



**HAL**  
open science

# Joint Exploitation of Residual Source Information and MAC Layer CRC Redundancy for Robust Video Decoding

Cedric Marin, Khaled Bouchireb, Michel Kieffer, Pierre Duhamel

► **To cite this version:**

Cedric Marin, Khaled Bouchireb, Michel Kieffer, Pierre Duhamel. Joint Exploitation of Residual Source Information and MAC Layer CRC Redundancy for Robust Video Decoding. IEEE Transactions on Wireless Communications, 2010, 9 (7), pp.2165 - 2175. hal-00549114

**HAL Id: hal-00549114**

**<https://centralesupelec.hal.science/hal-00549114>**

Submitted on 21 Dec 2010

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Joint Exploitation of Residual Source Information and MAC Layer CRC Redundancy for Robust Video Decoding

Cédric Marin, Khaled Bouchireb, Michel Kieffer, *Senior Member, IEEE*, and Pierre Duhamel, *Fellow, IEEE*

**Abstract**—This paper presents a MAP estimation method allowing the robust decoding of compressed video streams by exploiting the bitstream structure (*i.e.*, information about the source, related to variable-length codes and source characteristics) together with the knowledge of the MAC layer CRC (here considered as additional redundancy on the MAC packet). This method is implemented via a sequential decoding algorithm in which the branch selection metric in the decoding trellis incorporates a CRC-dependent factor, and the paths which are not compatible with the source constraints are pruned. A first implementation of the proposed algorithm performs exact computations of the metrics, and is thus computationally expensive. Therefore, we also introduce a suboptimal (with tunable complexity) version of the proposed metric computation. This technique is then applied to the robust decoding of sequences encoded using the H.264/AVC standard based on CAVLC, and transmitted using a WiFi-like packet structure. Significant link budget improvement results are demonstrated for BPSK modulated signals sent over AWGN channels, even in the presence of channel coding.

**Index Terms**—Communication systems, MAP estimation, Video Coding, Sequential Decoding, Codes

## I. INTRODUCTION

WIRELESS channels present a major challenge for high bitrate transmission. Factors such as signal attenuation, multiple access interference, inter-symbol interference, and Doppler shift can heavily degrade signal quality. Consequently, the typical BER encountered in mobile transmission can be several orders of magnitude higher than in wire line (*e.g.*, DSL) transmission.

High efficiency video transmission is usually dependent on the compression mechanism applied to the image stream [28]. Nevertheless, the compressed video flow is very sensitive to transmission errors. A single error can lead to a decoder desynchronization resulting in a total loss of remaining picture information or to inter-image error propagation due to inter-picture coding. Consequently, the video stream incoming in the video decoder has to be nearly error-free.

In wireless transmission, the received signal may be heavily corrupted and is not directly usable by the video decoder. A first solution to alleviate this problem consists in grouping data into packets protected by an error-detection code (CRC or checksum) [5], [16]. Packets for which integrity is not ensured at receiver side may then be retransmitted. Nevertheless, retransmissions may become difficult in scenarii with strong

delay constraints (*e.g.*, for visiophony), or even impossible when broadcasting data (*e.g.*, in satellite television).

In such situations, the standard solution is to make use of very strong error-correction codes (*e.g.*, turbo codes, LDPC) at the *Physical* (PHY) layer combined with packet-erasure codes (*e.g.*, Reed-Solomon) at intermediate protocol layers [19], [26]. However, due to the high channel variability, redundancy is rarely optimally dimensioned. It may be oversized when the channel is clear, reducing the bandwidth allocated for the data. In contrary, some corrupted packets cannot be recovered in bad channel conditions and are lost. Error-concealment techniques [9], [15] may then be used by the source decoders at the *Application* (APL) layer. They exploit the redundancy (temporal and/or spatial) in the decoded multimedia stream for estimating the missing information. However, even if very efficient for providing a video of acceptable visual quality, error concealment cannot replace a clean reception in terms of quality.

In the last years, joint source-channel decoding techniques have been proposed to correct damaged packets. Such methods involve robust source decoders, which exploit the inherent redundancy in the received packets for correcting errors. Several types of redundancy have been identified. Constraints in the syntax of variable-length codes [7], [8], [14], [24], [31] have been used first. Then, the properties due to the semantic of the source coders have been combined along with the syntax redundancy to improve the performance of robust decoders [4], [22], [27], [32]. Redundancy associated to the packetization of encoded data have been introduced in [18]. Recently, information introduced by the channel codes have been jointly employed together with the residual redundancy through iterative decoding processes [3], [21], [30]. These joint schemes improve the decoding performance when compared to classical schemes.

This paper focuses on robust decoding of video data in a downlink situation. We propose a sequential decoding algorithm jointly exploiting the syntax and semantic properties of the encoded video stream together with the redundancy at MAC layer provided by the CRC. Here, the CRC is not only used to detect errors but is also considered as an error correcting code. This CRC based decoding approach has been presented in [17], [23], [29] for correcting erroneous packets. The main contribution of this paper is to make a simultaneous usage of the CRC and the source redundancy to improve the video decoding performance. This paper is based on a variety of techniques (soft decoding of block codes [2],

sequential decoding [1], source decoding depending on syntax and semantic of bitstream [4]) which are combined to attain our objective.

Note that all robust techniques introduced above require soft information to be delivered from the PHY layer to the APL layer. Obviously, this does not correspond to a classical structure of the decoder, and requires the use of some additional tools, some of which being proposed elsewhere by the same authors. In particular, we proposed a header recovery technique exploiting the intra-layer and inter-layer redundancies along with the CRCs or checksums in [20]. With this technique, the header is very likely to be correctly decoded even for poor SNRs, and the payload may be forwarded to the upper layers, resulting in a *permeable* protocol stack. A complementary work, introducing a *transparent* network architecture, may be found in [10]. In this paper, we assume that, due to the use of such techniques, the headers are correctly received, and we concentrate on the evaluation of the payload (*i.e.*, the reception of the video)

This paper is organized as follows. After a brief description of the permeable protocol stack model in Section II, Section III describes the derivation of the decoding metric and proposes a general sequential decoding method. Reduction of complexity is presented in Section IV. Finally, the simulation results are described in Section V before drawing some conclusions.

## II. MODEL OF PERMEABLE PROTOCOL STACK

Multimedia packetized transmission usually relies on a multi-layer architecture [16] based on the RTP/UDP/IP stack.

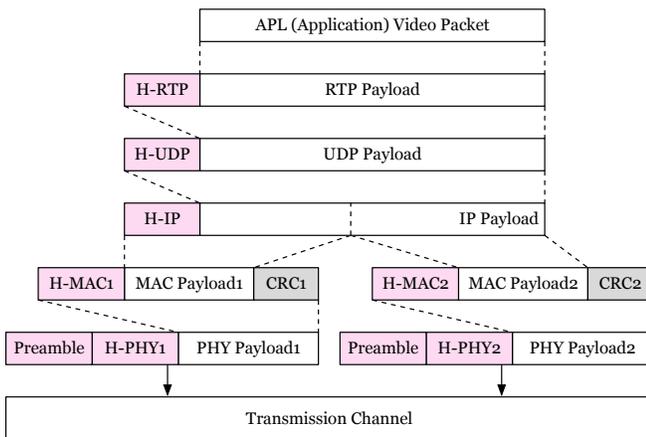


Fig. 1. Protocol stack for video transmission over WiFi.

Figure 1 illustrates an example of the segmentation and encapsulation mechanisms implemented at each protocol layer in the case of video transmitted with a WiFi radio interface [11] (802.11 standard). The data processed by the PHY layer are forwarded to the MAC layer which checks their integrity with the help of some CRC. For corrupted packets, a retransmission is requested. Correctly received data are assembled to form the binary stream that is then fed to the video decoder (at APL layer) after removal of IP, UDP, and RTP protocol headers.

A protocol stack design where the PHY, MAC, and APL layers of the receiver work very closely together is presented here. Three changes are required to implement the proposed solution:

- The PHY layer includes a *Soft-Input Soft-Output* (SISO) channel decoder for processing the incoming protected data. The soft information are transmitted to the next layer.
- In the MAC layer, the CRC check is deactivated and no retransmission is allowed. Complete MAC packets (composed of header, payload, and CRC) are transferred to the upper layer for being integrated in the payload of IP packets.
- The MAC header and MAC CRC which usually are not transmitted by the IP, UDP, and RTP layers, are now assumed to be available at the APL layer in the form of *soft* values.

These changes require some information to be available everywhere inside the receiver and are compatible with the usual transmission structure: only the receiver has to be modified, and both the transmitter operations and the signal sent are unchanged. As outlined above, they are facilitated by using the robust header recovery and permeable layer mechanisms presented in [10], [20]. Here, the headers are assumed to be available without errors at all layers.

With these modifications, the APL layer receives a succession of MAC packets, containing soft information (provided by the PHY layer). The format of data received by the APL layer is depicted in Figure 2.

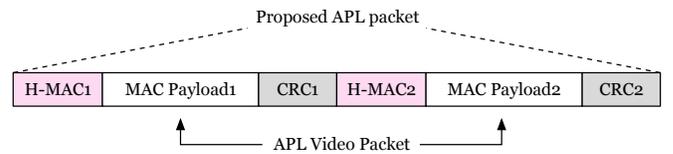


Fig. 2. New format of the APL packet at the input of the decoder. In this example, the original APL packet has been fragmented in two MAC packets.

In the proposed architecture, the CRC still plays some error-detection role, used to minimize the computational complexity: its first use is to deactivate the robust decoding process (which is computationally expensive) when:

- 1) normal CRC check is successful,
- 2) the quality of soft information provided by the lower layer is too poor, *i.e.*, when the signal power is smaller than a pre-defined threshold. In such a case, the packet is discarded (or retransmitted, see [6]).

The next section presents the analytical derivation of the decoding metric which may be used for robust reconstruction of the transmitted video sequence. We then propose a sequential decoding algorithm based on this metric.

