Feedback Channel Suppression In Pixel-Domain Distributed Video Coding

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*— Distributed Video (DV) Coding is a new coding paradigm, in which the video modelling task is moved, partially or totally, to the decoder side. To allocate a number of bits to each frame, most DV coding algorithms use a feedback channel (FBC). In this paper, we propose a rate allocation algorithm for pixel-domain distributed video 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, if we compare our scheme with a FBC scheme, the rate allocations and qualities provided by our algorithm are satisfactory.**

I. Introduction

In conventional motion-compensated video coders, motion estimation is performed at the encoder in order to exploit the temporal redundancy existent in the video frames [1]. Due to the large complexity of motion estimation algorithms, motion-compensated encoders are much more complicated than their correspondent decoders. Consequently, this coding strategy is appropriate for applications where video is encoded once but decoded many times, as occurs in broadcasting or video-on-demand. However, 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 [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. Moreover, the frame quality provided by our algorithm is close to the one provided by a FBC-based algorithm.

The paper is organized as follows. In Section II, we study the basics of PDDV coding. In Section III, we study the problem of rate allocation in DV coding and the feedback channel solution to this problem. In Section IV, we describe our RA algorithm and in Section V we compare the performance of a DV coder using feedback channel and using our algorithm. Finally, the conclusions are presented in Section VI.

II. 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 [4], [5]. Coding and decoding is done out of order in such a way that, before decoding the $i-\mathrm{th}$ 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 $b_{i,k}$ $(0 \leq m \leq M,\ 1 \leq k \leq m)$ are independently encoded by a Slepian-Wolf (SW) coder [6]. The transmission and decoding of BPs is done in order (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 $\hat{X}_i = E\{(X_i|S_i, b_{i,k})\}$.

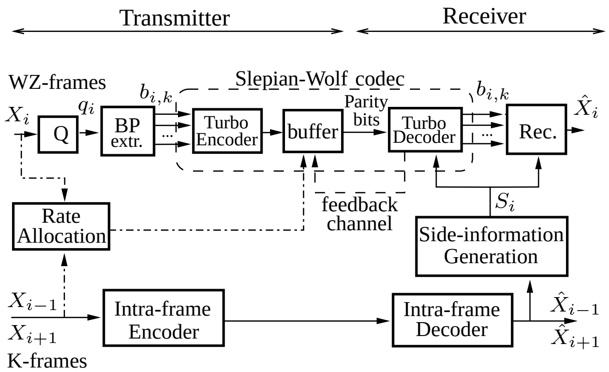

Fig. 1. General block diagram of a PDDV coder.

III. THE RATE ALLOCATION PROBLEM IN DV CODING

In practical PDDV coders, once the quantizer has been chosen, the optimum rate R^* is the *minimum* rate necessary to (nearly) losslessly transmit the BPs $b_{i,k}$ knowing S_i at the decoder. The use of a rate higher than R^* does not involve a 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 Bit Error Rate step nature of the channel codes used in DV coders.

Rate allocation at the encoder side of a DV coder is an ambiguous problem. Indeed, on the one hand, if we define U the difference between the original image X and the side information S, *i.e.* $U = S - X$, a good RA would involve an accurate modelling of the signal U. On the other hand, however, exploring the statistics model of U would imply a considerable increase of the complexity of the encoder, which is against the starting idea of DV coding. Additionally, because of the channel coding techniques used in DV coding, 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 sequence.

A common RA solution adopted in DV coding is to perform rate control at the decoder by making use of a feedback channel together with a rate-compatible turbo code (RCTP) [7]. In this configuration, all the parity bits generated by the turbo encoder are saved in a buffer (Figure 1). The puncturing matrix of the turbo code defines which and how many of these parity bits are sent. To determine the adequate puncturing matrix, one first transmits a set of parity bits, corresponding to a minimum number of bits, and if the decoder detects that the residual error probability Q is above a threshold t, the decoder requests more bits from the encoder through the FBC. The transmission-request process is repeated until $Q < t$.

With this approach, one can obtain a rate very close to R^*. However, in some applications, a FBC does not exist. For those cases, the traditional DV coding scheme fails and the RA problem need to be solved at the encoder in a computationally simple way. In the next section, we describe our solution to this RA problem, which is the key to feedback channel suppression.

