the average number of bit requests for the encoding of the kth BP ($k = 1, \ldots, 3$) between a PDWZ video coder *with* ($W_{k,\mathrm{RA}}$) and *without* ($W_{k,\mathrm{opt}}$) our RA algorithm. f is the bit request reduction ratio (see (22)). The key frames are intra-coded with H.263+ ($QP = 20$).

rate is 15 frames/s. In Figures 4 and 5, we show the RD curves of *Carphone*, *Foreman*, *Mobile*, and *Salesman* when coded with the RA-PDWZ video coder, and we compare them with the corresponding RD curves when, for the given puncturing period, an optimal rate is allocated. This comparison is done for two different QP-values for the encoding of the key frames: $QP = 10$ (Figure 4) and $QP = 20$ (Figure 5). The value of the PSNR at rate 0 ($m = 0$) shows the average quality of the interpolated frame **S**. The RD points at higher rates correspond to an increasing number of BPs sent, more specifically, $m = 1, 2$ and 3. We observe that for all the sequences the RD performance of the RA-PDWZ video coder is very close to the optimal one.

Tables 1 and 2 show for ten test video sequences the average number of bit requests needed to decode the kth BP ($k = 1, \ldots, 3$) for the PDWZ video coder with optimal rate allocation ($W_{k,\text{opt}}$) and for the RA-PDWZ video coder ($W_{k,\text{RA}}$). In Table 1 the key frames are intra-coded with $QP = 10$, while in Table 2 the key frames are intra-coded with $QP = 20$. We observe that with our RA algorithm the number of bit requests is reduced significantly. Tables 1 and 2 also show for each sequence the average bit request reduction ratio f, with

$$f = \frac{\sum_{k=1}^{m}(W_{k,\text{opt}} + 1)}{\sum_{k=1}^{m}(W_{k,\text{RA}} + 1)}. \tag{22}$$

In Figure 6, we plot for the ten test sequences (and for $QP = 10$ and $QP = 20$) these average bit request reduction ratios f. In particular, we show how for each sequence this average bit request reduction ratio compares to the mean value μ over all the sequences. We also indicated the standard deviation σ. As can be expected, we observe for both $QP = 10$ and $QP = 20$ higher bit request reduction ratios (around and above μ) for sequences that need a higher amount of parity bit sets, i.e. when the amount of correlation noise is larger. This is mostly the case for sequences that contain a lot of motion and camera movement (e.g. *Carphone*, *Coast*, *Foreman*, *Mobile*, *Tennis*). Indeed, in these sequences the motion compensated interpolation between the two adjacent key frames is more difficult and yields a worse estimate of the frame to be encoded than in the case of sequences with less motion and recorded with a static camera (e.g. *Akiyo*, *Container*, *Hall*, *Mother&Daughter*, *Salesman*). Moreover, we observe that, even though the number of parity bits sets is generally higher for $QP = 20$ than for $QP = 10$, lower bit request reduction ratios are achieved for $QP = 20$ than for $QP = 10$. This is due to the fact that the rate is more significantly underestimated for $QP = 20$ than for $QP = 10$. The reason for this is as follows. As explained in Section 4, the rate allocation is based on two estimates of the variance, one made at the encoder (Section 4.1.1), and one made at the decoder (Section 4.1.2). For the encoder estimate the intra-coding noise is taken into account (the original frame intervenes in the estimate), while in the decoder estimate this error is not incorporated (the subtracted frames are both coded). Since for most of the frames the decoder estimate is the final estimate, the rate allocation is for many frames done without taking into account the noise introduced by the intra-coder, which results in an underestimation of the rate. The impact of this effect is obviously more significant for a higher value of the quantization parameter QP, and therefore lower bit request reduction ratios are achieved for $QP = 20$ than for $QP = 10$. In order to refine the rate allocation for higher values of QP, the influence of the intra-coding noise should be incorporated in the decoder estimate of the variance. This is a matter for further investigation. Note, however, that in any case a certain amount of underestimation of the rate should be
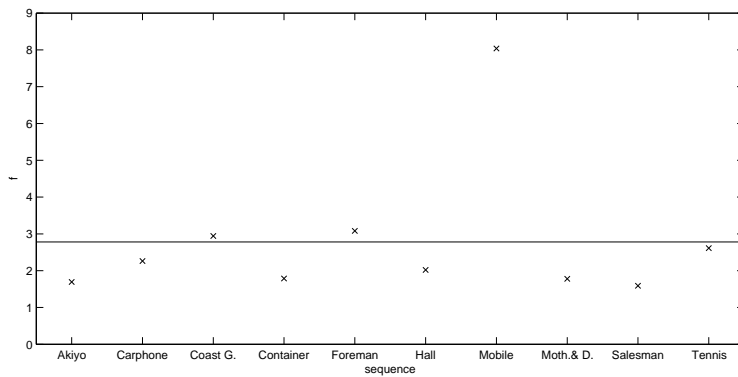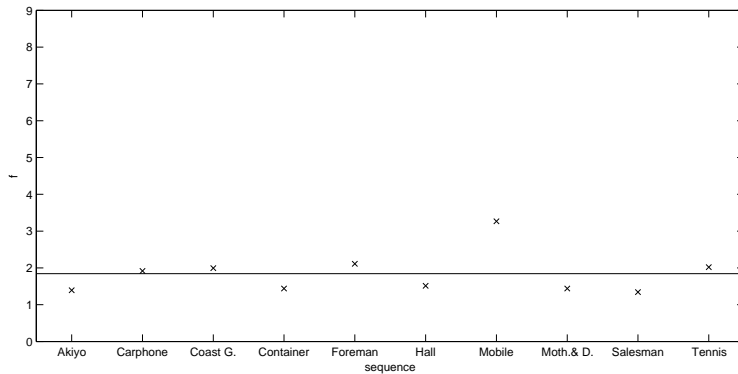
(a) $QP = 10$, $\mu = 2.78$, $\sigma = 1.92$

