Joint source/channel decoding of scalefactors in MPEG-AAC encoded bitstreams

Olivier Derrien, Michel Kieffer, Pierre Duhamel

JOINT SOURCE/CHANNEL DECODING OF SCALEFACTORS IN MPEG-AAC ENCODED BITSTREAMS

Olivier Derrien[1], *Michel Kieffer*[2], *and Pierre Duhamel*[2]

[1] Université du Sud Toulon-Var
Institut des Sciences de l'Ingénieur
Av. Georges Pompidou
BP 56, 83162 La Valette du Var FRANCE
email: olivier.derrien@univ-tln.fr

[2] L2S - CNRS - SUPELEC - Université Paris-Sud
Plateau de Moulon - 91192 Gif-sur-Yvette FRANCE
email: {kieffer,pierre.duhamel}@lss.supelec.fr

ABSTRACT

This paper describes a bandwidth-efficient method for improved decoding of MPEG-AAC bitstreams when the encoded data are transmitted over a noisy channel. Assuming that the critical part (headers) of each frame has been correctly received, we apply a soft-decoding method to reconstruct the scalefactors, which represent a highly noise-sensitive part of the bitstream. The damaged spectral data are reconstructed using an intra-frame error concealment method. Two methods for soft decoding of scalefactors are described: blind mode and informed mode. In the latter, a very small amount of additional data is included in the bitstream. At medium SNR, this method provides a significant improvement in perceptual signal quality compared to the classical hard-decoding method.

1. INTRODUCTION

High-quality audio codecs, such as the MPEG-AAC [1], are used to get high compression rates without perceptual distortion. Originally designed for data transmission (or data storage) over reliable channels, these codecs are highly vulnerable to transmission errors, largely because the bitstream they generate consists of variable-length codes (VLCs). New applications, such as audio streaming over wireless networks in 3rd generation mobile communications, are characterized by unreliable transmission channels which introduce noise at bit level and frame losses. To alleviate this problem, error-correcting codes may be used, such as those described in [2]. Nevertheless, this requires additional bandwidth, which results in a decrease of the global coding efficiency.

More bandwidth-efficient solutions to increase perceptual quality have been proposed. Error concealment techniques may be used to reduce the perceptual effect of noise and frame losses, without side information or with a small amount of side information. VLC packetization, data shuffling or interleaving and data redistribution over different payloads may be used [3].

One may also try to take advantage of the residual redundancy in the bitstream to improve robustness to noise. These techniques, called joint source/channel decoding, use a soft information at the output of the channel decoder instead of standard hardbit estimates. Joint source/channel decoding has been applied to speech coding [4, 5]. Markov chains are used to model the statistical dependency between coded parameters, and maximum *a posteriori* estimates are evaluated for each coded parameter. More recently, joint source/channel decoding of VLCs have been applied to compressed images and video [6–9]. The main idea is to take advantage of the redundancy due to the syntax of the VLC but also due to the semantics of the source coder. The main advantage of these approaches is that robustness is improved without changing the bitstream syntax.

In this paper, we apply joint source/channel decoding to MPEG-AAC bitstreams. We focus on robustness to the noise at bit level and will not consider frame losses. Since the final audio quality strongly depends on the efficient decoding of scalefactors, we choose to apply the soft decoder to the scalefactor part of the AAC bitstream.

Section 2 gives an overview of MPEG-AAC decoding and discuss error sensitivity in AAC bitstreams. Section 3 introduces soft decoding of VLCs and Section 4 proposes a scheme for joint source/channel decoding of scalefactors. Finally, Section 5 shows that when simulating a transmission over an AWGN channel, perceptual signal-quality is significantly improved compared to a standard hard-decoding process.

2. SYSTEM OVERVIEW

In this study, we target unreliable transmission channels which generate bit-level noise, typically wireless networks. Usually, data are segmented in transport-level frames, according to the network protocol. We assume that, at the receiver side, reliability information at the output of the physical layer can pass through the transport layer in order to get a softbit input for the source decoder.

The MPEG-AAC standard specifies a bitstream format for encoded audio data. The bitstream is segmented in frames, of fixed length in the case of fixed bitrate encoding, or variable length in the case of variable bitrate (VBR) encoding. These are source level frames, and one frame does not necessarily correspond to a unique transport level frame. In the following, we will briefly describe the AAC frames and discuss the error sensitivity of its different parts.