IV. FEEDBACK CHANNEL SUPPRESSION

The main idea of this paper is to remove the FBC from the traditional DV coding scheme. Therefore, we formulate a rate allocation algorithm that estimates at the encoder the number of parity bits to be sent to correct properly the side information. Moreover, this algorithm is used at the decoder to estimate parameters for the correct functioning of the reconstruction function and the turbo decoding.

Let U be a random variable representing the difference between pixel values of the original frame X and the corresponding pixel values of its side information frame S. In [4] and [5], 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

$$p(U = u) = \int_{u-0.5}^{u+0.5} f_U(z)\, dz \tag{1}$$

except for $u = -255$ and $u = 255$ where the integration intervals of (1) are $(-\infty, -254.5)$ and $(254.5, \infty)$, respectively. The resulting p.m.f. is then

$$p(U = u) = \begin{cases} 1 - e^{-\alpha/2}, & u = 0 \\ \sinh(\alpha/2) e^{-\alpha|u|}, & 1 \le |u| \le 254 \\ \frac{1}{2} e^{-254.5\alpha}, & |u| = 255 \end{cases} \tag{2}$$

Then, at the encoder, as every BP of X is separately encoded, a different number of bits B_k must be allocated to each BP 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 k.

At the encoder, the rate allocation algorithm first estimates the parameter σ^2 ($\hat{\sigma_e}^2$) (Section IV-A). Then, for each bit plane k, we use $\hat{\sigma_e}$ to estimate P_k ($\hat{P}_k^e$) (Section IV-B). Once $\hat{P}_k^e$ is estimated, we can determine the rate of each BP by taking into account the error correcting capacity of the turbo code and the frame rate of the video (Section IV-C).

At the decoder side, σ^2 is needed for the reconstruction and an estimate of P_k is used by the turbo decoder. To obtain estimates of these parameters (respectively $\hat{\sigma}_d^2$ and $\hat{P}_k^d$), we follow similar steps as at the encoder (Section IV-A and Section IV-B). The estimate of σ^2 at the decoder ($\hat{\sigma}_d^2$) will be more accurate than $\hat{\sigma}_e^2$, since it is based on the block matching performed during the motion compensated interpolation for the generation of the side information (Section IV-A).

In the following, we explain each step of our RA algorithm in a more detailed way, explaining its use at both encoder and decoder side.

A. Estimation of σ^2

In our RA solution, both encoder and decoder obtain an estimation of σ^2 for each frame.

At the encoder, however, estimation should be very simple to avoid increasing the encoder complexity significantly. $\hat{\sigma}_e^2$ is the mean square error between the actual frame and the average of its two closest K-frames. In general, the resulting $\hat{\sigma}_e^2$ will be an overestimated value of σ^2 since the motion compensated interpolation performed at the decoder will be more accurate than the simple averaging of the two closest K-frames. The implications of this overestimation will be discussed in Section V.

At the decoder, motion compensated interpolation is performed on a block-basis in order to generate the side information. During the interpolation process of a block of the $i-$th frame, best matching blocks in $\hat{X}_{i-1}$ and $\hat{X}_{i+1}$ are searched using a minimum mean square error criterion. Assuming linear motion between $\hat{X}_{i-1}$ and $\hat{X}_{i+1}$, the pixel values of both frames contribute with a 1/2 weight to the interpolated pixels that constitute the frame I_i. Then, the estimate of the variance between the original frame and the side information is [8]:

$$\hat{\sigma}_d^2 = \frac{1}{4}\frac{1}{N}\sum_{(v,w)\in I_i}(\hat{X}_{i-1}(v+dv_{i-1}, w+dw_{i-1}) - \hat{X}_{i+1}(v+dv_{i+1}, w+dw_{i+1}))^2 \quad (3)$$

where (dv_{i-1}, dw_{i-1}) and (dv_{i+1}, dw_{i+1}) represent the motion vectors with respect to I_i of $\hat{X}_{i-1}$ and $\hat{X}_{i+1}$, respectively. Note that frame I_i is different from the side information S_i, since in the used DV coder we will perform additional operations to obtain the final S_i (see [5] for a more detailed explanation of the generation of the side information).

