ROBUST DECODING OF A 3D-ESCOT BITSTREAM TRANSMITTED OVER A NOISY CHANNEL

Manel Abid, Michel Kieffer, Marco Cagnazzo, Beatrice Pesquet-Popescu

ROBUST DECODING OF A 3D-ESCOT BITSTREAM TRANSMITTED OVER A NOISY CHANNEL

M. Abid[1], *M. Kieffer*[1,2], *M. Cagnazzo*[1], *and B. Pesquet-Popescu*[1]

[1] Telecom ParisTech, Signal and Image Processing Department,
46 rue Barrault, 75634 Paris cedex 13, France
[2] on leave from L2S - CNRS - SUPELEC - Univ Paris-Sud

ABSTRACT

In this paper, we propose a joint source-channel (JSC) decoding scheme for 3D ESCOT-based video coders, such as Vidwav. The embedded bitstream generated by such coders is very sensitive to transmission errors unavoidable on wireless channels. The proposed JSC decoder employs the residual redundancy left in the bitstream by the source coder combined with bit reliability information provided by the channel or channel decoder to correct transmission errors. When considering an AWGN channel, the performance gains are in average 4 dB in terms of PSNR of the reconstructed frames, and 0.7 dB in terms of channel SNR. When considering individual frames, the obtained gain is up to 15 dB in PSNR.

1. INTRODUCTION

Video transmission over heterogeneous (wired and wireless) networks has become widespread. Video compression systems are thus asked not only to be efficient, but also to be robust against transmission errors, which are unavoidable when considering radio-mobile links. Recently developped video coders, such as H.264/SVC [1] or Vidwav [2] allow a high degree of compression efficiency and scalability. The price to be paid for this efficiency is a very high sensitivity to transmission errors, which may lead to desynchronisations of the entropy codes, impacting many frames of the decoded video.

A classical technique to increase the robustness to transmission errors of compressed multimedia contents is to use forward error-correcting codes (FEC) [3]. However, in presence of time-varying characteristics of the transmission channel, the FEC may be oversized, leading to a waste of the available channel capacity, or may not be strong enough, resulting in residual errors. Adaptation of the FEC is possible but requires some information on the transmission channel via a feedback link between the emitter and the receiver, which is not always available, as in the case of video broadcasting [4]. The robustness of the compressed bitstream may also be increased by introducing some redundancy in the headers and in the data via synchronisation markers to limit the desynchronisation of entropy codes, see, *e.g.*, [5]. Error concealment techniques [6, 7] may then be used to replace the damaged parts of the bitstream.

Joint Source-Channel (JSC) decoding techniques use the residual redundancy in the bitstream generated by classical multimedia encoders to combat transmission errors at receiver side, see [8] and the references therein. Redundancy introduced by FEC and synchronisation markers may readily be used by JSC decoders, which have been successfully applied to several multimedia coders such as JPEG2000 [9], MPEG4 [10], H.263+ [11], H.264/AVC [12], or MPEG4/AAC [13]. These techniques have been applied only recently to DWT-based video coders [14]. The difficulty here comes from the very little amount of residual redundancy left by an entropy coder such as 3D-ESCOT [15] used in Vidwav (no entropy coder was considered in [14]). Moreover, a trellis-based decoding technique such as that used in [16] cannot be applied here, since 3D-ESCOT cannot be easily represented by a finite-state automaton (from which the trellis is derived) with a reasonable number of states.

The aim of this paper is to propose a JSC decoding scheme suited to the maximum-likelihood (ML) decoding of DWT- and entropy-coded video bitstreams. The main idea is to perform an ML decoding among all sequences of bits which comply with the syntax of a video encoder such as Vidwav. As will be seen, it is not necessary to introduce additional redundancy: the residual redundancy is sufficient to check whether the syntax of the encoded blocks is satisfied.

The rest of the paper is organized as follows : Section 2 describes briefly the Vidwav codec and the structure of the bitstream it generates. The proposed transmission and JSC decoding schemes are introduced in Section 3. Experimental results are shown in Section 4, before providing some conclusions.