(b) $QP = 20$, $\mu = 1.84$, $\sigma = 0.58$

Fig. 6 Average bit request reduction ratio f for the ten test sequences. The key frames are intra-coded using H.263+ with (a) $QP = 10$ and (b) $QP = 20$. The mean value over all the sequences is denoted by μ and is indicated with the solid horizontal line. The standard deviation is denoted by σ.

maintained, since this assures an optimal rate-distortion performance (as explained in Section 4).

In general, however, we observe that significant bit request reduction ratios f are achieved. According to (10) and (11), the decoder complexity and the latency decrease by the same ratio f. The reduction of the latency is especially crucial when putting the discussed Wyner-Ziv video coding scheme into practice, since then the large delays of the feedback channel approach without RA are unacceptable. Note that as a counterpart our RA algorithm has made the encoder a bit more complex. More specifically, the main factors that influence the encoder burden are: the decoding of the key frames, the estimation of the correlation noise variance and the estimation of the bit rate for each bit plane. Nevertheless, the total computational load of the encoder is still small in comparison with the complexity of conventional encoders.

6 Conclusion

In this paper, we presented a RA algorithm to reduce the computational decoder complexity and latency of rate-compatible, turbo code-based PDWZ video coders with feedback channel. The algorithm estimates the appropriate number of bits for each frame without complicating the encoder, thereby considerably reducing the number of bit requests from the decoder over the feedback channel. Experimental results on several test sequences show that the decoder complexity and the latency are diminished by a significant factor, while a very near-to-optimal RD performance is preserved.