## III. GROUP-BASED SEQUENTIAL DECODING

### A. Notations

The symbols produced by a video coder before entropy coding are assumed to be generated by a source  $\mathcal{S}$ , which

has to satisfy some semantic rules. Consider a vector  $\mathbf{m} = [m_1 \dots m_K]$  of  $K$  symbols generated by this source. The entropy coder associates a variable-length codeword  $\mathbf{x}_{m_i}$  to each component  $m_i$  of  $\mathbf{m}$ ,  $i = 1 \dots K$ , which is then mapped onto a binary sequence  $\mathbf{x} = [\mathbf{x}_{m_1} \dots \mathbf{x}_{m_K}]$ , with

$$\sum_{i=1}^K \ell(\mathbf{x}_{m_i}) = \ell(\mathbf{x}). \quad (1)$$

In (1) and in what follows,  $\ell(\mathbf{v})$  denotes the length in bits of the vector  $\mathbf{v}$ . Thus,  $\mathbf{x}$  has to be compliant with the syntax of the variable-length code (VLC) and with the semantic rules of the source  $\mathcal{S}$ .

At MAC layer, a header  $\mathbf{h}$  is added at the beginning of the *payload*  $\mathbf{x}$ , resulting in a concatenated vector  $\mathbf{d} = [\mathbf{h}, \mathbf{x}]$ . A CRC  $\mathbf{c}$  is then computed from the data  $\mathbf{d}$  and appended to  $[\mathbf{h}, \mathbf{x}]$  to form a MAC *packet*. This set of information is collected in a vector  $\mathbf{t} = [\mathbf{h}, \mathbf{x}, \mathbf{c}] = [\mathbf{d}, \mathbf{c}]$ , where  $\mathbf{c} = \mathcal{F}(\mathbf{d})$ ,  $\mathcal{F}$  being a generic encoding function.

The computation of  $\mathbf{c}$  depends on some generator polynomial  $g(z) = \sum_{i=0}^{\ell(\mathbf{c})} a_i z^i$  characterizing the CRC [5]. A systematic generator matrix  $\mathbf{G} = [\mathbf{I}, \mathbf{\Pi}]$  may be associated to  $g(z)$ . Using  $\mathbf{G}$ ,  $\mathbf{c}$  may be determined by a recursive processing over the  $\ell(\mathbf{d})$  bits of  $\mathbf{d}$  as follows

$$\mathbf{c}^{j+1} = \mathcal{F}(\mathbf{d}^{j+1}) = \mathbf{c}^j \oplus (d_{j+1} \cdot \boldsymbol{\pi}(d_{j+1})). \quad (2)$$

In (2),  $\mathbf{d}^j = [d_1 \dots d_j, 0 \dots 0]$ ,  $\boldsymbol{\pi}(d_j)$  is the  $j$ -th row of  $\mathbf{\Pi}$ , *i.e.*, the parity vector related to  $d_j$ , and  $\oplus$  represents the XOR operator. At initialization,  $\mathbf{c}^0$  is set to  $\mathbf{0}$ . After  $\ell(\mathbf{d})$  iterations, the vector  $\mathbf{c}^{\ell(\mathbf{d})}$  contains the CRC value related to  $\mathbf{d}$  (*i.e.*,  $\mathbf{c}^{\ell(\mathbf{d})} = \mathbf{c}$ ).

In our model, vector  $\mathbf{t}$  is then BPSK-modulated and transmitted over an AWGN channel that corrupts the modulated packets with a Gaussian noise of zero mean and variance  $\sigma^2$ . At the receiver, the observed vector is  $\mathbf{y}_t = [\mathbf{y}_h, \mathbf{y}_x, \mathbf{y}_c]$ , where  $\mathbf{y}_h$ ,  $\mathbf{y}_x$ , and  $\mathbf{y}_c$  are the observations of  $\mathbf{h}$ ,  $\mathbf{x}$ , and  $\mathbf{c}$  respectively.  $\mathbf{y}_t$  contains the observations of  $\mathbf{t}$  and represents a segment of the APL packet depicted in Figure 2. An overview of the transmission scheme is illustrated in Figure 3.

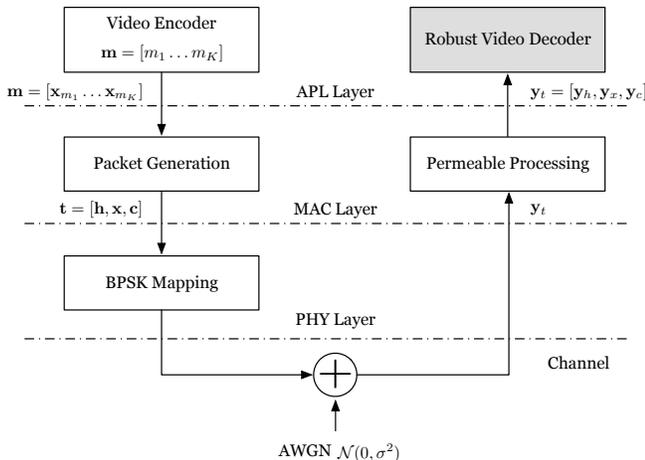


Fig. 3. Overview of the transmission scheme.

In practice,  $\mathbf{x}$  is usually organized in groups of codewords (*e.g.*, texture information of a block or a macroblock), which are assumed to be encoded independently<sup>1</sup>. Let  $\mathbf{a}_1 \dots \mathbf{a}_E$  be the  $E$  groups of codewords composing  $\mathbf{x}$ , *i.e.*,  $\mathbf{x} = [\mathbf{a}_1 \dots \mathbf{a}_E]$ . The lengths  $\ell(\mathbf{a}_e)$ , for  $e = 1 \dots E$ , are supposed to be transmitted reliably as side information to the decoder. In the following, these lengths are called *position markers*. Using these markers, the decoding of each block may be performed separately by synchronizing the decoder over the corresponding portion in the received packet.

### B. Decoding Algorithm

Assuming that the header  $\mathbf{h}$  has been correctly received, the optimal MAP estimator  $\hat{\mathbf{a}}_e$  for the  $e$ -th group is given by

$$\hat{\mathbf{a}}_e = \arg \max_{\mathbf{a}_e \in \Omega_a^e} P(\mathbf{a}_e | \mathbf{h}, \mathbf{y}_x, \mathbf{y}_c), \quad (3)$$

where  $\Omega_a^e$  is the set of *valid* combinations of  $\mathbf{a}_e$ , *i.e.*, compliant with the syntax of the VLC and the semantic of the source. Since  $\Omega_a^e$  is not well structured, obtaining  $\hat{\mathbf{a}}_e$  would require constructing the  $2^{\ell(\mathbf{a}_e)}$  possible combinations, keeping only the valid sequences (belonging to  $\Omega_a^e$ ), then evaluating  $P(\mathbf{a}_e | \mathbf{h}, \mathbf{y}_x, \mathbf{y}_c)$  for each of them. When  $\ell(\mathbf{a}_e)$  is large (which is usually the case since this reduces the overhead due to the transmission of the side information), a sequential decoder is involved in order to reduce the decoding complexity [1].

Consider the  $n$ -th step of the decoding of group  $e$ . One may write

$$\mathbf{x} = [\mathbf{b}_e, \mathbf{u}_{e,n}, \mathbf{s}_{e,n}, \mathbf{r}_{e,n}],$$

with :

- $\mathbf{b}_e = [\mathbf{a}_1 \dots \mathbf{a}_{e-1}]$ , the bits of the first  $e - 1$  groups. Note that for the decoding of  $\mathbf{a}_e$ ,  $\mathbf{b}_e$  is considered as a random vector and not as the decoded bitstream obtained previously.
- $\mathbf{u}_{e,n}$ , the first bits of  $\mathbf{a}_e$  for which a set of valid combinations  $\Omega_u^{e,n}$  has been evaluated at step  $n - 1$  by the decoder.
- $\mathbf{s}_{e,n}$ , a vector for which, regardless of the syntax of the VLC and the semantic of the video coder,  $2^{\ell(\mathbf{s}_{e,n})}$  binary combinations are possible. Let  $\Omega_s^{e,n}$  be the set of these sequences.
- $\mathbf{r}_{e,n}$ , the  $\ell(\mathbf{r}_{e,n})$  remaining bits of  $\mathbf{x}$ . These bits have not yet been processed by the decoder but they are involved in the computation of the CRC.

Figure 4 illustrates the considered structure of the packet. The observations associated to these four vectors are  $\mathbf{y}_b^e$ ,  $\mathbf{y}_u^{e,n}$ ,  $\mathbf{y}_s^{e,n}$ , and  $\mathbf{y}_r^{e,n}$ . Moreover, let  $\Omega_{[u,s]}^{e,n} \subset \Omega_u^{e,n} \times \Omega_s^{e,n}$  be the set of valid pairs  $[\mathbf{u}_{e,n}, \mathbf{s}_{e,n}]$ .

At the  $n$ -th step, the sequential decoding algorithm evaluates

$$P(\mathbf{u}_{e,n}, \mathbf{s}_{e,n} | \mathbf{y}_b^{e,n}, \mathbf{y}_u^{e,n}, \mathbf{y}_s^{e,n}, \mathbf{y}_r^{e,n}, \mathbf{h}) \propto P(\mathbf{u}_{e,n}, \mathbf{s}_{e,n}, \mathbf{y}_b^{e,n}, \mathbf{y}_u^{e,n}, \mathbf{y}_s^{e,n}, \mathbf{y}_r^{e,n} | \mathbf{h}). \quad (4)$$

<sup>1</sup>In realistic situations, the groups of codewords belonging to a specific class of video coding parameters are correlated. However, we consider here that the existing dependencies are small and may be neglected.