2. VIDWAV CODEC

Vidwav is a 3D wavelet scalable video codec using a motion compensated temporal filtering (MCTF) based on the Barbell filter [2]. In this paper it is used as a $T + 2D$ scheme which first performs the MCTF on the input video sequence then operates the spatial transform on the resulting temporal subbands, as shown in Figure 1. The spatio-temporal subbands obtained are then divided into 3D blocks which are sent to the entropy coding module.

2.1. Entropy coding

The 3D blocks are encoded independently using the 3D ESCOT algorithm [15] which performs a bitplane coding using for each bitplane three coding passes: significance propagation, magnitude refinement and cleanup. Each 3D block has a total number of N bitplanes and undergoes at most $3N - 2$ coding passes. At the end of each coding pass, a bitstream fragment is obtained and added to the bitstream formed by the fragments resulting from the previous passes. To generate an embedded bitstream consisting of several layers, rate-distortion information is calculated to help deciding in which layer each generated fragment has to be stored. Data is then packetized to form the output bitstream.

Fig. 1. Vidwav $T + 2D$ coding scheme

Fig. 2. Bitstream structure

2.2. Bitstream formation

The Vidwav encoder delivers a bitstream structured in GOPs, layers, and packets, each with its own header as shown in Figure 2.

Each layer is encoded with a target bit rate and a specific spatio-temporal resolution. A layer bitstream is formed by several packets, each of them corresponding to one component and to one temporal subband included in the layer. The packet body is divided in as many segments as spatial subbands included in the current layer. Each segment contains the 3D block substreams extracted from the current coded block bitstream. The packet header stores important information for each 3D block substream, such as the number of coding passes it results from as well as its size in bytes.

3. TRANSMISSION AND JSC DECODING SCHEMES

The proposed transmission and decoding scheme is represented in Figure 3. It consists of the Vidwav encoder delivering a bitstream composed by a set of headers and 3D block substreams, the transmission channel, and the robust Vidwav decoder. This paper proposes an error detection module at the decoder side, and an error correction module based on a sequential estimation method, which precedes the Vidwav decoder.

3.1. Transmission channel

When transmitting multimedia contents, some RTP/UDP/IP packetization process is usually considered [17] to ensure jitter compensation and playback of data packets at the receiver in the correct order. Usually, error detection mechanisms (CRCs or checksums) at lower protocol layers do not allow corrupted packets to reach the upper application (APL) layer. Implementing JSC decoding techniques at APL layer needs thus the use of *permeable* protocol layers at the receiver side [18, 19]. Such mechanism requires robust header decoding techniques [19] and transmission of bit *soft information* or reliability measures (coming from the channel decoders at physical layer) to the upper protocol layers, as detailed in [20].

Provided that such permeable protocol layers are implemented at the receiver side, it is possible to model the network packetization process, the transmission channel and the robust depacketization as an Additive White Gaussian Noise (AWGN) channel. We assume here that the noise affects only the data blocks (the main part of the bitstream) and that the headers are correctly received.

Let $\mathbf{x}^i = \left(x_1^i, ..., x_{\ell_i}^i\right)$ be the sequence of ℓ_i bits that corresponds to the i-th block substream in the final delivered compressed bitstream. The order corresponds here to the order in which the different block substreams have been inserted in the final bitstream during the packetisation process.
The vector $\mathbf{y}^i = \left(y_1^i, ..., y_{\ell_i}^i\right)$ is the corresponding output of the AWGN channel, where $y_j^i = x_j^i + b_j^i$, $j \in \{1, ..., \ell_i\}$ and b_j^i is zero-mean Gaussian noise with variance σ^2.