Here, we consider the most simple AAC bitstream: A monophonic audio signal encoded with the *Low Complexity* profile. A frame is made up of:

- A *fixed header*, containing mainly the following information: Number of audio channels, sampling frequency, encoder profile. The same header is repeated at the beginning of each frame.
- A *variable header*, containing mainly the length of the current frame in bytes.
- An *individual channel stream* (ICS) field, containing mainly the shape and length of the transform analysis-window and the first scalefactor (called *global gain*).
- A *section* field, describing the grouping of frequency subband in so-called *sections*. For each section, the section length and the Variable Length Code (VLC) used for coding quantized transform coefficients are specified.
- A *scalefactor* field. Scalefactors determine the quantization step for each subband. Uses a differential code followed by a VLC.
- A *spectral data* field, corresponding to the quantized transform coefficients. 11 VLCs can be used, one per section. Codewords can be interleaved with escape sequences for coding high magnitudes.

- An optional field, called *data stream element* (DSE) allows the inclusion of additional raw binary data in the bitstream. These data are ignored by a standard AAC decoder, but can be used by non-normalized decoding devices.

Three categories of data can be identified in AAC frames, according to their sensitivity to errors [10]. *Critical data* consists of headers, ICS and section fields. Without these data, decoding is almost impossible. Thus, errors on critical data are usually considered equivalent to a frame loss. *Intermediate data*, which are the scalefactors. If missing, the audio output is highly distorted. The remaining part of the bitstream (spectral data) is much less sensitive to errors. If missing, the audio quality can be maintained at a satisfactory level by using error concealment techniques.

Bit level noise can generate two different types of errors: data errors and syntax errors. A data error occurs when some data at decoder output are different from the data at the encoder input. This type of error is rather difficult to identify. A syntax error means that the decoding process can not be carried on, because the bitstream does not match the syntax defined by the standard.

The classical solution for decoding a noisy bitstream consists of applying a hard decoder (thresholding on each bit value) and then run the decoder. As we will see in the last part of the paper, this decoding scheme will result in brutal degradation of the perceived audio quality when the SNR decreases.

3. SOFT-DECODING OF VLCS

A good survey on soft decoding of VLCs can be found in [11]. The main idea is to take advantage of the residual redundancy in the bitstream. Redundancy due to the *syntax* of the codewords [12] has been exploited first. Uniquely decodable VLCs, for which the Kraft-McMillan inequality [13] is strict, are redundant. Since, for such codes, there exists some finite bit-sequences that cannot be interpreted as a succession of codewords. Other sources of redundancy have been identified, leading to more efficient decoders. The symbols generated by a Markov source, encoded using a VLC designed as if the source were memoryless, can be efficiently recovered, since the codewords and the symbols share the same correlation. This correlation can be exploited at the receiver side, as proposed, *e.g.*, by [14,15]. However, in practical situations, the conditional probabilities can not easily be estimated at the decoder side, even for a first-order Markov source. In contrast, redundancy due to the semantic rules followed by the source coder can be easily identified [16], since the bitstream generated by an image, sound, speech, or video coder has (to be decodable) to satisfy some specific rules, which are known at the decoder side. When compressed data are transmitted over a network, redundancy due to data packetization [9] or to the presence of CRCs or checksums in various protocol layers [17] can be exploited to recover more efficiently the compressed data.

Once redundancy has been identified, the main challenge remains to structure it in order to design a good decoder. Trellises is an efficient way to represent all possible successions of VLC codewords which satisfy some constraints, *e.g.*, on the number of bits, on the number of codewords [14,18], or even constraints imposed by the semantics of the source coder [9]. Figure 1 represents the symbol-clock trellis proposed in [14] for the VLC $\mathcal{C} = \{0, 10, 11\}$. Each node of this trellis is identified by a pair (k, n) and corresponds to one or several sequences of n bits made up of k codewords. Each line connecting two nodes represents a codeword. Consider a binary sequence of N bits made up of K VLC codewords of $\mathcal{C}$. When both K and N are known, one obtains a *closed* trellis represented in plain lines on Figure 1. When N is unknown, the resulting trellis also incorporates the dotted lines in Figure 1. For each k, $\mathcal{N}_k$ represents the set of all values of n such that the node (k, n) is connected to the node $(0, 0)$. The size of $\mathcal{N}_k$ depends on the length of the shortest and longest codeword of the VLC and on the knowledge of N. Trellises such as the one represented in Figure 1 can be used with decoding algorithms designed for convolutionnal codes, such as the Viterbi algorithm [19], BCJR algorithm [20] or SOVA [21].