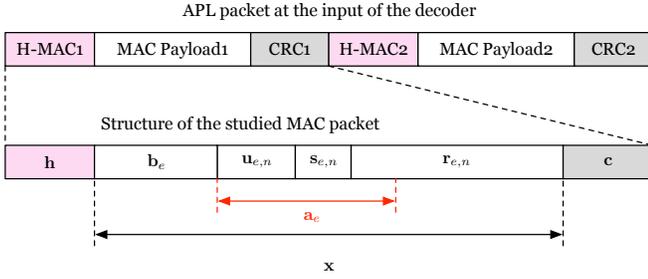


Fig. 4. Partitioning of the received MAC packet at the  $n$ -th iteration for the  $e$ -th group.

for each  $[\mathbf{u}_{e,n}, \mathbf{s}_{e,n}] \in \Omega_u^{e,n} \times \Omega_s^{e,n}$ . In (5), one may write

$$P(\mathbf{u}_{e,n}, \mathbf{s}_{e,n}, \mathbf{y}_b^{e,n}, \mathbf{y}_u^{e,n}, \mathbf{y}_s^{e,n}, \mathbf{y}_r^{e,n} | \mathbf{h}) = \sum_{\mathbf{b}_e} \sum_{\mathbf{r}_{e,n}} P(\mathbf{b}_e, \mathbf{u}_{e,n}, \mathbf{s}_{e,n}, \mathbf{r}_{e,n}, \mathbf{y}_b^{e,n}, \mathbf{y}_u^{e,n}, \mathbf{y}_s^{e,n}, \mathbf{y}_r^{e,n} | \mathbf{h}). \quad (5)$$

Moreover

$$P(\mathbf{b}_e, \mathbf{u}_{e,n}, \mathbf{s}_{e,n}, \mathbf{r}_{e,n}, \mathbf{y}_b^{e,n}, \mathbf{y}_u^{e,n}, \mathbf{y}_s^{e,n}, \mathbf{y}_r^{e,n} | \mathbf{h}) = P(\mathbf{u}_{e,n}, \mathbf{s}_{e,n} | \mathbf{h}) P(\mathbf{y}_u^{e,n} | \mathbf{u}_{e,n}, \mathbf{s}_{e,n}, \mathbf{h}) P(\mathbf{y}_s^{e,n} | \mathbf{y}_u^{e,n}, \mathbf{u}_{e,n}, \mathbf{s}_{e,n}, \mathbf{h}) P(\mathbf{b}_e, \mathbf{r}_{e,n}, \mathbf{y}_b^{e,n}, \mathbf{y}_r^{e,n}, \mathbf{y}_c | \mathbf{y}_u^{e,n}, \mathbf{y}_s^{e,n}, \mathbf{u}_{e,n}, \mathbf{s}_{e,n}, \mathbf{h}). \quad (6)$$

Using the fact that  $\mathbf{u}_{e,n}$  and  $\mathbf{s}_{e,n}$  do not depend on  $\mathbf{h}$  and that the channel is memoryless, (6) becomes

$$P(\mathbf{b}_e, \mathbf{u}_{e,n}, \mathbf{s}_{e,n}, \mathbf{r}_{e,n}, \mathbf{y}_b^{e,n}, \mathbf{y}_u^{e,n}, \mathbf{y}_s^{e,n}, \mathbf{y}_r^{e,n} | \mathbf{h}) = P(\mathbf{u}_{e,n}, \mathbf{s}_{e,n}) P(\mathbf{y}_u^{e,n} | \mathbf{u}_{e,n}) P(\mathbf{y}_s^{e,n} | \mathbf{s}_{e,n}) P(\mathbf{b}_e, \mathbf{r}_{e,n}, \mathbf{y}_b^{e,n}, \mathbf{y}_r^{e,n}, \mathbf{y}_c | \mathbf{h}, \mathbf{u}_{e,n}, \mathbf{s}_{e,n}). \quad (7)$$

Now, combining (5), (5), and (7), one obtains

$$P(\mathbf{u}_{e,n}, \mathbf{s}_{e,n} | \mathbf{y}_b^{e,n}, \mathbf{y}_u^{e,n}, \mathbf{y}_s^{e,n}, \mathbf{y}_r^{e,n}, \mathbf{h}) \propto P(\mathbf{u}_{e,n}, \mathbf{s}_{e,n}) P(\mathbf{y}_u^{e,n} | \mathbf{u}_{e,n}) P(\mathbf{y}_s^{e,n} | \mathbf{s}_{e,n}) \Phi(\mathbf{h}, \mathbf{u}_{e,n}, \mathbf{s}_{e,n}, \mathbf{y}_b^{e,n}, \mathbf{y}_r^{e,n}, \mathbf{y}_c), \quad (8)$$

with

$$\Phi(\mathbf{h}, \mathbf{u}_{e,n}, \mathbf{s}_{e,n}, \mathbf{y}_b^{e,n}, \mathbf{y}_r^{e,n}, \mathbf{y}_c) = \sum_{\mathbf{b}_e, \mathbf{r}_{e,n}} P(\mathbf{b}_e, \mathbf{r}_{e,n}, \mathbf{y}_b^{e,n}, \mathbf{y}_r^{e,n}, \mathbf{y}_c | \mathbf{h}, \mathbf{u}_{e,n}, \mathbf{s}_{e,n}). \quad (9)$$

In (8),  $P(\mathbf{u}_{e,n}, \mathbf{s}_{e,n})$  represents the *a priori* probability of sequence  $[\mathbf{u}_{e,n}, \mathbf{s}_{e,n}]$ , which is null if  $[\mathbf{u}_{e,n}, \mathbf{s}_{e,n}] \notin \Omega_{[u,s]}^{e,n}$ . As for the valid sequences, they are assumed to be equally likely *a priori*, i.e.,  $P(\mathbf{u}_{e,n}, \mathbf{s}_{e,n}) = 1/|\Omega_{[u,s]}^{e,n}|$ . Consequently, the metric  $\mathcal{M}_e$  associated to a *valid* sequence in group  $e$  is given by

$$\mathcal{M}_e([\mathbf{u}_{e,n}, \mathbf{s}_{e,n}] \in \Omega_{[u,s]}^{e,n} | \mathbf{h}, \mathbf{y}_t) = P(\mathbf{y}_u^{e,n} | \mathbf{u}_{e,n}) P(\mathbf{y}_s^{e,n} | \mathbf{s}_{e,n}) \Phi(\mathbf{h}, \mathbf{u}_{e,n}, \mathbf{s}_{e,n}, \mathbf{y}_b^{e,n}, \mathbf{y}_r^{e,n}, \mathbf{y}_c), \quad (10)$$

where  $P(\mathbf{y}_u^{e,n} | \mathbf{u}_{e,n})$  and  $P(\mathbf{y}_s^{e,n} | \mathbf{s}_{e,n})$  are the likelihoods of  $\mathbf{u}_{e,n}$  and  $\mathbf{s}_{e,n}$  respectively.

### C. Implementation Issues and Complexity

In (10),  $\Phi(\mathbf{h}, \mathbf{u}_{e,n}, \mathbf{s}_{e,n}, \mathbf{y}_b^{e,n}, \mathbf{y}_r^{e,n}, \mathbf{y}_c)$  is a sum the complexity of which is  $\mathcal{O}(2^{\ell(\mathbf{b}_e) + \ell(\mathbf{r}_{e,n})})$ . Consequently, the evaluation complexity of (5) for all  $[\mathbf{u}_{e,n}, \mathbf{s}_{e,n}] \in \Omega_{[u,s]}^{e,n}$  is  $\mathcal{O}(|\Omega_u^{e,n}| \cdot |\Omega_s^{e,n}| \cdot 2^{\ell(\mathbf{b}_e) + \ell(\mathbf{r}_{e,n})})$ .  $|\Omega_s^{e,n}|$  depends on the number

of bits taken into account at the  $n$ -th steps and may thus be upper bounded by a constant. The main difficulty comes from  $|\Omega_u^{e,n}|$ , which is growing exponentially with  $n$ . To limit the complexity increase, at each step, only the  $M$  most probable sequences belonging to  $\Omega_{[u,s]}^{e,n}$  are kept and stored in  $\Omega_u^{e,n+1}$ . The parameter  $M$  allows to tune the trade-off between complexity and efficiency.

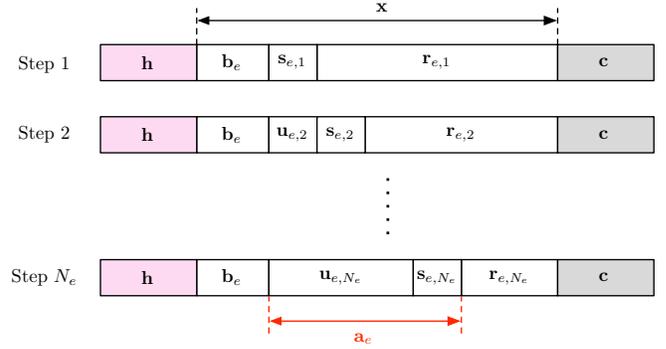


Fig. 5. Evolution of the partitions through the sequential decoding steps for the  $e$ -th group.

Figure 5 illustrates the evolution of parts  $\mathbf{b}_e$ ,  $\mathbf{u}_{e,n}$ ,  $\mathbf{s}_{e,n}$ , and  $\mathbf{r}_{e,n}$  through the different steps. The flowchart of the decoding algorithm is depicted in Figure 6 and explanations are given in the following. Note that the metric  $\mathcal{M}_e([\mathbf{u}_{e,n}, \mathbf{s}_{e,n}] | \mathbf{h}, \mathbf{y}_t)$  is computed using (10). At each step, one obtains a suboptimal algorithm the complexity of which becomes  $\mathcal{O}(2^{\ell(\mathbf{b}_e) + \ell(\mathbf{r}_{e,n})})$ , mainly due to the evaluation of  $\Phi$  in (10). Section IV describes optimal and suboptimal reduced-complexity algorithms for determining  $\Phi$  and  $\mathcal{M}_e$ .

Let  $N_e$  be the number of steps necessary to reach the end of group  $e$ . The number of bits  $\ell(\mathbf{s}_{e,n})$ , for  $i = 1 \dots N_e$ , must thus be adjusted such that

$$\sum_{i=1}^{N_e} \ell(\mathbf{s}_{e,i}) = \ell(\mathbf{a}_e), \quad (11)$$

for all  $e = 1 \dots E$ . In practice, the first  $N_e - 1$  decoding depths are set to a constant value and the last one, i.e.,  $\ell(\mathbf{s}_{e,N_e})$ , is chosen so that (11) is satisfied.

