EMBEDDED TRANSFORM CODING OF AUDIO SIGNALS BY MODEL-BASED BIT PLANE CODING

Thi Minh Nguyet Hoang[1]*, Marie Oger*[1]*, Stéphane Ragot*[1]*, and Marc Antonini*[2]

[1]France Télécom R&D/TECH/SSTP, Av. Pierre Marzin, 22307 Lannion Cedex
[2]Lab. I3S-UMR 6070 CNRS and Univ. of Nice Sophia Antipolis, rte des Lucioles, 06903 Sophia Antipolis
E-mail: {thiminhnguyet.hoang,marie.oger, stephane.ragot}@orange-ftgroup.com, am@i3s.unice.fr

ABSTRACT

This paper proposes a new model-based method for transform coding of audio signals. The input signal is mapped in "perceptual" domain by linear-predictive weighting filter followed by modified discrete cosine transform (MDCT). To provide bitstream scalability, model-based bit plane coding is then applied with respect to the mean square error (MSE) criterion. We present methods to estimate the symbol probability in bit planes assuming a generalized Gaussian model for the distribution of MDCT coefficients. We compare the performance of the proposed bitstream scalable coder with stack-run coding and ITU-T G.722.1. Objective and subjective quality results are presented. The proposed coder is equivalent to or slightly worse than reference coders, but presents the nice advantage of being scalable. Performance penalty due to bitstream scalability is evident at low bitrates.

Index Terms— Transform coding, audio coding.

1. INTRODUCTION

Nowadays, many speech and audio coding standards are available. Often they are optimized for specific constraints (e.g. bit rate range, sampling rate, frame length, ...) and they use trained structures such as stored codebooks or coding tables which make coder design inflexible. Besides, multimedia communications have to deal with the problem of increasing heterogeneity of access networks (e.g. mobile, WiFi, DSL, FTTH) and terminals (e.g. legacy narrowband phones, smartphones, ...). To address heterogeneity, Bitstream scalable coding, also known as embedded coding, is a promising solution to these problems of heterogeneity in networks and lack of flexibility.

This work aims at reaching more flexibility in coder design while retaining coding efficiency. For this purpose, we use a model-based approach. Model-based coding has already shown promising results for LSF parameter quantization [1], waveform coding of speech [2], coding of transform coefficients [3] and entropy-constrained vector quantization [4]. Specifically, we propose here an embedded coding method similar to the bit plane coding used for instance in MPEG-4 BSAC and proprietary coders [5, 6, 7] for audio and JPEG2000 [8] for images. Statistical modeling is used to estimate efficiently symbol probability in bit planes.

This paper is organized as follows. We present the principle of model-based transform coding in Section 2. The proposed coder is described in Section 3. The estimation of symbol probabilities is studied in Section 4. Objective and subjective quality results are presented and discussed in Section 5 before concluding in Section 6.

This work was supported in part by the European Union under Grant FP6-2002-IST-C 020023-2 FlexCode.

2. BACKGROUND: MODEL-BASED TRANSFORM CODING

2.1. Model-based transform coding principle

The coding principle adopted in this work consists in separating perception and quantization aspects. Therefore, the input signal $x(n)$ is mapped first to a "perceptual" domain by weighting and transform operations. We assume that this "perceptual" domain is such that coding with respect to the mean square error (MSE) criterion can be applied in this domain. The transform coding structure used

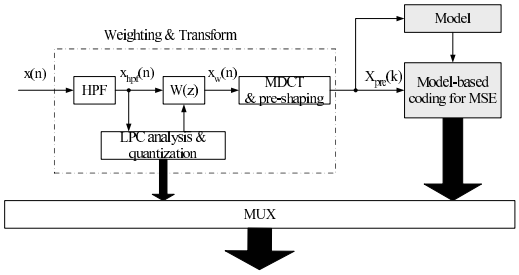

Fig. 1. Principle of model-based predictive transform coding(without noise injection).

here is illustrated in Fig. 1. This particular setup is derived from [3]. The encoder employs a linear-predictive weighting filter followed by MDCT coding. Here, the input signal $x(n)$ is sampled at 16 kHz. The frame length is 20 ms with a lookahead of 25 ms. A 2nd order elliptic high-pass filter (HPF) is applied to $x(n)$ in order to remove the frequency component under 50 Hz. An 18th order LPC analysis described in [3] is then performed on the resulting signal $x_{hpf}(n)$. The resulting LPC coefficients are quantized with 40 bits using a parametric quantization method based on a Gaussian mixture model (GMM) in the linear spectrum frequency (LSF) domain [1]. The signal $x_{hpf}(n)$ is filtered by perceptual weighting filter:

$$W(z) = \frac{\hat{A}(z/\gamma)}{1 - \beta z^{-1}} \tag{1}$$

where $\hat{A}(z/\gamma)$ is the quantized LPC filter, $\beta = 0.75$ and $\gamma = 0.92$. The coefficients of $W(z)$ are updated every 5 ms by interpolating LSF parameters. An MDCT analysis is applied on the weighted signal $x_w(n)$. The MDCT coefficients are pre-shaped to emphasize low frequencies [3] so as to correct imperfection in the short term marking curve approximated by $1/W(z)$. The distribution of pre-shaped coefficients $X_{pre}(k)$ is modeled by the pdf described next.