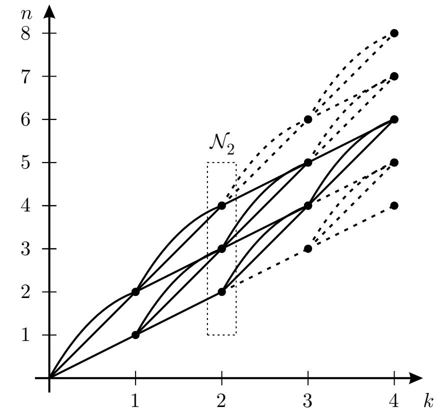

Figure 1: Symbol-level trellis, when the number K of VLC codewords is known and the corresponding number of bits N is known (plain lines) or unknown (plain and dashed lines).

4. PROPOSED DECODING SCHEME

In this section, we explain how we apply soft-decoding methods introduced in the previous section to the scalefactor field in the MPEG-AAC bitstream. We also describe the intra-frame error concealment method that we use for spectral data.

4.1 Soft-decoding of scalefactors

Soft-decoding of scalefactors is performed by the algorithm described in [14]. According to the MPEG standard, scalefactors are encoded using a single binary VLC $\mathcal{C} = \{\mathbf{c}_1, \mathbf{c}_2 \dots \mathbf{c}_M\}$ of $M = 121$ codewords. In this version of the AAC coder (Low Complexity profile), the three bits which immediately follow the scalefactors are set to zero (*Pulse data present*, *Temporal noise shaping present* and *Gain control present*). Thus, in order to improve the detection of the scalefactor sequence, we consider an additional codeword $\mathbf{c}_{M+1} = (000)$, used only to mark the end of the sequence of scalefactors. We get a new code $\mathcal{C}'$ with $M + 1$ codewords.

Consider a sequence of $K - 1$ scalefactors encoded with $\mathcal{C}$ followed by the codeword $\mathbf{c}_{M+1}$. We get a sequence of N bits $\mathbf{b}_{1:N} = (b_1, \dots, b_N)$, made up of K VLC codewords $\mathbf{c}_{1:K} = (\mathbf{c}_{i_1} \dots \mathbf{c}_{i_K})$, with $\mathbf{c}_{i_K} = \mathbf{c}_{M+1}$ and

$$N = \sum_{k=1}^{K} \ell\left(\mathbf{c}_{i_k}\right).$$

$\mathbf{b}$ passes through a memoryless channel described by $p\left(y|b\right)$. One observes a sequence of N real channel outcomes, *e.g.*, log-likelihood ratios, $\mathbf{y}_{1:N} = (y_1, \dots, y_N)$.

Assuming that N is known at the decoder, the MAP estimator of the index $X_k \in \mathcal{C}'$ of the k-th scalefactor is

$$\widehat{i}_k^{\mathrm{MAP}} = \arg \max_{i=1 \dots M+1} p\left(X_k = i | \mathbf{y}_{1:N}\right). \qquad (1)$$

A version of BCJR algorithm designed to compute (1) can be found in [14]. We have slightly adapted this algorithm to the decoding of scalefactors. First, consider $p_k(n|n') = \Pr(S_k = n|S_{k-1} = n')$, the transition probability on the tree representing the VLC $\mathcal{C}'$ and $q_k(i|n', n) = \Pr(X_k = i|S_{k-1} = n', S_k = n)$, the probability of the input symbol. These probabilities are useful, *e.g.*, to take into account the fact that the K-th symbol has the index $M + 1$, thus, for all $n \in \mathcal{N}_K$,

$$q_K(i|n-3, n) = \begin{cases} 1 & \text{if } i = M+1, \\ 0 & \text{else.} \end{cases}$$

