RATE ALLOCATION ALGORITHM FOR PIXEL-DOMAIN DISTRIBUTED VIDEO CODING WITHOUT FEEDBACK CHANNEL

Marleen Morbée[1], Josep Prades-Nebot[2], Aleksandra Pižurica[1] and Wilfried Philips[1]

[1] TELIN
Ghent University
Ghent, Belgium
mmorbee@telin.ugent.be

[2] GTS-ITEAM
Universidad Politécnica de Valencia
Valencia, Spain
jprades@dcom.upv.es

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. Distributed Video (DV) Coding is a new paradigm that aims to achieve this. To allocate a proper number of bits to each frame, most DV coding algorithms use a feedback channel (FBC). However, in some cases, a FBC does not exist. In this paper, we therefore propose a rate allocation (RA) algorithm for pixel-domain distributed video coders without FBC. Our algorithm estimates at the encoder the number of bits for every frame without significantly increasing the encoder complexity. Experimental results show that our RA algorithm delivers satisfactory estimates of the adequate encoding rate, especially for sequences with little motion.

Index Terms— Distributed Video Coding, Wyner-Ziv coding, rate allocation

1. INTRODUCTION

Some video applications, e.g., mobile video telephony, wireless video surveillance and disposable video cameras, require low-complexity coders. Distributed Video (DV) coding is a new paradigm that fulfills this requirement by performing intra-frame encoding and inter-frame decoding [1, 2]. As DV decoders perform motion estimation and motion compensated interpolation, most of the computational load is moved from the encoder to the decoder.

One of the most difficult tasks in DV coding is to allocate 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 DV coders solve this problem by using a feedback channel (FBC) which allows the decoder to request additional bits from the encoder when needed. Although the use of a FBC allows an accurate rate allocation (RA), it is not a valid solution in unidirectional and offline applications, and can introduce an excessive delay [3].

In this paper, we propose a RA algorithm for pixel-domain distributed video (PDDV) coders that do not use a FBC. Our algorithm computes the number of bits to encode each video frame without significantly increasing the encoder complexity. Experimental results show that the RA algorithm delivers satisfactory estimates of the rate, especially for sequences with little motion.

This work has been supported by the Spanish Ministry of Education and Science and the European Commission (FEDER) under grant TEC2005-07751-C02-01.

A. Pižurica is a postdoctoral research fellow of FWO, Flanders

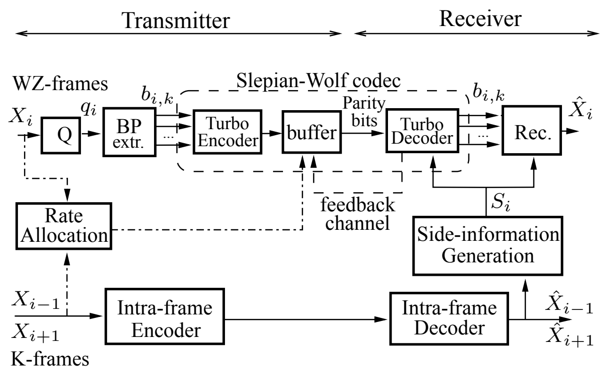

Fig. 1. General block diagram of a PDDV coder.

The paper is organized as follows. In Section 2, we study the basics of PDDV coding. In Section 3, we study the RA problem and the advantages and inconveniences of using a FBC. In Section 4, we describe our RA algorithm and, in Section 5, we compare the performance of a DV coder using a FBC and the performance of the same DV coder using our RA algorithm. Finally, the conclusions are presented in Section 6.

2. PIXEL-DOMAIN DV CODING

In DV coding, the frames are organized into key frames (K-frames) and Wyner-Ziv frames (WZ-frames). The K-frames are coded using a conventional intra-frame coder. The WZ-frames are coded using the Wyner-Ziv paradigm, i.e., they are intra-frame encoded, but they are conditionally decoded using side information (Figure 1). In most DV coders, the odd frames are encoded as K-frames and the even frames are encoded as WZ-frames [1, 2]. Coding and decoding is done unsequentially in such a way that, before decoding the i^{th} WZ-frame X_i, the preceding and succeeding K-frames (X_{i-1} and X_{i+1}) have already been transmitted and decoded. Thus, the receiver can obtain a good approximation S_i of X_i by interpolating its two closest decoded frames ($\hat{X}_{i-1}$ and $\hat{X}_{i+1}$). S_i constitutes the side information to conditionally decode X_i.