3.2. Joint source-channel decoding

The aim of this paper is to evaluate at the receiver side the MAP estimate $\widehat{\mathbf{x}}_{\text{MAP}}^i$ of $\mathbf{x}^i$ defined as

$$\widehat{\mathbf{x}}_{\text{MAP}}^i = \arg\max_{\mathbf{x}\in\mathcal{E}_{\ell_i}} p(\mathbf{x}|\mathbf{y}^i), \tag{1}$$

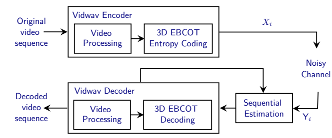

Fig. 3. Transmission and decoding scheme

where $\mathcal{E}_{\ell_i}$ is the set of all sequences of ℓ_i bits which may be generated by the encoder for the i-th 3D block. Since $\mathcal{E}_{\ell_i}$ consists of 3D ESCOT entropy-coded substreams of the same length, it is very likely that all of them have similar *a priori* probabilities. In this paper, they will be assumed to be equal to $1/\left|\mathcal{E}_{\ell_i}\right|$, where $\left|\mathcal{E}_{\ell_i}\right|$ is the cardinal number of $\mathcal{E}_{\ell_i}$. With this hypothesis, (1) becomes

$$\widehat{\mathbf{x}}_{\text{MAP}}^i = \arg\max_{\mathbf{x}\in\mathcal{E}_{\ell_i}} p(\mathbf{y}^i|\mathbf{x}). \tag{2}$$

The main difficulty here consists in determining whether a sequence $\mathbf{x}$ belongs to $\mathcal{E}_{\ell_i}$. The decoded data for the i-th block substream correspond to a known number n_i of quantized wavelet coefficients. Assume that the length in bits ℓ_i of the substream is known from the headers. Thus, the 3D ESCOT decoder has to process *exactly* ℓ_i bits of data. When there is an error in the ℓ_i bits, the decoder may be desynchronised [16],*i.e.*, more or less than ℓ_i bits may be needed to generate the n_i coefficients. This indicates the occurence of at least one transmission error. Nevertheless, some transmission errors do not lead to a desynchronisation, and are thus not detectable. This technique allows to get a test $t_i(\mathbf{x})$ satisfying

$$t_i(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in \mathcal{E}_{\ell_i}, \\ 0 & \text{else.} \end{cases} \tag{3}$$

Note that, as for channel codes, since some transmission errors are not detected, $t_i(\mathbf{x}) = 1$ does only mean that $\mathbf{x}$ belongs to $\mathcal{E}_{\ell_i}$, and not that $\mathbf{x}$ has been sent by the transmitter.

In practice, ℓ_i is only known at the precision of a *byte* (data are byte-aligned). Error detection is thus only possible when the desynchronisation is sufficient. The test $t_i'(\mathbf{x})$ which may be implemented is thus only able to detect whether the candidate sequence

$\mathbf{x}$ belongs to the set $\mathcal{E}'_{\lceil \ell_i/8 \rceil} = \mathcal{E}_{8(\lceil \ell_i/8 \rceil -1)+1} \cup \cdots \cup \mathcal{E}_{8\lceil \ell_i/8 \rceil}$, of all sequences which may be stored in $\lceil \ell_i/8 \rceil$ bytes, where $\lceil \cdot \rceil$ means upwards rounding. Since $\mathcal{E}_{\ell_i} \subset \mathcal{E}'_{\lceil \ell_i/8 \rceil}$, one has

$$t'_i(\mathbf{x}) = 0 \Rightarrow \mathbf{x} \notin \mathcal{E}'_{\lceil \ell_i/8 \rceil} \Rightarrow \mathbf{x} \notin \mathcal{E}_{\ell_i}. \quad (4)$$

A way of implementing the test $t'_i(\mathbf{x})$, is to first decode $\mathbf{x}$ using the 3D ESCOT decoder. The obtained sequence is then encoded with the 3D ESCOT encoder leading to a sequence $\tilde{\mathbf{x}}$ of h_i bits. One has

$$(t'_i(\mathbf{x}) = 1) \Leftrightarrow (8\lceil h_i/8 \rceil = 8\lceil \ell_i/8 \rceil). \quad (5)$$

An exact MAP estimation would have to consider all sequences in $\mathcal{E}'_{\lceil \ell_i/8 \rceil}$. This set is not well structured, *i.e.*, it may not be described using a trellis, for which efficient channel-coding inspired decoding techniques are available, as is the case for Huffman-like entropy codes [8]. A sequential decoder [21] is thus used to perform a suboptimal evaluation of (2) as shown in the next section.