B. Estimation of $\{P_k\}$

Let X_k and Y_k denote the transmitted and the received bit in the k-th bit plane, respectively. The total error probability for the corresponding bit plane is:

$$P_k = P(X_k = 1, Y_k = 0) + P(X_k = 0, Y_k = 1) \quad (4)$$

Taking into account the symmetry of the error distribution,

$$\begin{aligned} P_k &= 2\ P(X_k = 1, Y_k = 0) & (5)\\ &= 2\sum_{u=-255}^{255} P(X_k = 1, Y_k = 0|U = u)p(U = u). & (6)\end{aligned}$$

After some calculations we obtain

$$P(X_k = 1, Y_k = 0|U = u) = \begin{cases} \frac{\left(2^{k-1}-\lfloor\frac{d-1}{2}\rfloor\right)\left(2^{9-k}\lfloor\frac{d}{2}\rfloor-u\right)(-1)^d}{256-u}, & u > 0 \\ \frac{\left(2^{k-1}-\lceil\frac{d}{2}\rceil\right)\left(2^{9-k}\lfloor\frac{d}{2}\rfloor+u\right)(-1)^d}{256+u}, & u < 0 \end{cases} \quad (7)$$

where $d = \lceil\frac{|u|}{2^{8-k}}\rceil$. By using (2), (6) and (7) together with the variance estimate $\hat{\sigma}_e^2$, we obtain an estimate of P_k at the encoder ($\hat{P}_k^e$). The same way, but using $\hat{\sigma}_d^2$ instead of $\hat{\sigma}_e^2$, we estimate P_k at the decoder ($\hat{P}_k^d$). $\hat{P}_k^d$ will be more precise than $\hat{P}_k^e$, since $\hat{\sigma}_e^2$ is estimated with more accuracy than $\hat{\sigma}_d^2$.

C. Estimation of $\{R_k\}$

At the encoder, once P_k has been estimated, we have to choose the adequate turbo code puncturing matrix that allows to decode with a residual error probability Q below a threshold t ($Q < t$). To do that, we first need to obtain the functions providing the residual error probability Q as a function of the bit error probability P_k and the puncturing matrix. This functions are obtained by averaging simulations over a set of input random sequences with different error probabilities. The chosen matrix determines the number of bits B_k and the corresponding encoding rate R_k is obtained using $R_k = fB_k$, where f is the frame rate.

V. Experimental Results and discussion

In this section, we experimentally study the accuracy of a PDDV coder without FBC that uses our RA algorithm and compare it with the same PDDV coder that uses 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 RCTP; 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 $(1, 33/23)$ 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. The K-frames are losslessly transmitted. Side information is generated at the decoder by using the interpolation tools described in [5].

Video sequence	% of frames with ΔR				
	≤-48 kb/s	-24 kb/s	0 kb/s	+24 kb/s	≥+48 kb/s
Foreman	5.0	18.1	24.1	14.6	38.2
Carphone	6.7	8.7	26.2	34.2	24.2
Akiyo	1.3	7.4	67.1	20.8	3.4
Salesman	0	4.7	48.3	35.6	11.4

TABLE I

Percentage of frames that differ ΔR from the rate of the FBC (for the first bit plane).

We encoded several test QCIF sequences (176×144 pixels/frame, 30 frames/s) with two RA strategies: our RA algorithm and the allocations provided by the coder using a FBC. The threshold t for Q is set to 0.001, *i.e.* the transmission-request process in the FBC-coder is repeated until $Q < t$.

Table I 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

Video sequence	% of frames with ΔR				
	$\leq$-48 kb/s	-24 kb/s	0 kb/s	+24 kb/s	$\geq$+48 kb/s
Foreman	0	2.0	14.1	9.1	74.8
Carphone	0	2.0	7.4	18.1	72.5
Salesman	0	10.1	32.9	16.8	40.2
Akiyo	0	2.0	30.2	35.6	32.2

TABLE II

PERCENTAGE OF FRAMES THAT DIFFER ΔR FROM THE RATE OF THE FBC (FOR THE SECOND BIT PLANE).