We now describe the complete sequential decoding algorithm for the  $e$ -th group. At initialization ( $n = 1$ ),  $\Omega_u^{e,1} = \emptyset$ . Afterwards, at each step  $n > 1$ , the algorithm explores the new branches (on  $\ell(\mathbf{s}_{e,n})$ -bit depth) and only preserves the  $M$  most probable extended sequences  $[\mathbf{u}_{e,n}, \mathbf{s}_{e,n}]$ . These  $M$  sequences are temporarily stored in a stack (corresponding to  $\Omega_u^{e,n+1}$ ), before being extended again at the next step.

In Section V, this algorithm is applied to the decoding of H.264/AVC CAVLC sequences.

## IV. PRACTICAL EVALUATION OF THE MAP METRIC

For the sake of simplicity, the exponents  $e$  and  $n$  are omitted in what follows. Moreover,  $\Phi(\mathbf{h}, \mathbf{u}, \mathbf{s}, \mathbf{y}_b, \mathbf{y}_r, \mathbf{y}_c)$  and  $\mathcal{M}([\mathbf{u}, \mathbf{s}] \in \Omega_{[u,s]} | \mathbf{h}, \mathbf{y}_t)$  are replaced by  $\Phi$  and  $\mathcal{M}([\mathbf{u}, \mathbf{s}])$ .

In (10), only the computation of  $\Phi$  requires a large complexity. Assuming that the bits of  $\mathbf{b}$  and  $\mathbf{r}$  are i.i.d. and do not depend on  $\mathbf{h}$ ,  $\mathbf{u}$ , and  $\mathbf{s}$ , (9) becomes

$$\Phi = \sum_{\mathbf{b}} \sum_{\mathbf{r}} P(\mathbf{b})P(\mathbf{y}_b|\mathbf{b})P(\mathbf{r})P(\mathbf{y}_r|\mathbf{r})P(\mathbf{y}_c|\mathcal{F}([\mathbf{h}, \mathbf{b}, \mathbf{u}, \mathbf{s}, \mathbf{r}])). \quad (12)$$

Assuming that all  $\mathbf{b}$  and all  $\mathbf{r}$  are equally likely *a priori*, the evaluation of (12) requires summing the product of the likelihoods related to  $\mathbf{b}$ ,  $\mathbf{r}$ , and their corresponding CRC, over the  $2^{\ell(\mathbf{b})+\ell(\mathbf{r})}$  combinations of  $\mathbf{b}$  and  $\mathbf{r}$ . In this section, two reduced-complexity methods are proposed for evaluating (10) based on two evaluations of  $\Phi$ . The first one provides an exact evaluation of  $\mathcal{M}$ , whereas the second results in an approximate (but faster) evaluation of the metric.

### A. Exact Computation

The CRC can be evaluated recursively over the data  $\mathbf{d}$ , as shown by (2). More precisely, the value of the CRC associated to the first  $j+1$  bits of  $\mathbf{d}$  (in short, at time  $j+1$ ) only depends on the value of the CRC at time  $j$  and on the  $j+1$ -st bit of  $\mathbf{d}$ . Each value of the CRC at time  $j$  leads to two different values of the CRC at time  $j+1$ . Consequently, the evolution of the CRC values according to the bits of  $\mathbf{d}$  can be described by a trellis. In this trellis, states correspond to the  $2^{\ell(\mathbf{c})}$  possible values of the CRC. Transitions are determined by the bits of  $\mathbf{d}$ . At each time  $j = 1 \dots \ell(\mathbf{d})$ , we study the contribution of  $d_j$  (the  $j$ -th bit of  $\mathbf{d}$ ) to the global CRC.

In our case,  $\mathbf{d} = [\mathbf{h}, \mathbf{b}, \mathbf{u}, \mathbf{s}, \mathbf{r}]$ . The header  $\mathbf{h}$  is assumed to be known and we want to find the best combination of  $[\mathbf{u}, \mathbf{s}] \in \Omega_{[\mathbf{u}, \mathbf{s}]}$  by taking into account the redundancy of the code (given by  $\mathbf{c}$ ). The trellis is thus applied to the portions  $\mathbf{b}$ ,  $\mathbf{r}$ , and  $\mathbf{c}$  for given  $\mathbf{h}$ ,  $\mathbf{u}$ , and  $\mathbf{s}$ . This trellis consists in grouping combinations of  $\mathbf{b}$  and  $\mathbf{r}$  giving the same value of the CRC.

Consequently, (12) may be rewritten as

$$\Phi = \sum_{\mathbf{c}} P(\mathbf{y}_c|\mathbf{c}) \sum_{\mathbf{b}, \mathbf{r} | \mathcal{F}([\mathbf{h}, \mathbf{b}, \mathbf{u}, \mathbf{s}, \mathbf{r}]) = \mathbf{c}} P(\mathbf{b})P(\mathbf{y}_b|\mathbf{b})P(\mathbf{r})P(\mathbf{y}_r|\mathbf{r}). \quad (13)$$

In the sequel, the state associated to a possible value  $\mathbf{c}'$  of CRC is denoted by  $S(\mathbf{c}')$ ,  $\mathbf{c}'$  being the binary representation of  $S(\mathbf{c}') \in \{0 \dots 2^{\ell(\mathbf{c})} - 1\}$ . For instance with a 3-bit CRC, if  $\mathbf{c}' = [1, 0, 1]$  then  $S(\mathbf{c}') = 5$ . After some derivations, one can show that (13) may be generalized as follows (see Appendix)

$$\begin{aligned} \Phi &= \sum_{\mathbf{c}'} \left[ \sum_{\mathbf{b} | \mathcal{F}([\mathbf{h}, \mathbf{b}, \mathbf{0}, \mathbf{0}, \mathbf{0}]) = \mathbf{c}'} P(\mathbf{b})P(\mathbf{y}_b|\mathbf{b}) \right] \\ &= \sum_{\mathbf{c}'} \alpha(S(\mathbf{c}')) \cdot \beta(S(\mathbf{c}' \oplus \mathcal{F}([\mathbf{0}, \mathbf{0}, \mathbf{u}, \mathbf{s}, \mathbf{0}]))) \end{aligned} \quad (14)$$

with

$$\begin{aligned} \alpha(S(\mathbf{c}')) &= \sum_{\mathbf{b} | \mathcal{F}([\mathbf{h}, \mathbf{b}, \mathbf{0}, \mathbf{0}, \mathbf{0}]) = \mathbf{c}'} P(\mathbf{b})P(\mathbf{y}_b|\mathbf{b}), \quad (15) \\ \beta(S(\mathbf{c}'')) &= \sum_{\mathbf{r}} P(\mathbf{r})P(\mathbf{y}_r|\mathbf{r})P(\mathbf{y}_c|\mathbf{c}' \oplus \mathcal{F}([\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{r}])) \end{aligned}$$

for all  $\mathbf{c}', \mathbf{c}'' \in GF(2)^{\ell(\mathbf{c})}$ . In (15),  $\alpha(S(\mathbf{c}'))$  represents the sum of the probabilities associated to the combinations of  $\mathbf{b}$  reaching state  $S(\mathbf{c}')$  when starting from state  $S(\mathcal{F}([\mathbf{h}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}]))$ . In (16),  $\beta(S(\mathbf{c}''))$  denotes the sum of the probabilities associated to all combinations of  $[\mathbf{r}, \mathbf{c}'' \oplus \mathcal{F}([\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{r}])]$  when starting from state  $S(\mathbf{c}'')$ . In fact, the evaluation of  $\Phi$  using (14) is efficiently performed using the BCJR algorithm for block codes [2], [33]. Thus,  $\alpha(S(\mathbf{c}'))$  and  $\beta(S(\mathbf{c}''))$  are easily evaluated recursively as follows

$$\begin{aligned} \alpha_{j+1}(S(\mathbf{c}')) &= P(b_{j+1} = 0)P(y_{b_{j+1}}|b_{j+1} = 0)\alpha_j(S(\mathbf{c}')) \\ &+ P(b_{j+1} = 1)P(y_{b_{j+1}}|b_{j+1} = 1) \\ &\alpha_j(S(\mathbf{c}' \oplus \pi(b_{j+1}))), \end{aligned} \quad (17)$$

with the boundary conditions (at  $j = 0$ )

$$\alpha_0(S(\mathbf{c}')) = \begin{cases} 1 & \text{for } \mathbf{c}' = \mathcal{F}([\mathbf{h}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}]) \\ 0 & \text{for all } \mathbf{c}' \neq \mathcal{F}([\mathbf{h}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}]) \end{cases}, \quad (18)$$

and

$$\begin{aligned} \beta_{j-1}(S(\mathbf{c}'')) &= P(r_j = 0)P(y_{r_j}|r_j = 0)\beta_j(S(\mathbf{c}'')) \\ &+ P(r_j = 1)P(y_{r_j}|r_j = 1)\beta_j(S(\mathbf{c}'' \oplus \pi(r_j))), \end{aligned} \quad (19)$$

with the boundary conditions (at  $j = \ell(\mathbf{r})$ )

$$\beta_{\ell(\mathbf{r})}(S(\mathbf{c}'')) = P(\mathbf{y}_c|\mathbf{c}''), \text{ for all } \mathbf{c}'' \in GF(2)^{\ell(\mathbf{c})}. \quad (20)$$

The equations in (17) and (19) are the key for computing  $\alpha(S(\mathbf{c}'))$  with a forward recursion over the bits of  $\mathbf{b}$  and  $\beta(S(\mathbf{c}''))$  with a backward recursion over the bits of  $\mathbf{r}$ . After  $\ell(\mathbf{b})$  iterations,  $\alpha_{\ell(\mathbf{b})}(S(\mathbf{c}')) = \alpha(S(\mathbf{c}'))$ , and after  $\ell(\mathbf{r})$  iterations,  $\beta_0(S(\mathbf{c}'')) = \beta(S(\mathbf{c}''))$ .