In a practical PDDV coder, the pixel values of X_i are first quantized with a uniform fixed-rate quantizer Q of 2^M levels. Subsequently, bit planes (BPs) are extracted from the quantization indices q_i. Then, the m most significant BPs $b_{i,k}$ ($1 \leq k \leq m$, $0 \leq m \leq M$) are independently encoded by a Slepian-Wolf (SW) coder [4]. The transmission and decoding of BPs is done in order of significance (the most significant BPs are transmitted and decoded first). The SW coding is implemented with efficient channel codes that yield the parity bits of $b_{i,k}$, which are transmitted. ¿From these parity bits and the corresponding BP $b'_{i,k}$ extracted from the side information, the SW decoder obtains $b_{i,k}$. Note that $b'_{i,k}$ can be considered the result of transmitting $b_{i,k}$ through a noisy *virtual channel*. The SW decoder is a channel decoder that recovers $b_{i,k}$ from its noisy version $b'_{i,k}$ and the received parity bits. Finally, the decoded BPs $b_{i,k}$ together with the side information S_i allow the decoder to reconstruct the signal $\hat{X}_i$ using $\hat{X}_i = E\{(X_i|S_i, b_{i,k})\}$.

3. THE RATE ALLOCATION PROBLEM

In this section, we first study the rate-distortion (RD) function for Gaussian sources when the decoder has access to side information about the source. Then, we discuss the RA problem for PDDV coders in particular.

Let X and U be two independent Gaussian random variables with standard deviations σ_X and σ_U respectively, and let $S = X + U$. Let us assume that the mean square error is used as a measure for the distortion D. Then, the RD function of X when both the encoder and the decoder know the side information S ($R_{X|S}(D)$) is equal to the RD function in the case that only the decoder knows the side information ($R_S(D)$) [5]:

$$R_{X|S}(D) = R_S(D) = \begin{cases} \frac{1}{2}\log_2 \dfrac{\sigma_X^2\sigma_U^2}{(\sigma_X^2+\sigma_U^2)D}, & 0 < D \leq \frac{\sigma_X^2\sigma_U^2}{\sigma_X^2+\sigma_U^2} \\ 0, & D > \frac{\sigma_X^2\sigma_U^2}{\sigma_X^2+\sigma_U^2} \end{cases} \quad (1)$$

Consequently, there is no rate loss for the case of Gaussian random variables for an encoder that does not have access to S.

In the following, we are interested in encodings where $D \leq \sigma_X^2\sigma_U^2/(\sigma_X^2+\sigma_U^2)$. For a given distortion D, an encoder can obtain the optimum rate from (1) if the value σ_U^2 is known. In those applications where the encoder does not have access to S, an estimate $\hat{\sigma}_U^2$ will be used and the allocated rate $\hat{R}_S$ will be

$$\hat{R}_S(D) = \frac{1}{2}\log_2 \frac{\sigma_X^2\hat{\sigma}_U^2}{(\sigma_X^2+\hat{\sigma}_U^2)D}. \quad (2)$$

From (1) and (2), we derive that the difference in rate between the encoder using $\hat{\sigma}_U^2$ and the encoder using σ_U^2 is

$$\Delta R = \hat{R}_S(D) - R_S(D) = \frac{1}{2}\log_2 \frac{1+\sigma_X^2/\sigma_U^2}{1+\sigma_X^2/\hat{\sigma}_U^2}. \quad (3)$$

On the other hand, if we define the variation in distortion when we encode at $\hat{R}_S$ instead of at R_S as

$$\Delta D \triangleq 10\log_{10} D(\hat{R}_S)/D(R_S), \quad (4)$$

we derive $\Delta D \backsimeq -6\Delta R$. Note that if $\hat{\sigma}_U^2 < \sigma_U^2$, then $\Delta D > 0$ and $\Delta R < 0$, i.e., we reduce the rate at the expense of proportionally increasing the distortion. In a similar way, if $\hat{\sigma}_U^2 > \sigma_U^2$, then $\Delta D < 0$ and $\Delta R > 0$. Therefore, errors in estimating σ_U^2 introduce coding efficiency losses in rate- or distortion-constrained coders.