Using these notations, to evaluate (1), we perform the expansion

$$\Pr(X_k = i|\mathbf{y}_{1:N}) \doteq \sum_{n \in \mathcal{N}_k} \sum_{n' \in \mathcal{N}_{k-1}} \alpha_{k-1}(n') \gamma_i(y_{n'+1:n}, n', n) \beta_k(n), \quad (2)$$

where $\doteq$ denotes equality up to a multiplicative constant. In (2), $\alpha_{k-1}(n')$ is evaluated using a forward recursion

$$\alpha_k(n) = \sum_{n' \in \mathcal{N}_{k-1}} \sum_{i=0}^{M-1} \alpha_{k-1}(n') \gamma_i(\mathbf{y}_{n'+1:n}, n', n),$$

with $\alpha_0(0) = 1$ and $\alpha_0(n) = 0$ for $n \neq 0$. The β_ks are evaluated with a backward recursion

$$\beta_k(n) = \sum_{n' \in \mathcal{N}_{k+1}} \sum_{i=0}^{M-1} \beta_{k+1}(n') \gamma_i(\mathbf{y}_{n'+1:n}, n, n').$$

The recursion is initialized, *e.g.*, with

$$\beta_K(n) = \begin{cases} 1/|\mathcal{N}_K| & \text{if } n \in \mathcal{N}_K, \\ 0 & \text{else,} \end{cases} \quad (3)$$

with $|\mathcal{N}_K|$ the cardinal number of $\mathcal{N}_K$.

Finally,

$$\gamma_i(y_{n'+1:n}, n', n) = q_k(i|n', n) . \Pr(\mathbf{y}_{n'+1:n}|X_k = i) . p_k(n|n'),$$

with

$$\Pr(\mathbf{y}_{n'+1:n}|X_k = i) = \prod_{j=1}^{\ell(\mathbf{c}_i)} p(y_{n'+j}|c_{i,j}).$$

A first version of the described method for estimating the scalefactors, called *informed mode*, requires the prior knowledge of both K and N at the decoder side. Then $\mathcal{N}_K = \{N\}$, and the decoding trellis is closed. Nevertheless, in classical AAC bitstream, N is difficult to obtain, since only K may be extracted from the header (assumed error-free). To know N at decoder side, one has to transmit it as a side information, using, *e.g.*, the additional data stream element (DSE) described in the MPAG-AAC standard. This results in fact in a very low relative increase in bandwidth requirements (0.7% of the total bitrate at 64 kbits/s).

Assume now that N is not known at the decoder. This situation corresponds to a second version of (2), called *blind mode*. In such case, N has to be estimated jointly with i_k, $k = 1 \dots M$. Assume that an upper bound $N_{\max}$ for N is available and that a sequence of $N_{\max}$ channel outcomes $\mathbf{y}_{1:N_{\max}}$ is observed. If $\ell_{\min}$ is the length of the smallest codeword of $\mathcal{C}$, $N_{\min} = (K-1)\ell_{\min} + 3$ is a lower bound for N. The first part of this sequence corresponds to the $K-1$ encoded scalefactors followed by the codeword $\mathbf{c}_{M+1}$. The remaining part corresponds to the beginning of the spectral-coefficients data, which immediately follow the scalefactors in the bitstream. In the blind mode, one gets $\mathcal{N}_K = \{N_{\min} \dots N_{\max}\}$ and the decoding trellis is not closed, which will make the decoding less efficient. If more information is available on the distribution of the length of the scalefactor field, it may be included in (3).

4.2 Error concealment on spectral data

Even if scalefactors have been correctly decoded, hard decoding of spectral coefficients may generate data errors. An error concealment module is used to detect errors and eventually minimize the perceived distortion. Classical error concealment techniques, designed for packet voice, do not apply to MPEG-AAC: Voice codec are usually linear-prediction based, while MPEG-AAC is a transform coder.

In [22], Korhonen proposes to replace the missing codewords by the most probable one, with respect to the codebook number specified in the *section* field. We found out that this solution results in a much lower energy than with the original signal. We propose another approach, inspired by the perceptual noise substitution (PNS) technique, proposed by Herre et al. [23] and included in the second version of MPEG-AAC [2]. The main idea is as follows: When the signal in one subband is mainly noise, coding bits can be saved by transmitting only the signal energy and replace the missing spectral coefficients by noise at the decoder side. The energy is coded instead of the scalefactor for this particular subband. In a standard AAC bitstream, the signal energy for each subband is not available, but the codebook number is closely related to the amplitude of spectral coefficients. Thus, when a data error is detected in the spectral data, spectral coefficients are replaced by a white noise. The amplitude of the coefficients is normalized with respect to the codebook number. In Table 1, we give the amplitude normalization values that give approximately the same energy as with the original signal. Our experiments showed that the best results are obtained with a Gaussian noise.