References

1. Aaron, A., Girod, B.: Compression with side information using turbo codes. In: IEEE Data Compression Conference, pp. 252–261. Snowbird, UT, USA (2002)
2. Aaron, A., Rane, S., Setton, E., Girod, B.: Transform-domain Wyner-Ziv codec for video. In: SPIE Visual Communications and Image Processing. San Jose, CA, USA (2004)
3. Aaron, A., Zhang, R., Girod, B.: Wyner-Ziv coding of motion video. In: Asilomar Conference on Signals, Systems and Computers, vol. 1, pp. 240–244. Pacific Grove, CA, USA (2002)
4. Ascenso, J., Brites, C., Pereira, F.: Improving frame interpolation with spatial motion smoothing for pixel domain distributed video coding. In: 5th EURASIP Conference. Smolenice, Slovak Republic (2005)
5. Ascenso, J., Brites, C., Pereira, F.: Motion compensated refinement for low complexity pixel distributed video coding. In: IEEE International Conference on Advanced Video and Signal Based Surveillance. Como, Italy (2005)
6. Belkoura, Z., Sikora, T.: Towards rate-decoder complexity optimisation in turbo-coder based distributed video coding. In: Picture Coding Symposium. Beijing, China (2006)
7. Belkoura, Z.M., Sikora, T.: Improving Wyner-Ziv video coding by block-based distortion estimation. In: European Signal Processing Conference (EUSIPCO). Florence, Italy (2006)
8. Brites, C., Ascenso, J., Pereira, F.: Feedback channel in pixel domain Wyner-Ziv video coding: myths and realities. In: European Signal Processing Conference (EUSIPCO). Florence, Italy (2006)
9. Brites, C., Ascenso, J., Pereira, F.: Improving transform domain Wyner-Ziv video coding performance. In: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), vol. 2, pp. II–525–II–528. Toulouse, France (2006)
10. Brites, C., Ascenso, J., Pereira, F.: Modeling correlation noise statistics at decoder for pixel based Wyner-Ziv video coding. In: Picture Coding Symposium. Beijing, China (2006)
11. Cheung, N.M., Ortega, A.: A model-based approach to correlation estimation in wavelet-based distributed source coding with application to hyperspectral imagery. In: IEEE International Conference on Image Processing (ICIP), pp. 613–616. Atlanta, USA (2006)
12. Cheung, N.M., Wang, H., Ortega, A.: Correlation estimation for distributed source coding under information exchange constraints. In: IEEE International Conference on Image Processing (ICIP), vol. 2, pp. II–682–II–685. Genova, Italy (2005)
13. Dalai, M., Leonardi, R., Pereira, F.: Improving turbo codec integration in pixel-domain distributed video coding. In: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), vol. 2, pp. II–537–II–540. Toulouse, France (2006)
14. Liu, L., Li, Z., Delp, E.J.: Backward channel aware Wyner-Ziv video coding. In: IEEE International Conference on Image Processing (ICIP), pp. 1677–1680. Atlanta, USA (2006)
15. Liveris, A., Xiong, Z., Georghiades, C.: Compression of binary sources with side information using low-density parity-check codes. In: Global Telecommunications Conference, vol. 2, pp. 1300–1304. Taipei, Taiwan (2002)
16. International Telecommunication Union: Video coding for low bit rate communication. ITU-T Recommendation H.263 (1998). Http://www.itu.int/rec/T-REC-H.263/
17. Morbee, M., Prades-Nebot, J., Pižurica, A., Philips, W.: Rate allocation algorithm for pixel-domain distributed video coding without feedback channel. In: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), vol. 1, pp. I–521–I–524. Honolulu, HI, USA (2007)

18. Puri, R., Ramchandran, K.: PRISM: A new robust video coding architecture based on distributed compression principles. In: Allerton Conference on Communication, Control, and Computing. Allerton, IL, USA (2002)
19. Roca, A., Morbee, M., Prades-Nebot, J., Delp, E.J.: A distortion control algorithm for pixel-domain Wyner-Ziv video coding. In: Picture Coding Symposium. Lisbon, Portugal (2007)
20. Rowitch, D., Milstein, L.: On the performance of hybrid FEC/ARQ systems using rate compatible punctured turbo codes. IEEE Transactions on Communications **48**(6), 948–959 (2000)
21. Slepian, J., Wolf, J.: Noiseless coding of correlated information sources. IEEE Transactions on Information Theory **19**(4), 471– 480 (1973)
22. Tagliasacchi, M., Trapanese, A., Tubaro, S., Ascenso, J., Brites, C., Pereira, F.: Intra mode decision based on spatio-temporal cues in pixel domain Wyner-Ziv video coding. In: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), vol. 2, pp. II–57–II–60. Toulouse, France (2006)
23. Trapanese, A., Tagliasacchi, M., Tubaro, S., Ascenso, J., Brites, C., Pereira, F.: Improved correlation noise statistics modeling in frame-based pixel domain Wyner-Ziv video coding. In: International workshop on very low bit rate video. Sardinia, Italy (2005)
24. Westerlaken, R.P., Borchert, S., Gunnewiek, R.K., Lagendijk, R.L.: Dependency channel modeling for a LDPC-based Wyner-Ziv video compression scheme. In: IEEE International Conference on Image Processing (ICIP), pp. 277–280. Atlanta, USA (2006)
25. Wyner, A., Ziv, J.: The rate-distortion function for source coding with side information at the decoder. IEEE Transactions on Information Theory **22**(1), 1–10 (1976)
26. Xu, Q., Xiong, Z.: Layered Wyner-Ziv video coding. In: SPIE Visual Communications and Image Processing: Special Session on Multimedia Technologies for Embedded Systems. San Jose, CA, USA (2004)
27. Xu, Q., Xiong, Z.: Layered Wyner-Ziv video coding. IEEE Transactions on Image Processing **15**(12), 3791–3803 (2006)

Manuscript details

Number of manuscript pages: 18
Number of figures: 6
Number of tables: 2

List of Figures

List of Tables