In practical PDDV coders, once the quantizer has been chosen, the optimum rate R^* is the *minimum* rate necessary to (nearly) losslessly decode the BPs $b_{i,k}$ knowing S_i at the decoder. The use of a rate higher than R^* does not involve a proportional reduction in distortion, but only 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 $b_{i,k}$, which can greatly increase the distortion. This is because of the threshold effect of the channel codes used in DV coders. Consequently, σ_U^2 should be accurately estimated so that a rate that guarantees lossless (or nearly lossless) encoding of the BPs $b_{i,k}$ is allocated. In practice, an accurate estimation of σ_U^2 would imply a considerable increase in the complexity of the encoder. Moreover, because of the channel coding techniques used in DV coders (mostly turbo codes or LDPC codes) it is difficult to find out exactly what the minimum rate is to achieve an almost lossless encoding of a particular signal.

A common RA solution adopted in DV coders is the use of a FBC and a rate-compatible punctured turbo code (RCPTC) [6]. In this configuration, all the parity bits generated by the turbo encoder are saved in a buffer (Figure 1) and divided into parity bit sets. To determine the adequate number of parity bit sets to be sent, one first transmits one parity bit set, and if the decoder detects that the residual error probability Q is above a threshold t, it requests an additional parity bit set from the buffer through the FBC. This transmission-request process is repeated until $Q < t$.

However, although the FBC allows to achieve an optimal RD performance, this FBC cannot be implemented in offline applications or in those applications where communication from the decoder to the encoder is not possible. In those applications, an appropriate RA algorithm at the encoder can take over its role. In the following section, we will describe this RA algorithm to suppress the FBC in more detail.

4. THE PROPOSED RATE ALLOCATION ALGORITHM

The main idea of the proposed method is to allocate a proper number of bits to each BP of a WZ-frame of the video sequence. Let U be a random variable representing the difference between pixel values of the original frame X_i and the corresponding pixel values of its side information frame S_i. In [1,2,7,8], U is assumed to follow a Laplacian distribution $f_U(u) = \alpha/2\exp(-\alpha|u|)$ where $\alpha = \sqrt{2}/\sigma$. In practice, however, pixels can only take integer values in the interval $[0, 255]$, so U is a discrete random variable that can only take integer values u in $[-255, 255]$. Hence, we derive the probability mass function (p.m.f.) for each value u as follows

$$P(U = u) = \int_{u-0.5}^{u+0.5} f_U(z)\,dz \quad (5)$$

except for $u = -255$ and $u = 255$ where the integration intervals of (5) are $(-\infty, -254.5)$ and $(254.5, \infty)$, respectively. The resulting p.m.f. is then

$$P(U = u) = \begin{cases} 1 - e^{-\frac{1}{\sqrt{2}\sigma}}, & u = 0 \\ \sinh(\frac{1}{\sqrt{2}\sigma})e^{-\frac{\sqrt{2}|u|}{\sigma}}, & 1 \leq |u| \leq 254 \\ \frac{1}{2}e^{-254.5\frac{\sqrt{2}}{\sigma}}, & |u| = 255 \end{cases} \quad (6)$$

As every BP of X_i is separately encoded, a different number of bits B_k must be allocated to each BP $b_{i,k}$. 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 (BSC). Consequently, to obtain B_k, we need to know the bit error probability P_k of each BP $b_{i,k}$.

Our algorithm first obtains an estimate $\hat{\sigma}^2$ of parameter σ^2 at the encoder (Section 4.1). Then, for each BP $b_{i,k}$, we use $\hat{\sigma}$ to estimate P_k (Section 4.2). Once P_k is estimated, we can determine the number of bits and the corresponding rate for each BP by taking into account the error correcting capacity of the turbo code and the frame rate of the video (Section 4.3).