Finally, substituting (14) in (10), one obtains

$$\begin{aligned} \mathcal{M}([\mathbf{u}, \mathbf{s}]) &= \sum_{\mathbf{c}'} \alpha(S(\mathbf{c}')) \cdot P(\mathbf{y}_u|\mathbf{u})P(\mathbf{y}_s|\mathbf{s}) \\ &\beta(S(\mathbf{c}' \oplus \mathcal{F}([\mathbf{0}, \mathbf{0}, \mathbf{u}, \mathbf{s}, \mathbf{0}]))) \\ &= P(\mathbf{y}_u|\mathbf{u})P(\mathbf{y}_s|\mathbf{s}) \\ &\sum_{\mathbf{c}', \mathbf{c}'' | \mathbf{c}' = \mathbf{c}'' \oplus \mathcal{F}([\mathbf{0}, \mathbf{0}, \mathbf{u}, \mathbf{s}, \mathbf{0}])} \alpha(S(\mathbf{c}')) \cdot \beta(S(\mathbf{c}'')). \end{aligned} \quad (21)$$

The evaluation of  $\mathcal{M}([\mathbf{u}, \mathbf{s}])$  consists in summing the probabilities associated to the  $2^{\ell(\mathbf{c})}$  paths linking state  $S(\mathbf{c}')$  to state  $S(\mathbf{c}'')$ , such as  $\mathbf{c}'' = \mathbf{c}' \oplus \mathcal{F}([\mathbf{0}, \mathbf{0}, \mathbf{u}, \mathbf{s}, \mathbf{0}])$ .

The steps for evaluating the global metric (10) with the above mentioned method are summarized below:

**Step 1:** Initialize  $\alpha_0(S(\mathbf{c}'))$  and  $\beta_{\ell(\mathbf{r})}(S(\mathbf{c}''))$  according to (18) and (20).

**Step 2:** Compute  $\alpha_j(S(\mathbf{c}'))$ , for all  $\mathbf{c}' \in GF(2)^{\ell(\mathbf{c})}$  and for all  $j = 1 \dots \ell(\mathbf{b})$ , by using (17) in a forward way (partial BCJR forward step).

**Step 3:** Compute  $\beta_j(S(\mathbf{c}''))$ , for all  $\mathbf{c}'' \in GF(2)^{\ell(\mathbf{c})}$  and for all  $j = \ell(\mathbf{r}) - 1 \dots 0$ , by using (19) in a backward way (partial BCJR backward step).

**Step 4:** For each  $[\mathbf{u}, \mathbf{s}] \in \Omega_{[\mathbf{u}, \mathbf{s}]}$ , compute the metric  $\mathcal{M}([\mathbf{u}, \mathbf{s}])$  by using (21), recalling that  $\alpha(S(\mathbf{c}')) = \alpha_{\ell(\mathbf{b})}(S(\mathbf{c}'))$  and  $\beta(S(\mathbf{c}'')) = \beta_0(S(\mathbf{c}''))$ .

Hence, one step of the sequential decoding is performed with a complexity  $\mathcal{O}((\ell(\mathbf{b}) + \ell(\mathbf{r}) + |\Omega_{[\mathbf{u}, \mathbf{s}]}|)2^{\ell(\mathbf{c})})$ , compared to  $\mathcal{O}(|\Omega_{[\mathbf{u}, \mathbf{s}]}|2^{\ell(\mathbf{b})+\ell(\mathbf{r})})$  for a decoding with a straightforward metric computation.

*Remark 1:* As presented above, the decoding of  $\mathbf{x}$  requires repeating steps 1 to 4 for each portion  $[\mathbf{u}, \mathbf{s}]$  in  $\mathbf{x}$  since the portions  $\mathbf{b}$  and  $\mathbf{r}$  change according to the position of  $[\mathbf{u}, \mathbf{s}]$ . To optimize the global decoding, as soon as  $\mathbf{y}_t$  is received,  $\alpha_j(S(\mathbf{c}'))$  and  $\beta_j(S(\mathbf{c}''))$  may be computed, for all  $\mathbf{c}', \mathbf{c}'' \in GF(2)^{\ell(\mathbf{c})}$  and for all  $j = 0 \dots \ell(\mathbf{x})$ , and may be stored in matrices  $\mathbf{A}$  and  $\mathbf{B}$ . This is equivalent to perform a *complete* BCJR algorithm over  $\mathbf{x}$ : the forward step is performed on  $\mathbf{b} = \mathbf{x}$  and the backward step on  $\mathbf{r} = \mathbf{x}$ . The global decoding of  $\mathbf{x}$  begins after this step. As explained previously, each portion  $[\mathbf{u}, \mathbf{s}]$  is sequentially decoded by using (21) in which the values of  $\alpha(S(\mathbf{c}'))$  and  $\beta(S(\mathbf{c}''))$  are extracted from  $\mathbf{A}$  and  $\mathbf{B}$  depending on the position of the current portion  $[\mathbf{u}, \mathbf{s}]$ .

Note that in this case, steps 1 to 3 are performed once as a preamble, and step 4 is performed repeatedly for each  $[\mathbf{u}, \mathbf{s}]$  in  $\mathbf{x}$ .

### B. Approximate Computation

In practice, most CRCs are larger than 16 bits and the complexity  $\mathcal{O}(2^{\ell(\mathbf{c})})$  is too large to allow a real-time implementation of the method presented in Section IV-A. An approximate computation consists in splitting the CRC into  $m_b$  partitions of  $\ell(\mathbf{c})/m_b$  bits, each partition being assumed statistically independent from the others. A trellis is thus associated to each of the  $m_b$  partitions. Thus,  $\mathbf{y}_c$  may be written as  $\mathbf{y}_c = [\mathbf{y}_{c_1} \dots \mathbf{y}_{c_{m_b}}]$ . Using the independence approximation, as explained with more details in [20], the global metric in (21) becomes

$$\mathcal{M}([\mathbf{u}, \mathbf{s}]) = P(\mathbf{y}_u | \mathbf{u}) P(\mathbf{y}_s | \mathbf{s}) \prod_{m=1}^{m_b} \sum_{\mathbf{c}'_m, \mathbf{c}''_m | \mathbf{c}'_m \oplus \mathbf{c}''_m = \mathbf{c}_m \oplus \mathcal{F}_m(\{0, \mathbf{0}, \mathbf{u}, \mathbf{s}, \mathbf{0}\})} \alpha^m(S(\mathbf{c}'_m)) \cdot \beta^m(S(\mathbf{c}''_m)), \quad (22)$$

where  $\alpha^m(S(\mathbf{c}'_m))$  and  $\beta^m(S(\mathbf{c}''_m))$  represent the probabilities associated to states  $S(\mathbf{c}'_m)$  and  $S(\mathbf{c}''_m)$  respectively, for  $\mathbf{c}'_m, \mathbf{c}''_m \in GF(2)^{\ell(\mathbf{c})/m_b}$ , in the  $m$ -th trellis.

The total complexity for evaluating (22) is now  $\mathcal{O}((\ell(\mathbf{b}) + \ell(\mathbf{r}) + |\Omega_{[\mathbf{u}, \mathbf{s}]}|) m_b 2^{\ell(\mathbf{c})/m_b})$ , at the cost of a slightly suboptimal performance.

*Remark 2:* To reduce the complexity of the global decoding of  $\mathbf{x}$ , we can apply the principle introduced in Remark 1 to the new method. In this case, the algorithm generates first the  $m_b$  submatrices  $\mathbf{A}^m$  and  $\mathbf{B}^m$  associated to partition  $\mathbf{c}_m$ . During the decoding, the values of  $\alpha^m(S(\mathbf{c}'_m))$  and  $\beta^m(S(\mathbf{c}''_m))$  in (22) are extracted from  $\mathbf{A}^m$  and  $\mathbf{B}^m$  according to the position of the current portion  $[\mathbf{u}, \mathbf{s}]$ .

### C. Decoding complexity

From the two previous sections, one may evaluate the computational complexity for evaluating (10) as  $\mathcal{O}((\ell(\mathbf{b}) + \ell(\mathbf{r}) + |\Omega_{[\mathbf{u}, \mathbf{s}]}|) 2^{\ell(\mathbf{c})})$  with the exact computation and as  $\mathcal{O}((\ell(\mathbf{b}) + \ell(\mathbf{r}) + |\Omega_{[\mathbf{u}, \mathbf{s}]}|) m_b 2^{\ell(\mathbf{c})/m_b})$  with the suboptimal algorithm.

A careful comparison for the decoding of an erroneous payload  $\mathbf{x}$  depends on the choice of several parameters, which which also have an impact on the performance. It is shown in

the simulation section below that significant improvements can be obtained with respect to the classical reception algorithm even when the complexity is reduced by a factor larger than  $4.10^6$ , compared to an optimal decoding algorithm.

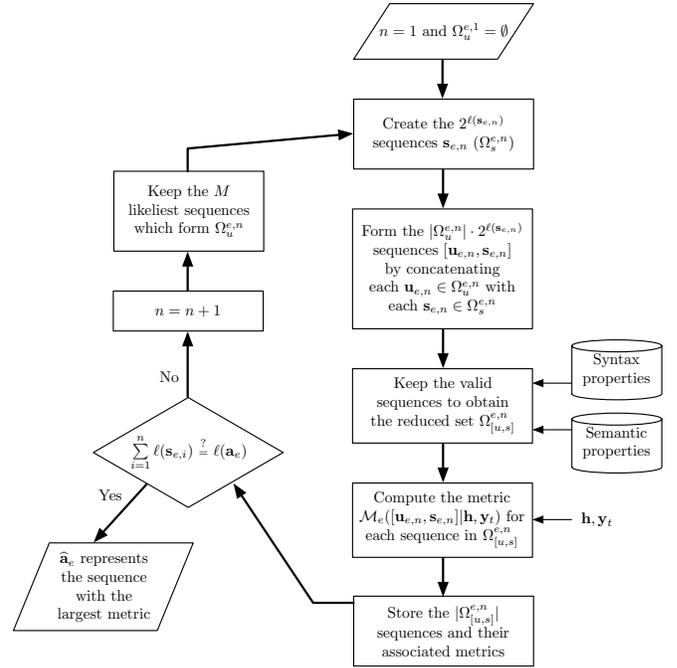


Fig. 6. Proposed sequential decoding scheme.