3.3. Sequential estimation

The set of all sequences of $\ell'_i = 8\lceil \ell_i/8 \rceil$ bits may be organised in a tree $\mathcal{T}_{\ell'_i}$ containing $2^{\ell'_i}$ leaves, each of which represents a sequence of ℓ'_i bits. A path starting from R, the root of the tree and leading to a node at depth ℓ in the tree, represents a sequence of ℓ bits. Only $\left|\mathcal{E}'_{\lceil \ell_i/8 \rceil}\right|$ leaves correspond to sequences belonging to $\mathcal{E}'_{\lceil \ell_i/8 \rceil}$. The M-algorithm [21] is a sequential decoder which performs a partial exploration of the tree $\mathcal{T}_{\ell'_i}$ in order to find the M best sequences according to a given metric $\mathcal{M}$. It uses then the test $t'_i(\mathbf{x})$ to eliminate candidates which do not belong to $\mathcal{E}'_{\lceil \ell_i/8 \rceil}$. The considered metric here is

$$\mathcal{M}(\mathbf{x}_j, \mathbf{y}^i_j) = -\log p(\mathbf{y}^i_{1:j}|\mathbf{x}_{1:j}), \quad (6)$$

where $\mathbf{x}_j = (x_1, ..., x_j)$ and $\mathbf{y}^i_j = (y^i_1, ..., y^i_j)$. The M-algorithm manages a list $\mathcal{L}$ of candidates, initialized with an empty path corresponding to the root R of $\mathcal{T}_{\ell'_i}$, to which the null metric is assigned. It goes through the following steps:

1. Extend all paths in $\mathcal{L}$ to the following nodes in $\mathcal{T}_{\ell'_i}$.
2. Among the extended paths, keep at most the M best paths according to the metric (6).
3. Go to step 1. until all paths reach ℓ'_i bits.
4. Select the path in $\mathcal{L}$ with the largest metric satisfying $t'_i(\mathbf{x}) = 1$.

The M algorithm is suboptimal: if M is not large enough, the correct path may be lost at Step 2. When none of the obtained paths satisfies $t'_i(\mathbf{x}) = 1$, one concludes that the path corresponding to the actual 3D block substream has been removed. Ideally, when this case occurs, an error concealment technique should be used to replace the damaged 3D block. Here, for the sake of simplicity, the estimate corresponding to the M-th path is used even if it is wrong.

4. EXPERIMENTAL RESULTS

Experiments have been conducted using the transmission scheme described in Section 3. The 32 first frames of the QCIF sequence `foreman` are encoded in one layer at 128 kbps with both temporal and spatial transforms involving 3 decomposition levels. Entropy coding is done on 3D blocks having a depth of 4 frames. The term *Block size* will further be used to refer to the width and height of these blocks. The encoded bitstream of this sequence is then sent to the AWGN channel simulator, which only affects the block substreams. Header data, which represent about 18% of the bitstream, are assumed to be error-free. This could be obtained by using the FEC codes for example. The sequential decoding described in Section 3.3, is then run with $M = 2, 6$ and 10 for each block substream. Experiments are done with block sizes of 22×18 and 11×9. Considering small 3D blocks reduces somewhat the encoder efficiency, in the case of no error, as illustrated in Table 1. Decreasing the block size increases the amount of blocks in the bitstream and thus the frequency at which the arithmetic encoder is reinitialized. The number of noise realisations is set to 100. Figures 4 and 5 show the average PSNR as a function of the channel SNR, for a block size of 22×18 and 11×9 respectively. For a channel SNR of 11 dB, the performance gain in PSNR is 4 dB for a block size of 22×18 and 7 dB for a block size of 11×9, when compared to the standard Vidwav hard decoder (which also benefits from the noiseless headers to resynchronise in presence of errors). The gain in channel SNR for an average PSNR of 30 dB is about 1.5 dB for a block size of 11×9. Performance increases when increasing M. Experiments have been also conducted on the first 150 images of `foreman`, with the same parameters. Figure 6 shows the PSNR of the decoded sequence as a function of the frame number, for a block size of 11×9 and for a channel SNR of 11 dB. The performance increase provided by the joint decoder when compared to the standard decoder is above 10 dB in PSNR on some frames.