Video sequence	PSNR (dB) FBC		PSNR (dB) our RA algorithm	
	BP 1	BP 2	BP 1	BP 2
Foreman	36.75	37.25	36.65	37.07
Carphone	33.57	34.34	33.40	34.07
Salesman	44.09	44.36	44.10	44.37
Akiyo	50.04	50.15	50.03	50.19

TABLE III

AVERAGE PSNR AFTER TURBO DECODING AND RECONSTRUCTING THE FIRST AND THE SECOND BP FOR THE FBC AND FOR OUR RA ALGORITHM.

of frames with a difference in rate of ΔR kb/s is shown. Note that the ideal rate is allocated in between 24% and 67% of the frames. In many frames, an overestimation of the rate is observed. This is especially due to the fact that $\hat{\sigma}_e^2$ is too high (as explained in Section IV-A), which causes an overestimation of the corresponding $\hat{P}_k^e$ (see Section IV-B) and R_k (see Section IV-C). In sequences with little motion (Salesman, Akiyo), we allocate a more appropriate rate since the estimate $\hat{\sigma}_e^2$ is more accurate in this case. Table II 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 (2), (6) and (7) that for a certain σ^2, $P_2 \approx 3P_1$, so that an inaccuracy in $\hat{\sigma}_e^2$ will have a three times larger influence on the RA of the second BP than on the first.

Table III shows the average PSNR after turbo decoding and reconstructing the first and the second BP with both RA algorithms. Note that for both algorithms, in those exceptional frames where the quality after turbo decoding and reconstructing with the WZ-bits is worse than the quality of the side information, it is assumed the decoded frame is equal to the side information frame, to avoid that the mean PSNR is affected by this issue. The qualities of the frames using our RA algorithm are a little worse but still quite similar to the ones obtained by the FBC approach. Table IV shows the average rate used for the encoding of the first and the second BP of the video sequences, comparing both RA algorithms. For the first BP, the mean rates for both approaches are quite close,

Video sequence	RATE (kb/s) FBC		RATE (kb/s) our RA algorithm	
	BP 1	BP 2	BP 1	BP 2
Foreman	71	112	91	171
Carphone	80	138	88	178
Salesman	28	63	35	75
Akiyo	24	29	26	44

TABLE IV

AVERAGE RATE USED FOR ENCODING THE FIRST AND THE SECOND BIT PLANE USING THE FBC AND OUR RA ALGORITHM.

with differences between 2 kb/s and 20 kb/s. For the second BP, the differences are larger (as explained before), namely between 12 kb/s and 59 kb/s.

VI. CONCLUSION

In this paper, we have presented a RA algorithm, that enables us to suppress the FBC from the traditional PDDV coding scheme. Without complicating the encoder, the algorithm estimates at the encoder the appropriate number of bits for each WZ-frame. Comparing the results of our RA algorithm and the FBC scheme, we observe that the loss in rate and quality due to the suppression of the FBC is acceptable and hence, our algorithm can be very useful in those applications where a FBC does not exist.

ACKNOWLEDGMENT

This work has been supported by a grant for the Secretaría de Estado de Educación y Universidades by the program CICYT TIC-2002-02469.

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

REFERENCES

[1] B. G. Haskell, A. Puri, and A. N. Netravali, *Digital Video: An introduction to MPEG-2*. Chapman and Hall, 1997.

[2] R. Puri and K. Ramchandran, “PRISM: A new robust video coding architecture based on distributed compression principles,” in *Proceedings Allerton Conference on Communication, Control, and Computing*, IL, USA, Oct. 2002.

[3] C. Brites, J. Ascenso, and F. Pereira, “Feedback channel in pixel domain Wyner-Ziv video coding: myths and realities,” in *14th European Signal Processing Conference (EUSIPCO’06)*, Florence, Italy, September 2006.

[4] A. Aaron, R. Zhang, and B. Girod, “Wyner-Ziv video coding with hash-based motion compensation at the receiver,” in *Proc. Asilomar Conference on Signals and Systems*, Pacific Grove, California, USA, November 2002.

[5] J. Ascenso, C. Brites, and F. Pereira, “Improving frame interpolation with spatial motion smoothing for pixel domain distributed video coding,” in *5th EURASIP Conference on Speech and Image Processing, Multimedia Communications and Services*, Slovack, Republic, July 2005.

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

[7] 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.

[8] C. Brites, J. Ascenso, and F. Pereira, “Modeling correlation noise statistics at decoder for pixel based Wyner-Ziv video coding,” in *Picture Coding Symposium (PCS)*, Beijing, China, April 2006.