A payload  $\mathbf{x}$  is divided in  $E$  groups, each group  $e$  is processed iteratively in  $N_e$  steps. At each step  $n$  of the decoding of the  $e$ -th group,  $\ell(\mathbf{s}_{e,n}) \approx \ell(\mathbf{x})/E/N_e$  bits are thus decoded. As  $\Omega_{[\mathbf{u}, \mathbf{s}]}^{e,n}$  contains only the valid sequences obtained by concatenating the sequences in  $\Omega_{[\mathbf{u}]}^{e,n}$  with the  $2^{\ell(\mathbf{s}_{e,n})}$  possible sequences  $\mathbf{s}_{e,n}$ , one has  $|\Omega_{[\mathbf{u}, \mathbf{s}]}^{e,n}| \approx |\Omega_{[\mathbf{u}]}^{e,n}| \cdot 2^{\frac{\ell(\mathbf{x})}{E \cdot N_e}} = M \cdot 2^{\frac{\ell(\mathbf{x})}{E \cdot N_e}}$ , since in  $\Omega_{[\mathbf{u}]}^{e,n}$ , only the  $M$  best candidates are kept, see Figure 6. Finally, since  $\ell(\mathbf{b}) + \ell(\mathbf{r}) \approx \ell(\mathbf{x})$ , the decoding complexity is

$$C_e = \mathcal{O} \left( E \cdot N_e \cdot \left( \ell(\mathbf{x}) + M \cdot 2^{\frac{\ell(\mathbf{x})}{E \cdot N_e}} \right) 2^{\ell(\mathbf{c})} \right) \quad (23)$$

when the exact computation is performed for evaluating (10) and

$$C_e = \mathcal{O} \left( E \cdot N_e \cdot \left( \ell(\mathbf{x}) + M \cdot 2^{\frac{\ell(\mathbf{x})}{E \cdot N_e}} \right) m_b 2^{\ell(\mathbf{c})/m_b} \right) \quad (24)$$

when the suboptimal algorithm is used. The tuning parameters are thus  $E$ ,  $N_e$ ,  $M$ , and  $m_b$  in the case of the suboptimal algorithm.

Considering a large (but not too large) value of  $E$ , *i.e.*, considering many groups reduces the computational complexity. The price to be paid is an increased overhead, since more position markers are required to localize these groups. The number of decoding steps  $N_e$  of a group has also to be optimized to minimize the decoding complexity. When  $M$  is increased, the decoding complexity increases, but since more candidates are kept in  $\Omega_{[\mathbf{u}]}^{e,n}$ , the decoder may perform better. The role of  $m_b$  has already been discussed in Section IV-B.

## V. SIMULATION RESULTS

In the *extended* profile of H.264/AVC [13], an error-resilience mode is provided. In this mode, the compressed picture data are classified according to their influence on the video quality. Three partitions are defined:

- Partition A contains the headers and the motion vectors of each encoded picture.
- Partition B consists of the texture coefficients of INTRA coded blocks.
- Partition C contains the texture coefficients of INTER coded blocks.

This stream decomposition allows an adaptation of the protection to the sensitivity of the partition to be sent. After compression, each partition is encapsulated in a *Network Abstraction Layer Unit* (NALU) which is delivered to the RTP layer. In our simulation, packets associated to the A partition are assumed heavily protected and correctly interpreted at the receiver. On the other hand, B and C packets are transmitted over a noisy channel and are corrupted by transmission errors. As previously mentioned, these packets contain the texture coefficients of the various  $4 \times 4$  blocks of a picture. These blocks are encoded in CAVLC [25].

In this paper, we focus on the decoding of the CAVLC sequences included in the B and C packets. Each CAVLC sequence is considered as an independent group of codewords which can be separated from the others by using the position markers, transmitted as side information (see Section III-A). Consequently, the group-based sequential decoding method of Section III may be used for their estimation. Note that in H.264/AVC, the CAVLC sequences are not totally independent (adaptive context) but the existing dependencies are indeed small and are neglected here. The performance of the presented method has been evaluated by simulations and compared to that of two other decoding methods: a standard decoding method and a classical robust decoding method (exploiting only the source properties).

The simulated system consists of a transmitter, a channel, and a receiver. The transmitter uses repeatedly the 5 first pictures of *Foreman.cif* with the IPPPP frame structure and generates the encoded partitions using the CAVLC H.264/AVC video coder. Video packets (partitions) are then processed by the protocol stack defined in Figure 1. At the MAC layer, IP packets are fragmented in several MAC packets of variable payload size. A CRC of 4 bytes, consistent with the 802.11 standard, is added at the end of each MAC fragment. At the PHY layer of the transmitter, the data are encoded by the convolutional channel coder of the 802.11a standard [12]. Next, the coded PHY packets are mapped onto BPSK symbols before being sent over the physical medium. To improve the decoding performance, the aforementioned position markers are sent as side information, indicating the location of each  $4 \times 4$  encoded texture block in B and C packets. This side information is transmitted in a specific NALU and the markers are compressed using the Exp-Golomb coding of H.264/AVC. The overhead due to the transmission of this redundancy represents about 30 % of the total bitrate. The channel does not degrade the data contained in A packets nor the side

information. On the other hand, it does add a white Gaussian noise to the other packets. At the receiver, the data are processed by a SISO channel decoder (BCJR algorithm) and are then delivered to the APL layer (following the permeable mechanism explained in Section II). At the APL layer, three different decoders are considered:

- 1) A *standard* decoder performs hard decisions on the received soft data and makes usage of position markers to decode each block.
- 2) A *robust* decoder uses the source properties, the soft data as well as the position markers, but does not use the redundancy provided by the CRC. This decoder exploits the algorithm depicted in Section III, but the metric in (10) does not include the term  $\Phi$ .
- 3) A *CRC-robust* decoder combines all the previous sources of redundancy along with the CRC properties through the decoding method presented in Sections III and IV.

Note that, in our simulations, the two robust decoders use the same stack size  $M = 20$  and the same default decoding depth  $\ell(s) = 4$  bits. The size of a group is 11 bits in average. The CRC-robust decoder uses the suboptimal method presented in Section IV-B. For this purpose, the CRC is split into 4 blocks of 8 bits, this allows to reduce the decoding complexity by a factor of more than  $4 \cdot 10^6$ . The decoding complexity of the exact algorithm is not manageable in this context and is thus not considered.

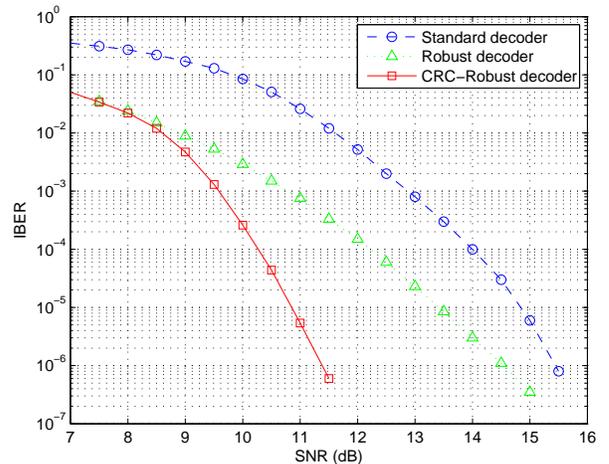


Fig. 7. Image block error rate (IBER) vs SNR for the three types of decoders, with MAC payload size of 120 bytes and deactivated channel coder/decoder at PHY layer.

Figures 7 to 10 show the evolution of the *Image Block Error Rate* (IBER) and of the *Peak Signal to Noise Ratio* (PSNR) of the decoded video as a function of the SNR for the three different decoders, with and without channel coding. In Figures 7 and 8, the channel coder/decoder at PHY layer was deactivated. In all figures, the standard, robust, and CRC-robust decoders are compared for a MAC payload size of 120 bytes.

We can notice that in terms of IBER, the standard decoder is outperformed by the two robust decoders independently

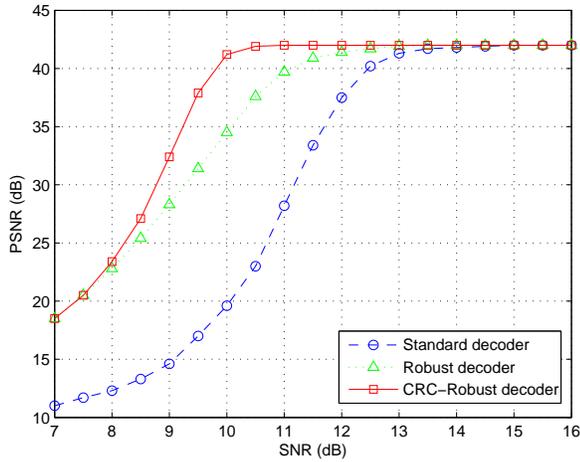


Fig. 8. Decoded image quality (PSNR) vs SNR for the three types of decoders, with MAC payload size of 120 bytes and deactivated channel coder/decoder at PHY layer.

of the presence of the outer channel code. Moreover, the two robust decoders are equivalent for low SNRs and the CRC-robust decoder outperforms the classical robust decoder above a given SNR threshold. In this region, the coding gain increases with the SNR. This behavior is specific to channel decoding performance: the CRC plays the role of an error-correcting code above this threshold. In our simulations, the threshold is about 8.5 dB in Figure 7, and 1.8 dB in Figure 9.