Block size \ Rate (kbps)	96	128	256	512
22×18	33.49	35.03	39.30	43.82
16×16	33.31	34.88	39.14	43.66
11×9	33.19	34.78	39.02	43.48

Table 1. PSNR (dB) obtained for various width and height of 3D blocks and target bitrate in the case of no-error

4.1. Discussion on the complexity

Increasing M leads to an increase in the computational complexity of the overall system. Basically, the complexity of the M-algorithm is linear in M. Since the M algorithm operates a sequential decoding only on 3D blocks that are detected as erroneous, the JSC decoding complexity for a given block is equal to that of standard decoding if no error is detected, and is at most M times the standard decoding complexity when all estimates provided by the M-algorithm are detected as erroneous. Table 2 presents the percentage of erroneous blocks and of blocks detected as erroneous, among the blocks that were stored in the bitstream (some blocks are skipped) for a sequence coded with a block size of 22×18, and with $M = 1, 2,$ and 10. It provides an upper bound on the total JSC decoding complexity as a function of the channel SNR. The error-detection rate is relatively high and has an average of 87% over the considered channel SNR values in table 2. Using the test $t_i(\mathbf{x})$ rather than $t'_i(\mathbf{x})$ would increase this error-detection rate even more, but requires additional information which is not available in current implementations of the Vidwav coder.

Fig. 4. PSNR of standard and joint decoders as a function of the channel SNR, for a block size of 22x18

Fig. 5. PSNR of standard and joint decoders as a function of the channel SNR, for a block size of 11x9

SNR	10	11	12	13
EB	37.8208	14.2893	3.4434	1.1604
BDE (M=1)	36.3239	13.5503	3.0283	0.8396
BDE (M=2)	24.4591	5.9511	0.9151	0.3522
BDE (M=10)	11.9497	1.5943	0.4591	0.1289
Upper bound on Complexity	4.2692	2.2195	1.2725	1.0756

Table 2. Evaluation of the percentage of erroneous blocks (EB) and blocks deemed as erroneous (BDE) as a function of the SNR and corresponding decoding complexity.

Fig. 6. PSNR of the encoded sequence with a block size of 11×9 for a SNR of 11dB

5. CONCLUSIONS

This paper proposes a JSC decoder able to correct transmission errors introduced by error-prone channels. It employs the residual redundancy left in the bitstream by the entropy encoder. The decoding complexity depends on the channel SNR and on the parameter M of the M-algorithm. Increasing M improves the decoding performance at the price of a higher complexity. With the considered simulation conditions, the JSC decoding complexity remains less than three times the complexity of a standard decoder for a channel SNR higher than 15 dB which is considered to be reasonable. For all 3D blocks, which were not corrected using JSC decoding, error-concealment techniques should be employed [6, 7, 22]. In this work, all headers were assumed error-free. This may be obtained by using strong FEC for the headers or JSC decoding techniques for the reliable estimation of headers, such as those presented in [23].

To improve the amount of errors detected, one may refine the substream length precision by writing the number of remaining bits in the last byte, in the corresponding packet header. Only 3 additional bits per substream would be necessary. This may improve the performance of the proposed scheme, at the price of a slight modification of the bitstream syntax.

6. REFERENCES

[1] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE trans. Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103–1120, 2007.

[2] D. Zhang J. Xu G. Pau M. Trocan S. Brangoulo R. Xiong, X. Ji and V. Bottreau, "Vidwav wavelet video coding specifications," Tech. Rep., MPEG document, Poznan, July 2005.

[3] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and applications*, Prentice-Hall, Englewood Cliffs, 1983.

[4] ETSI, "Digital video broadcasting (DVB); transmission system for handheld terminals (DVB-h)," Tech. Rep., ETSI EN 302 304 v1.1.1, nov. 2004.