Once the parity bits have been decoded, the residual error probability Q_k is estimated at the decoder ($\hat{Q}_k$) (Section 4.4). If $\hat{Q}_k$ is above a threshold t_k, the parity bits are discarded and the decoded frame is set equal to the side information ($\hat{X}_i = S_i$). This way, we prevent an increase in the distortion caused by an excessive number of errors in the decoded BPs. In the following, we explain each step of our RA algorithm in a more detailed way.

4.1. Estimation of σ^2

At the encoder, we obtain an estimation of σ^2 for each frame. The estimation should be very simple to avoid increasing the encoder complexity significantly. In our algorithm, $\hat{\sigma}^2$ is the mean square error between the current WZ-frame X_i and the average of its two closest K-frames (X_{i-1} and X_{i+1}):

$$\hat{\sigma}^2 = \frac{1}{N} \sum_{(v,w)\in X_i} \Big(X_i(v,w) - \frac{X_{i-1}(v,w) + X_{i+1}(v,w)}{2}\Big)^2 \quad (7)$$

where N is the number of pixels in each frame. In general, the resulting $\hat{\sigma}^2$ is an overestimate of the real σ^2 since it is expected that the motion compensated interpolation performed at the decoder to obtain the side information will be more accurate than the simple averaging of the two closest K-frames. The implications of this overestimation will be discussed in Section 5.

4.2. Estimation of the error probabilities $\{P_k\}$

Let $x_{i,k}$ and $s_{i,k}$ denote a bit in the k^{th} BP of the original frame X_i and the side information S_i, respectively. The error probability for the corresponding BP is:

$$P_k = P(x_{i,k} = 1, s_{i,k} = 0) + P(x_{i,k} = 0, s_{i,k} = 1) \quad (8)$$

Taking into account the symmetry of the error distribution,

$$\begin{aligned} P_k &= 2P(x_{i,k} = 1, s_{i,k} = 0) \\ &= 2 \sum_{u=-255}^{255} P(x_{i,k} = 1, s_{i,k} = 0 | U = u) P(U = u). \end{aligned} \quad (9)$$

By assuming X_i is uniformly distributed in $[0, 255]$, we obtain after some calculations:

$$P(x_{i,k} = 1, s_{i,k} = 0 | U = u) = \begin{cases} \dfrac{\left(2^{k-1} - \lfloor \frac{d-1}{2} \rfloor\right)\left(2^{9-k} \lfloor \frac{d}{2} \rfloor - u\right)(-1)^d}{256 - u}, u > 0 \\ \dfrac{\left(2^{k-1} - \lceil \frac{d}{2} \rceil\right)\left(2^{9-k} \lfloor \frac{d}{2} \rfloor + u\right)(-1)^d}{256 + u}, \quad u < 0 \end{cases} \quad (10)$$

where $d = \lceil \frac{|u|}{2^{8-k}} \rceil$. By using the variance estimate (7) together with (6), (9) and (10), the encoder obtains an estimate of P_k. Note that in the computation of P_k, we assumed that for the decoding of $b_{i,k}$, the decoder does not take into account the information provided by the $k-1$ previously decoded BPs.

4.3. Estimation of the number of bits $\{B_k\}$ and the rates $\{R_k\}$

Once P_k is estimated, we choose the adequate number of parity bit sets that enables us to decode with a residual error probability Q_k below a threshold t_k ($Q_k < t_k$). To do that, we estimated functions that provide the residual error probability Q_k as a function of P_k and the number of parity bit sets. These functions are obtained by averaging simulations over a set of input random sequences with different error probabilities. Using these experimental functions and knowing the estimate of P_k and the threshold t_k, we estimate the adequate number of bits B_k. The corresponding encoding rate is obtained using $R_k = f B_k$, where f is the frame rate.