Codebook	Minimum amplitude	Maximum amplitude	PNS amplitude
1	1	1	1
2	1	1	1
3	2	2	2
4	2	2	2
5	3	4	3
6	3	4	3
7	5	7	5
8	5	7	5
9	8	12	8
10	8	12	8
11	16	∞	16

Table 1: Minimum/maximum amplitudes of spectral coefficients for each Huffman codebook and amplitude normalization for PNS reconstruction.

Error detection is difficult task: Even if no syntax error is detected in the current frame, data errors may have occurred. We observed that data errors usually result in data clipping, which is characterized by a higher energy than with the original signal. The lower-energy case is allways possible, but no severe distorsion will be percieved. In contrast, the spectral coefficients reconstructed with the proposed PNS method have approximately the same energy than the original ones. Thus, we propose a very simple detection criterion: We perform both a hard decoding of the spectral data and a PNS reconstruction. If a syntax error occurs while hard decoding, the PNS reconstruction is used. Else, we compare the energy of the reconstructed coefficients for both methods. If the hard-decoded coefficients have a significantly higher energy than the PNS coefficients, it is highly probable that a data error has occurred, and the PNS coefficients are used (see Figure 2). The parameter α allows us to set the threshold for PNS. We performed mainly an empirical optimization with respect to the final audio quality: A lower α will lead to more undetected errors, a higher value will lead to more false errors. Finally, α is set to 2 dB.

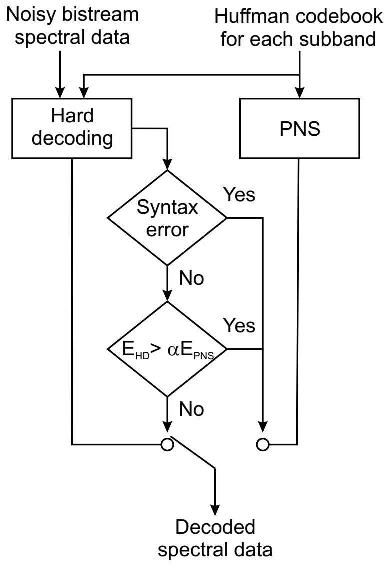

Figure 2: Algorithm for error concealment on spectral data.

5. RESULTS AND DISCUSSION

To evaluate our decoding scheme, we modulate the encoded data with a BPSK and send them on an AWGN channel. The SNR is set to the same value for each frame. Various decoding methods are then applied. Performance is measured in terms of Scalefactor Error Rate (SER) and objective perceptual quality at the decoder output, by running the PEMO-Q algorithm [24]. PEMO-Q gives a reliable prediction of subjective quality evaluations. The quality level is given by the Objective Differential Grade (ODG). An ODG of 0 means that the decoded signal is perceptually identical to the reference signal (unprocessed). An ODG of -4 means a maximum perceptual distortion. The audio material we chose for the tests is "Tom's Diner" by Suzanne Vega (first 5 seconds, sample rate 48 kHz), which was extensively used for evaluating audio codecs. The signal is coded at 64 kbits/s with a MPEG-AAC Low Complexity profile.

The decoding methods of scalefactors under test are Hard-decoding, Soft-decoding (blind mode and informed mode), and Noiseless decoding. No correlation between successive scalefactors has been considered in this work. Since errors on critical data is usually considered equivalent to a frame loss, critical data are assumed to be received without error. Spectral data are decoded with the error concealment algorithm described in Section 4.2. The noiseless decoding of scalefactors represents the ground truth, but we still consider errors affecting the spectral data.

Figure 3 shows the SER for different values of SNR. For an SER of 10^{-3}, about 1.5 dB and 1.0 dB are gained with the informed and the blind soft decoders respectively. Figure 4 represents the ODG for different values of SNR. In order to get reasonably smooth plots, 10 noise realizations have been generated and the average ODG values have been plotted.