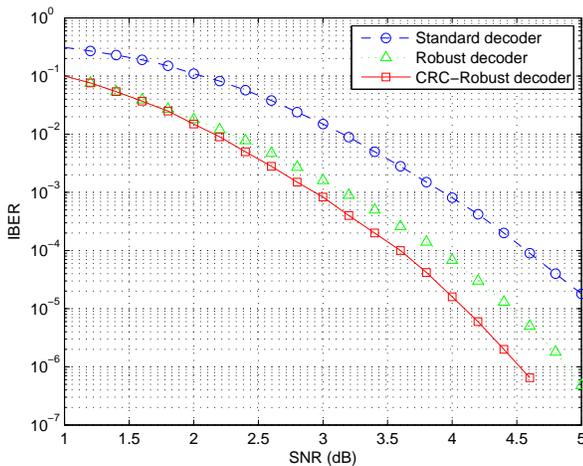


Fig. 9. Image block error rate (IBER) vs SNR for the standard, robust, and CRC-robust decoders. In this case, the 802.11a channel coder/decoder is considered at PHY layer and the MAC layer protocol of the transmitter generates 120-byte MAC payload.

In terms of PSNR, the behavior is almost similar. However, when the channel conditions are good enough, the difference in IBER does not translate in PSNR improvements, since the number of erroneous blocks is very small for all decoders. As a result, the CRC-robust decoder improves over the robust decoder only in a specific SNR range (from 8 dB to 12 dB

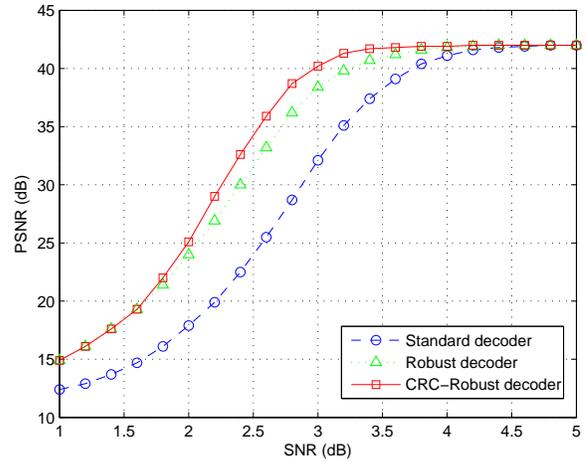


Fig. 10. Decoded image quality (PSNR) vs SNR for the standard, robust, and CRC-robust decoders. In this case, the 802.11a channel coder/decoder is considered at PHY layer and the MAC layer protocol of the transmitter generates 120-byte MAC payload.

without channel code and from 1.8 dB to 3.8 dB with a channel code).

Globally, the comparison between Figures 7 and 8 on one side and Figures 9 and 10 on the other side, shows that the presence of the convolutional code at PHY layer reduces largely the improvements brought by robust decoders, but that significant improvements are still observed. The robust decoders provide improvements as soon as the convolutional code leaves some (and not too many) uncorrected errors in the bitstream.

Figure 11 illustrates the 5-th image of the *Foreman.cif* video sequence, along with its reproductions obtained after this image is transmitted and decoded by the standard, robust and CRC-robust decoders respectively. In this figure, the channel coder/decoder is considered. This result was obtained with a payload size of 120 bytes and at an SNR of 2.8 dB for which the PSNR of the standard, robust and CRC-robust decoders are 29, 35 and 38 dB respectively (see Figure 10). Obviously, the image obtained with the standard decoder contains many artifacts and is of a very poor quality. On the other hand, the robust decoder strongly improves the quality even though some distortions are still visible. Finally, no visual difference may be noticed between the original image and the image obtained by the CRC-robust decoder.

## VI. CONCLUSION

In this paper, we have presented a MAP estimator for robust video decoding. The decoder jointly exploits the inherent source coder information along with the MAC layer CRC redundancy. The implementation of this MAP estimator was shown to be a combination of a sequential decoding algorithm along with the BCJR algorithm for obtaining appropriate metrics. We applied this method to H.264/AVC decoding of CAVLC sequences. Simulation results show that the information carried by the CRC does improve the decoding efficiency.



Fig. 11. 5-th image of *Foreman.cif* obtained after (a) error-free decoding, (b) standard decoding, (c) robust decoding, and (d) CRC-robust decoding for a SNR of 2.8 dB and a MAC payload size of 120 bytes, with channel coding.

More precisely, joint use of CRC and source properties becomes interesting above a certain threshold. It should be noted that, the bitrate used for transmitting the side information is rather high (about 30 %) in the presented experiments. We are currently working at reducing this overhead. One possibility is to consider position markers indicating, *e.g.*, the location of each macroblock of  $16 \times 16$  pixels.

The proposed method could readily be applied to H.264 with a Context Adaptive Binary Arithmetic Coder (CABAC) as entropy code. Nevertheless, the residual redundancy left by the CABAC in the compressed bitstream is less than that left by the CAVLC. The performance improvement provided by the robust decoders would probably be less significant.

#### APPENDIX

Below, we detail the derivation of (14). Assuming that the bits of  $\mathbf{b}$  and  $\mathbf{r}$  are i.i.d. and do not depend on  $\mathbf{h}$ ,  $\mathbf{u}$ , and  $\mathbf{s}$ ,  $\Phi$  in (9) may be developed as follows

$$\begin{aligned} \Phi &= \sum_{\mathbf{b}, \mathbf{r}} P(\mathbf{b})P(\mathbf{y}_b|\mathbf{b})P(\mathbf{r})P(\mathbf{y}_r|\mathbf{r}) \\ &= \sum_{\mathbf{r}} P(\mathbf{r})P(\mathbf{y}_r|\mathbf{r}) \sum_{\mathbf{b}} P(\mathbf{b})P(\mathbf{y}_b|\mathbf{b}) \\ &\quad P(\mathbf{y}_c|\mathcal{F}([\mathbf{h}, \mathbf{b}, \mathbf{u}, \mathbf{s}, \mathbf{r}])) \end{aligned} \quad (25)$$

In (25), the sum over  $\mathbf{b}$  is a sum over all the possible values that  $\mathbf{b}$  can take, each value corresponding to a path in the trellis. On the other hand, any possible path  $\mathbf{b}$  ends up at a state  $S(\mathbf{c}') \in \{0 \dots, 2^{\ell(\mathbf{c})} - 1\}$  (*i.e.*, one of the  $2^{\ell(\mathbf{c})}$  possible states). As a result, summing over all the possible paths  $\mathbf{b}$  is equivalent to summing over all the paths  $\mathbf{b}$  that end up at state 0, and all the paths that end up at state 1, ..., and all the paths that end up at state  $2^{\ell(\mathbf{c})} - 1$ . Hence, (25) becomes

$$\begin{aligned} \Phi &= \sum_{\mathbf{r}} P(\mathbf{r})P(\mathbf{y}_r|\mathbf{r}) \sum_{\mathbf{c}'} \sum_{\mathbf{b}|\mathcal{F}([\mathbf{h}, \mathbf{b}, \mathbf{0}, \mathbf{0}, \mathbf{0}])=\mathbf{c}'} P(\mathbf{r})P(\mathbf{y}_r|\mathbf{r}) \\ &\quad P(\mathbf{y}_c|\mathbf{c}' \oplus \mathcal{F}([\mathbf{0}, \mathbf{0}, \mathbf{u}, \mathbf{s}, \mathbf{r}])) \\ &= \sum_{\mathbf{c}'} \sum_{\mathbf{b}|\mathcal{F}([\mathbf{h}, \mathbf{b}, \mathbf{0}, \mathbf{0}, \mathbf{0}])=\mathbf{c}'} P(\mathbf{b})P(\mathbf{y}_b|\mathbf{b}) P(\mathbf{r})P(\mathbf{y}_r|\mathbf{r}) \\ &\quad P(\mathbf{y}_c|\mathbf{c}' \oplus \mathcal{F}([\mathbf{0}, \mathbf{0}, \mathbf{u}, \mathbf{s}, \mathbf{r}])) \\ &= \sum_{\mathbf{c}'} \left[ \sum_{\mathbf{b}|\mathcal{F}([\mathbf{h}, \mathbf{b}, \mathbf{0}, \mathbf{0}, \mathbf{0}])=\mathbf{c}'} P(\mathbf{b})P(\mathbf{y}_b|\mathbf{b}) \right] \\ &\quad \left[ \sum_{\mathbf{r}} P(\mathbf{r})P(\mathbf{y}_r|\mathbf{r})P(\mathbf{y}_c|\mathbf{c}' \oplus \mathcal{F}([\mathbf{0}, \mathbf{0}, \mathbf{u}, \mathbf{s}, \mathbf{r}])) \right] \\ &= \sum_{\mathbf{c}'} \alpha(S(\mathbf{c}')) \cdot \beta(S(\mathbf{c}' \oplus \mathcal{F}([\mathbf{0}, \mathbf{0}, \mathbf{u}, \mathbf{s}, \mathbf{0}]))), \end{aligned}$$

with

$$\alpha(S(\mathbf{c}')) = \sum_{\mathbf{b}|\mathcal{F}([\mathbf{h}, \mathbf{b}, \mathbf{0}, \mathbf{0}, \mathbf{0}])=\mathbf{c}'} P(\mathbf{b})P(\mathbf{y}_b|\mathbf{b}),$$

$$\beta(S(\mathbf{c}'')) = \sum_{\mathbf{r}} P(\mathbf{r})P(\mathbf{y}_r|\mathbf{r})P(\mathbf{y}_c|\mathbf{c}'' \oplus \mathcal{F}([\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{r}])).$$