4.4. Estimation of the residual error probabilities $\{Q_k\}$

If the rate allocated to encode a BP was too low, the decoded BP can contain such a large number of errors that the quality of the reconstructed frame is worse than the quality of the side information. To prevent this situation, we need to know the residual error probability Q_k of each BP at the decoder. So far, most PDDV coders assume that Q_k can be perfectly estimated at the decoder [1–3]. In practice, this is not possible since X_i is only available at the encoder. We estimate Q_k as [9]

$$\hat{Q}_k = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{1 + e^{|L_n|}} \quad (11)$$

where N is the number of pixels in each frame and L_n the log-likelihood ratio of the n^{th} bit in the considered BP $b_{i,k}$. If $\hat{Q}_k$ is above a certain threshold ($\hat{Q}_k > t_k$), the decoded BPs are discarded and the side information is used as reconstructed frame. The impact of the bit errors depends on the BP: the more significant the BP, the larger its impact on the distortion. Therefore, the threshold t_k is set differently depending on k, as we will explain in Section 5.

5. EXPERIMENTAL RESULTS

In this section, we experimentally study the accuracy of our RA algorithm when it is used in a PDDV coder without FBC and compare it with the rate allocations provided by the same coder using a FBC.

The DV 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 RCPTC; the other BPs are discarded. In our experiments, m is chosen to be 2. 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 is set to 32 which allows our RA algorithm to allocate parity bit multiples of 792 bits to each BP. Side information is generated at the decoder by using the interpolation tools described in [2]. The K-frames are losslessly transmitted. Note that in the case of lossy K-frames the noise variance estimate (7) should be adapted to take into account the coding noise of the K-frames and its influence on the motion compensated interpolation accuracy. This is however beyond the scope of this paper.

We encoded several test QCIF sequences (176×144 $\frac{\text{pixels}}{\text{frame}}$, 30 $\frac{\text{frames}}{\text{s}}$) with two RA strategies: our RA algorithm and the allocations provided by the coder using a FBC. The thresholds t_k for Q_k (FBC) and for $\hat{Q}_k$ (our RA approach) are set to $\frac{1}{N}$ for the first BP (t_1) and $\frac{2}{N}$ for the second BP (t_2), where N is the number of pixels in each frame.

Table 1 shows the difference between the RA (in kb/s) provided by our algorithm and the RA using the FBC when encoding the first BP of each frame. More specifically, the percentage of frames with a difference in rate of ΔR kb/s is shown. Note that the ideal rate is allocated in between 23% and 60% of the frames. In many frames, an overestimation of the rate is observed. This is especially due to the fact that $\hat{\sigma}^2$ is too high (as explained in Section 4.1), which causes

Video sequence	% of frames with ΔR				
	≤-24 kb/s	-12 kb/s	0 kb/s	+12 kb/s	≥+24 kb/s
Akiyo	12.1	14.7	59.7	10.1	3.4
Carphone	7.4	10.1	23.5	34.9	24.2
Foreman	7.5	17.6	23.1	13.6	38.2
Salesman	8.0	10.1	45.0	26.8	10.1

Table 1. Percentage of frames that differ by ΔR from the rate of the FBC (for the first BP).

Video sequence	% of frames with ΔR				
	≤-24 kb/s	-12 kb/s	0 kb/s	+12 kb/s	≥+24 kb/s
Akiyo	0.7	8.0	31.5	28.9	30.9
Carphone	0	2.0	7.4	16.1	74.5
Foreman	0	1.5	12.6	10.5	75.4
Salesman	0	10.1	31.5	18.1	40.3

Table 2. Percentage of frames that differ by ΔR from the rate of the FBC (for the second BP).