With the hard decoder, the audio quality falls down when the SNR gets below 16 dB. The SNR/ODG slope is almost the same with the soft decoders and the noiseless decoder, but the fall comes at a lower SNR. With the noiseless decoder, the gain is about 1 dB. This improvement is due only to the error concealment method applied on spectral data. With the informed soft decoder, the gain is about 0.5 dB.

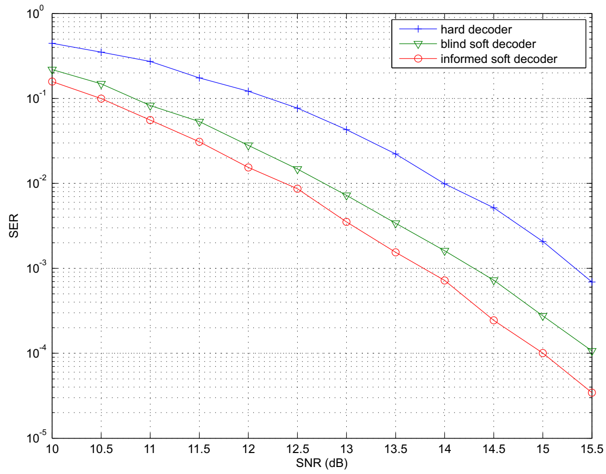

Figure 3: Scalefactor error rate for different values of the SNR on *Suzanne Vega*.

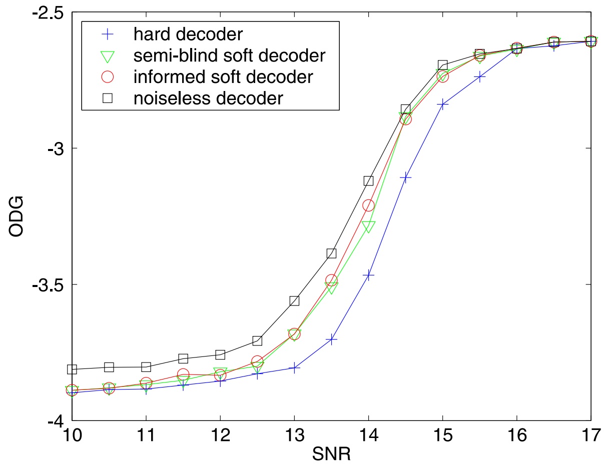

Figure 4: Objective quality evaluation for different SNR values on *Suzanne Vega*.

This is lower than the SER previously measured, because here, we combine the effect of noise on scalefactors and on spectral data. Globally, the performance of the blind and informed soft-decoders are very close. Below 13 dB, soft-decoding methods are close to the hard decoder.

6. CONCLUSION

In this paper, we consider a scheme for soft-decoding of MPEG-AAC bitstreams transmitted over noisy channels. A soft decoding of the scalefactors is done and damaged spectral data are concealed. Two soft-decoding methods are proposed: a blind mode, where no additional information is required, and an informed mode, where the bit-length of the scalefactor part of the bitstream is transmitted in the bitstream, using the additional data stream element (DSE). Compared to a classical hard-decoding scheme, the signal quality is improved at medium SNR, between 13 and 16 dB, with both soft-decoding schemes. This shows that soft decoding is an efficient method for improving the audio quality while transmitting MPEG-AAC bitstreams over noisy channels. This technique could efficiently be combined with other approaches, such as data shuffling or interleaving.

Nevertheless, this study is preliminary work. First, because we did not consider errors affecting the critical part of the bitstream. Joint source-channel decoding techniques may be used for efficient decoding of headers, by taking advantage of the high correlation between successive frames. Then, since VLCs are also used for coding spectral coefficients, joint source-channel decoding techniques may also be applied to spectral data, combined with Markov models.

REFERENCES

[1] ISO/IEC. MPEG-2 advanced audio coding, AAC. Technical Report 13818-7, International Organization for Standardization, 1997.

[2] ISO/IEC. MPEG-4 information technology - very low bitrate audio-visual coding - part3: Audio). Technical Report 14496-3, International Organization for Standardization, 1998.

[3] J. Korhonen, Y. Wang, and D. Isherwood. Towards bandwidth-efficient and error-robust audio streaming over lossy packet networks. *Multimedia Systems*, 2005.