[5] Z. Wu A. Bilgin and M.W. Marcellin, "Decompression of corrupt jpeg2000 codestreams," in *Data Compression Conference*, 2003.

[6] Y. Wang and Q. Zhu, "Error control and concealment for video communication: A review," *Proceedings of the IEEE*, vol. 86, pp. 974–997, 1998.

[7] M. C. Hong, H. Schwab, L. P. Kondi, and A. K. Katsaggelos, "Error concealment algorithms for compressed video," *Signal Processing: Image Communication*, vol. 14, pp. 473–492, 1999.

[8] P. Duhamel and M. Kieffer, *Joint source-channel decoding: A cross-layer perspective with applications in video broadcasting*, Academic Press, 2009.

[9] M. Grangetto, E. Magli, and G. Olmo, "Reliable JPEG 2000 wireless imaging by means of error-correcting MQ coder," in *Proc. IEEE Int. Conf. on Multimedia and Expo*, 2004, vol. 1, pp. 9–12.

[10] A. Kopansky and M. Bystrom, "Sequential decoding of MPEG-4 coded bitstreams for error resilience," in *Proc. Conference on Information Sciences and Systems*, 1999.

[11] C.M. Lee, M. Kieffer, and P. Duhamel, "Soft decoding of VLC encoded data for robust transmission of packetized video," in *Proceedings of ICASSP*, 2005, pp. 737–740.

[12] G Sabeva, S. Ben-Jamaa, M . Kieffer, and P. Duhamel, "Robust decoding of h.264 encoded video transmitted over wireless channels," in *Proceedings of MMSP*, Victoria, Canada, 2006.

[13] O. Derrien, M. Kieffer, and P. Duhamel, "Joint source/channel decoding of scalefactors in MPEG-AAC encoded bitstreams," in *Proc. European Signal Processing Conference*, 2008.

[14] M. A. Agostini, M. Kieffer, and M. Antonini, "MAP estimation of multiple description encoded video transmitted over noisy channels," in *Proc. ICIP*, 2009.

[15] S. Li J. Xu, Z.Xioong and Y. Zhang, *3-D embedded subband coding with optimal truncation (3-D ESCOT)*, J. Appl. Comput Harmon Analysis, vol. 10, pp.290-315, May 2001.

[16] S. Malinowski, H. Jegou, and C. Guillemot, "Error recovery properties and soft decoding of quasi-arithmetic codes," *EURASIP Journal on Advances in Signal Processing*, pp. 1–12, 2008, to appear.

[17] J. F. Kurose and K. W. Ross, *Computer Networking: A Top-Down Approach Featuring the Internet*, Addison Wesley, Boston, third edition, 2005.

[18] H. Jenkac, T. Stockhammer, and W. Xu, "Permeable-layer receiver for reliable multicast transmission in wireless systems," in *Proc. IEEE Wireless Communications and Networking Conference*, 13-17 March 2005, vol. 3, pp. 1805–1811.

[19] C. Marin, Y. Leprovost, M. Kieffer, and P. Duhamel, "Robust MAC-lite and soft header recovery for packetized multimedia transmission," *IEEE Trans. on Communications*, vol. 58, no. 3, 2010, to appear.

[20] G. Panza, E. Balatti, G. Vavassori, C. Lamy-Bergot, and F. Sidoti, "Supporting network transparency in 4G networks.," in *Proc. IST Mobile and Wireless Comm. Summit*, 2005.

[21] J. B. Anderson and S. Mohan, *Source and Channel Coding: An Algorithmic Approach*, Kluwer, 1991.

[22] G. Feideropoulou and B. Pequet-Popescu, "Stochastic modelling of the spatio-temporal wavelet coefficients. Application to quality enhancement and error concealment," in *EURASIP, Journal of Signal Processing and Applications*, No. 12, Sept 2004, pp. 1931-1942.

[23] R. Hu, X. Huang, M. Kieffer, O. Derrien, and P. Duhamel, "Robust critical data recovery for MPEG-4 AAC encoded bitstreams," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 2010, to appear.