an overestimation of the corresponding P_k (see Section 4.2) and R_k (see Section 4.3). In sequences with little motion (*Salesman*, *Akiyo*), we allocate a more appropriate rate since the estimate $\hat{\sigma}^2$ is more accurate in this case. Table 2 shows the difference between the RA (in kb/s) provided by our algorithm and the RA using the FBC when encoding the second BP of each frame. Here the rate allocations are further away from the ideal rate. This is logical since we can derive from (6), (9) and (10) that for a certain σ^2, $P_2 \approx 3P_1$, so that an inaccuracy in $\hat{\sigma}^2$ will have a three times larger influence on the RA of the second BP than on the RA of the first BP. Table 3 shows the average PSNR after turbo decoding and reconstructing the first and the second BP for both our RA algorithm and the FBC case. We observe that the gain in quality with respect to the side information is smaller when we use our RA algorithm than with the FBC approach. The loss in quality is for the first BP between 21% (*Foreman*) and 60% (*Akiyo*) of the gain in quality that one can optimally achieve, and between 18% (*Foreman*) and 68% (*Akiyo*) for the second BP. These relative losses are especially large for Akiyo, since in this sequence the quality of the side information is so high that the impact on the distortion of erroneously decoded WZ-bits (which can occur due to an inaccuracy in the estimation of $\hat{Q}_k$) is very large. In those sequences where the side information is of a lower quality (due to the presence of more motion), the impact of an erroneously decoded WZ-bit is much smaller or even negligible. Table 4 shows the average rate used for the encoding of the first and the second BP of the video sequences, comparing our RA with the FBC approach. For the first BP, the mean rates for both approaches are quite close, with differences between 2 kb/s and 19 kb/s. For the second BP, the differences are larger (as explained earlier this section), namely between 13 kb/s and 60 kb/s.

6. CONCLUSION

In this paper, we presented a RA algorithm for rate-compatible, turbo code-based PDDV coders. Without complicating the encoder, the algorithm estimates the appropriate number of bits for each frame. Using our algorithm, the FBC can be removed from the traditional scheme, but in general at the expense of a certain loss in RD performance.

Video sequence	PSNR (dB)				
	Side inform.	FBC		our RA	
		BP 1	BP 2	BP 1	BP 2
Akiyo	49.85	50.10	50.29	49.95	49.99
Carphone	32.95	33.58	34.39	33.43	34.10
Foreman	36.23	36.76	37.27	36.65	37.08
Salesman	43.90	44.14	44.48	44.05	44.35

Table 3. Average PSNR after turbo decoding and reconstructing the 1^{st} and the 2^{nd} BP for the FBC and for our RA algorithm. Shown is also the PSNR of the side information.

Video sequence	Rate (kb/s)			
	FBC		our RA	
	BP 1	BP 2	BP 1	BP 2
Akiyo	30	31	26	44
Carphone	80	138	88	178
Foreman	72	111	91	171
Salesman	33	62	35	75

Table 4. Average rate used for encoding the 1^{st} and the 2^{nd} BP using the FBC and our RA algorithm.

7. REFERENCES

[1] A. Aaron, R. Zhang, and B. Girod, "Wyner-Ziv coding of motion video," in *Proc. Asilomar Conference on Signals and Systems*, Pacific Grove, California, USA, November 2002.

[2] J. Ascenso, C. Brites, and F. Pereira, "Improving frame interpolation with spatial motion smoothing for pixel domain distributed video coding," in *5th EURASIP*, Slovack, Republic, June 2005.

[3] C. Brites, J. Ascenso, and F. Pereira, "Feedback channel in pixel domain Wyner-Ziv video coding: myths and realities," in *14th EUSIPCO*, Florence, Italy, September 2006.

[4] J. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inform. Theory*, vol. 19, no. 4, July 1973.

[5] 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, Jan. 1976.

[6] D. Rowitch and L. Milstein, "On the performance of hybrid FEC/ARQ systems using rate compatible punctured turbo codes," *IEEE Trans. Commun.*, vol. 48, no. 6, pp. 948–959, June 2000.

[7] N-M. Cheung and A. Ortega, "A model-based approach to correlation estimation in wavelet-based distributed source coding with application to hyperspectral imagery," in *ICIP*, Atlanta, USA, October 2006.

[8] M. Grangetto, E. Magli, and G. Olmo, "Near-lossless distributed coding of hyperspectral images using a statistical model for probability estimation," in *14th EUSIPCO*, Florence, Italy, September 2006.

[9] P. Hoeher, I. Land, and U. Sorger, "Log-likelihood values and monte carlo simulation– some fundamental results," in *Int. Symp. on Turbo Codes and Rel. Topics*, Sept. 2000, pp. 43–46.