## REFERENCES

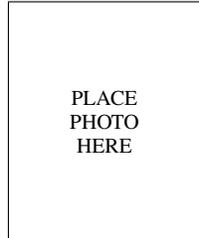
- [1] J. B. Anderson and S. Mohan. *Source and Channel Coding: An Algorithmic Approach*. Kluwer, 1991.
- [2] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv. Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Trans. Info. Theory*, 20(2):284–287, 1974.
- [3] R. Bauer and J. Hagenauer. On variable length codes for iterative source/channel decoding. In *Proc. of DCC*, pages 272–282, Snowbird, UT, 1998.
- [4] C. Bergeron and C. Lamy-Bergot. Soft-input decoding of variable-length codes applied to the H.264 standard. In *Proc. of MSP*, pages 87–90, 29 Sept.-1 Oct. 2004.
- [5] R. E. Blahut. *Theory and Practice of Error Control Codes*. Addison-Wesley, Reading, MA, 1984.
- [6] K. Bouchireb, C. Marin, P. Duhamel, and M. Kieffer. Improved retransmission scheme for video communication systems. In *Proc. IEEE PIMRC*, pages 1–5, Cannes, France, 2008.
- [7] V. Buttigieg and P. Farrell. A MAP decoding algorithm for variable-length error-correcting codes. In *Codes and Cyphers: Cryptography and Coding IV*, pages 103–119, Essex, England, 1995. The Inst. of Mathematics and its Appl.
- [8] J. Hagenauer. Source-controlled channel decoding. *IEEE Trans. Com.*, 43(9):2449–2457, 1995.
- [9] M.-C. Hong, H. Schwab, L. P. Kondi, and A. K. Katsaggelos. Error concealment algorithms for compressed video. *Signal Processing: Image Communication*, 14(6-8):473–492, 1999.
- [10] J. Huusko, J. Vehkaperä, P. Amon, C. Lamy-Bergot, G. Panza, J. Peltoja, and M. G. Martini. Cross-layer architecture for scalable video transmission in wireless network. *IEEE Signal Processing: Image Communication*, 22(3):317–330, 2007.
- [11] IEEE. 802.11, part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications. Technical report, 1999.
- [12] IEEE. 802.11a, part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, high-speed physical layer in the 5 GHz band. Technical report, 1999.
- [13] ITU-T and ISO/IEC JTC 1. Advanced video coding for generic audiovisual services. Technical report, ITU-T Rec. H.264, and ISO/IEC 14496-10 AVC, nov. 2003.
- [14] S. Kaiser and M. Bystrom. Soft decoding of variable-length codes. In *Proc. of ICC*, volume 3, pages 1203–1207, New Orleans, 2000.
- [15] W.-Y. Kung, C.-S. Kim, and C.-C. J. Kuo. Spatial and temporal error concealment techniques for video transmission over noisy channels. *IEEE Trans. Circuits and Systems for Video Technology*, 16(7):789–803, 2006.
- [16] J. F. Kurose and K. W. Ross. *Computer Networking: A Top-Down Approach Featuring the Internet*. Addison Wesley, Boston, third edition, 2005.
- [17] C. Lamy and S. Merigeault. Procédé de correction d’une trame erronée par un récepteur. French patent no. 0206501, 2002.
- [18] C. Lee, M. Kieffer, and P. Duhamel. Soft decoding of VLC encoded data for robust transmission of packetized video. In *Proc. of ICASSP*, pages 737–740, 2005.
- [19] D. J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, Cambridge, 2003.
- [20] C. Marin, Y. Leprovost, M. Kieffer, and P. Duhamel. Robust MAC-lite and soft header recovery for packetized multimedia transmission. *IEEE Trans. Com.*, 2008. submitted.
- [21] H. Nguyen and P. Duhamel. Iterative joint source-channel decoding of variable length encoded video sequences exploiting source semantics. In *Proc. of ICIP*, pages 3221–3224, 2004.
- [22] H. Nguyen, P. Duhamel, J. Brouet, and D. Rouffet. Robust VLC sequence decoding exploiting additional video stream properties with reduced complexity. In *Proc. of ICME*, pages 375–378, June 2004. Taipei, Taiwan.
- [23] J. W. Nieto and W. N. Furman. Cyclic redundancy check (CRC) based error method and device. US Patent US 2007/0192667 A1, Aug. 16 2007.
- [24] L. Perros-Meilhac and C. Lamy. Huffman tree based metric derivation for a low-complexity soft VLC decoding. In *Proc. of ICC*, pages 783–787, 2002.
- [25] I. Richardson. *H.264 and MPEG-4 Video Compression: Video Coding for Next-Generation Multimedia*. John Wiley and Sons, 2003.
- [26] T. Richardson and U. Urbanke. *Modern Coding Theory*. Cambridge University Press, 2008.
- [27] G. Sabeva, S. Ben Jamaa, M. Kieffer, and P. Duhamel. Robust decoding of H.264 encoded video transmitted over wireless channels. In *Proc. of MMSP*, pages 9–13, Victoria, Canada, 2006.
- [28] K. Sayood. *Introduction to Data Compression, Second Edition*. Morgan Kaufmann, San Francisco, 2000.
- [29] Q. Spencer. Method of correcting message errors using cyclic redundancy checks. US patent no. 0040644, 2008.
- [30] R. Thobaben and J. Kliewer. On iterative source-channel decoding for variable-length encoded markov sources using a bit-level trellis. In *Proc. of SPAWC*, pages 50–54, Rome, 2003.
- [31] R. Thobaben and J. Kliewer. Robust decoding of variable-length encoded markov sources using a three-dimensional trellis. *IEEE Com. Letters*, 7(7):320–322, 2003.
- [32] T. Tillo, M. Grangetto, and G. Olmo. A flexible error resilient scheme for JPEG 2000. In *Proc. of MSP*, pages 295–298, 29 Sept.-1 Oct. 2004.
- [33] J. K. Wolf. Efficient maximum-likelihood decoding of linear block codes using a trellis. *IEEE Trans. Info. Theory*, 24:76–80, 1978.



PLACE  
PHOTO  
HERE

**Cédric Marin** Cédric Marin was born in Chaumont, France, in 1980. He received the Ing. degree in electrical engineering from the University Paris-Sud, Orsay, in 2003, and a master degree in signal processing in 2004 from the same institution. In 2009, he receives a PhD degree in electrical and computer engineering at the University of Paris-Sud. This PhD was prepared jointly with the Alcatel-Lucent Bell Labs center.

His interests are in the areas of video compression, joint source-channel decoding, iterative decoding, wireless transmission protocol, and cross-layer-design. His thesis focuses on the problematics of video transmission over wireless channels.



PLACE  
PHOTO  
HERE

**Khaled Bouchireb** Yann Leprovost is a research-engineer in the Triple-Play Wireless Networks group, Networking and Networks Research Domain, at Bell Labs in Villarsceaux, France. He holds a degree in electronic embedded systems from the Ecole Polytechnique d’Orléans, France. He began his carrier as a consultant in embedded software development working for EADS, Thales and Wavecom. He joined Alcatel’s Research and Innovation department in 2006 to work on end to end technologies for efficient compression and delivery of video streams over wireless networks. His current research interests are video compression, adaptable and robust streaming platforms, video quality monitoring.

PLACE  
PHOTO  
HERE

**Michel Kieffer** (M'02 – SM'07) was born in Sarreguemines, France in 1972. In 1995, he received the Agrégation in Applied Physics at the Ecole Normale Supérieure de Cachan, France. He received a PhD degree in Control and Signal Processing in 1999, and the Habilitation à Diriger des Recherches degree in 2005, both from the Univ Paris-Sud, Orsay, France. Michel Kieffer is an assistant professor in signal processing for communications at the Univ Paris-Sud and a researcher at the Laboratoire des Signaux et Systèmes, Gif-sur-Yvette, France. His research

interests are in joint source-channel coding and decoding techniques for the reliable transmission of multimedia contents. He is also interested in guaranteed parameter and state estimation for systems described by non-linear models.

Michel Kieffer is co-author of more than 90 contributions in journals, conference proceedings, or books. He is one of the co-author of the book *Applied Interval Analysis* published by Springer-Verlag in 2001 and of the book *Joint source-channel decoding: A crosslayer perspective with applications in video broadcasting* published by Academic Press in 2009. He is associate editor of Signal Processing since 2008.

PLACE  
PHOTO  
HERE

**Pierre Duhamel** (F'98) was born in France in 1953. He received the Eng. Degree in Electrical Engineering from the National Institute for Applied Sciences (INSA) Rennes, France in 1975, the Dr. Eng. Degree in 1978, and the Doctorat ès sciences degree in 1986, both from Orsay University, Orsay, France.

From 1975 to 1980, he was with Thomson-CSF, Paris, France, where his research interests were in circuit theory and signal processing, including digital filtering and analog fault diagnosis. In 1980, he joined the National Research Center in Telecommunications (CNET), Issy les Moulineaux, France, where his research activities were first concerned with the design of recursive CCD filters. Later, he worked on fast algorithms for computing Fourier transforms and convolutions, and applied similar techniques to adaptive filtering, spectral analysis and wavelet transforms. From 1993 to Sept. 2000, he has been professor at ENST, Paris (National School of Engineering in Telecommunications) with research activities focused on Signal processing for Communications. He was head of the Signal and Image processing Department from 1997 to 2000. He is now with CNRS/LSS (Laboratoire de Signaux et Systemes, Gif sur Yvette, France), where he is developing studies in Signal processing for communications (including equalization, iterative decoding, multicarrier systems, cooperation) and signal/image processing for multimedia applications, including source coding, joint source/channel coding, watermarking, and audio processing. He is currently investigating the application of recent information theory results to communication theory.

Dr. Duhamel was chairman of the DSP committee from 1996 to 1998, and a member of the SP for Com committee until 2001. He was an associate Editor of the IEEE Transactions on Signal Processing from 1989 to 1991, an associate Editor for the IEEE Signal Processing Letters, and a guest editor for the special issue of the IEEE Trans. on SP on wavelets.

He was Distinguished lecturer, IEEE, for 1999, and was co-general chair of the 2001 International Workshop on Multimedia Signal Processing, Cannes, France. He was also co-technical chair of ICASSP 06, Toulouse, France. The paper on subspace-based methods for blind equalization, which he co-authored, received the "Best paper award" from the IEEE transactions on SP in 1998. He was awarded the "grand prix France Telecom" by the French Science Academy in 2000. He is the co-author of the book *Joint source-channel decoding: A crosslayer perspective with applications in video broadcasting* published by Academic Press in 2009.