[4] F. Alajaji, N. Phamdo, and T. Fuja. Channel codes that exploit the residual redundancy in celp-encoded speech. *IEEE Trans. on Speech and Audio Processing*, 4(5):325–336, 1996.

[5] T. Fingscheidt and P. Vary. Softbit speech decoding: A new approach to error concealment. *IEEE Trans. on Speech and Audio Processing*, 9(3):240–251, 2001.

[6] M. Bystrom, S. Kaiser, and A. Kopansky. Soft source decoding with applications. *IEEE Trans. on Circuits Syst. Video Technol.*, 11(10):1108–1120, 2001.

[7] C. Bergeron and C. Lamy-Bergot. Soft-input decoding of variable-length codes applied to the H.264 standard. In *Proc. IEEE 6th Workshop on Multimedia Signal Processing*, pages 87–90, 29 Sept.-1 Oct. 2004.

[8] H. Nguyen, P. Duhamel, J. Brouet, and D. Rouffet. Robust vlc sequence decoding exploiting additional video stream properties with reduced complexity. In *Proc. IEEE International Conference on Multimedia and Expo (ICME)*, pages 375–378, June 2004. Taipei, Taiwan.

[9] C.M. Lee, M. Kieffer, and P. Duhamel. Soft decoding of VLC encoded data for robust transmission of packetized video. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'05)*, pages 737–740, 2005.

[10] J. Korhonen and Y. Wang. Schemes for error resilient streaming of perceptually coded audio. In *Proc. International Conference on Multimedia and Expo (ICME '03)*, volume 3, pages 165–168, 6-9 July 2003.

[11] C. Guillemot and P. Siohan. Joint source-channel decoding with soft information: A survey. *Elsevier Journal on Applied Signal Processing, special issue on the turbo principle*, 6:906–927, 2005.

[12] V. Buttigieg and P.G. Farrell. On variable-length error-correcting codes. In *Information Theory, 1994. Proceedings., 1994 IEEE International Symposium on*, page 507, 27 June-1 July 1994.

[13] T. M. Cover and J. M. Thomas. *Elements of Information Theory*. Wiley, New-York, 1991.

[14] R. Bauer and J. Hagenauer. Symbol-by-symbol MAP decoding of variable length codes. In *Proc. 3rd ITG Conference Source and Channel Coding*, pages 111–116, München, 2000.

[15] L. Guivarch, P. Siohan, and J. C. Carlach. Low complexity soft decoding of huffman encoded markov sources using turbo-codes. In *Proc. ICT*, pages 872–876, Acapulco, 2000.

[16] H. Nguyen and P. Duhamel. Compressed image and video redundancy for joint source-channel decoding. In *Proc. Globecom'03*, 2003.

[17] C. Marin, P. Duhamel, K. Bouchireb, and M. Kieffer. Robust video decoding through simultaneous usage of residual source information and MAC layer CRC redundancy. In *Proc. Globecom'07*, 2007. to appear.

[18] R. Thobaben and J. Kliewer. On iterative source-channel decoding for variable-length encoded markov sources using a bit-level trellis. In *Proc. IV IEEE Signal Processing Workshop on Signal Processing Advances in Wireless Communications (SPAWC'03)*, Rome, 2003.

[19] A. J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. on Information Theory*, 13:260–269, 1967.

[20] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv. Optimal decoding of linear codes for minimizing symbol error rate. *IEEE Trans. on Information Theory*, 20:284–287, 1974.

[21] J. Hagenauer and P. Hoeher. A Viterbi algorithm with soft-decision outputs and its applications. In *Proc. Globecom'89*, pages 1680–1686, Dallas, TX, 1989.

[22] J. Korhonen. Error robustness scheme for perceptually coded audio based on interframe shuffling of samples. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'02)*, volume 2, pages 2053–2056, 2002.

[23] J. Herre and D. Shulz. Extending the mpeg-4 aac codec by perceptual noise substitution'. In *Proc. of the 104th International Conference of the Audio Engineering Society*, 1998.

[24] R. Huber and B. Kollmeier. PEMO-Q, a new method for objective audio quality assessment using a model of auditory perception. *IEEE Trans. on Audio, Speech, and Language Processing*, 14(6):1902 – 1911, 2006.