2.2. Generalized Gaussian model

In this work, in continuation of [3] we use the generalized Gaussian model to approximate the probability density function (pdf) transform coefficients $X_{pre}(k)$. Generality speaking, the pdf of a zero-mean generalized Gaussian random variable x of standard deviation σ is given by [9]:

$$g_{\sigma,\alpha}(x) = \frac{A(\alpha)}{\sigma}\, e^{-|B(\alpha)x/\sigma|^{\alpha}}, \tag{2}$$

where α is a shape parameter describing the exponential rate of decay and the tail of the density function,

$$A(\alpha) = \frac{\alpha B(\alpha)}{2\Gamma(1/\alpha)} \qquad \text{and} \qquad B(\alpha) = \sqrt{\frac{\Gamma(3/\alpha)}{\Gamma(1/\alpha)}}, \tag{3}$$

with

$$\Gamma(\alpha) = \int_0^{\infty} e^{-t} t^{\alpha+1}\, dt. \tag{4}$$

The special cases $\alpha = 1$ and 2 correspond to the Laplacian and Gaussian distributions respectively. In order to estimate the shape parameter α we use a method proposed by Mallat [3].

3. PROPOSED CODING STRUCTURE

3.1. Encoder

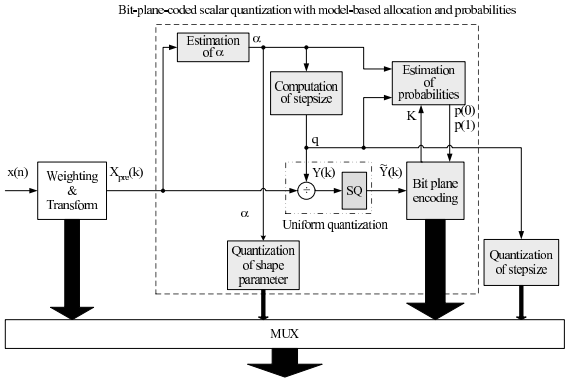

Fig. 2. Block diagram of the proposed predictive transform encoder (noise injection is not shown here).

The proposed encoder is illustrated in Fig. 2. Weighting and transform of the input signal $x(n)$ are the same as presented in Section 2.1 (see also [3]). In particular, the input sampling rate is 16 kHz and the frame length is 20 ms. A generalized Gaussian model approximates the distribution of the spectrum $X_{pre}(k)$ composed of $N = 320$ coefficients. Mallat's method [3] is used to estimate the shape parameter α on-line. The pre-shaped spectrum $X_{pre}(k)$ is divided by stepsize q and the resulting coefficients $Y(k)$ are encoded by uniform scalar quantization. Only the first 280 coefficients of the spectrum $Y(k)$ corresponding to the 0-7000 Hz band are coded; the last 40 coefficients are discarded. The integer sequence $\tilde{Y}(k)$ is encoded by bit plane coding. Note that the encoding stops when the bit budget is reached; all the non-coded bits in bit planes are replaced by zero. Therefore there is no need to implement a rate control procedure, unlike [3].

Here the stepsize q is set based on the asymptotic estimation [9]:

$$q = q_{opt} \times 2^{-margin} = \sqrt{\frac{6\lambda_{opt}}{\ln(2)}} \times 2^{-margin} \tag{5}$$

where $margin$ is a value chosen to ensure that the encoder will always use the whole bit budget, and λ_{opt} is given by :

$$\lambda_{opt} = 2^{-2B} 2\ln(2) h\sigma^2 \tag{6}$$

where σ and h are respectively the standard deviation and a function of the p.d.f. of $X_{pre}(k)$ given by [9], B is the number of bits per frame to code $X_{pre}(k)$ and $margin = 2$.
In this work bit planes are coded using adaptive arithmetic coding [10, 11]. Before using bit plane coding, the probabilities of 0 and 1 in each bit plane are needed. We propose to exploit the knowledge of the model parameters σ, α and stepsize q to estimate efficiently those probabilities.

3.2. Decoder

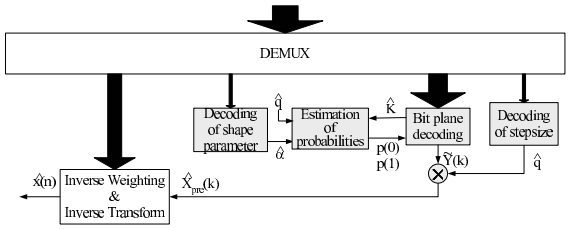

Fig. 3. Block diagram of the proposed predictive transform decoder (noise injection is not shown here).

The decoder in error-free conditions is illustrated in Figure 3. The reconstructed spectrum of pre-shaped coefficients $\hat{X}_{pre}(k)$ is given by $\hat{X}_{pre}(k) = \hat{q}\tilde{Y}(k)$, where $\tilde{Y}(k)$ is found by bit plane decoding and $\hat{q}$ is the decoded stepsize. The coefficients $\hat{X}_{pre}(k)$ are de-shaped by using an inverse weighting and inverse transform presented in [3] to find the synthesis signal $\hat{x}(n)$.

3.3. Bit allocation

The parameters of the proposed coder are line spectrum frequency (LSF) parameters, step size q, shape parameter α and noise floor level σ. The bit allocation to the parameters is detailed in Table 1, where B_{tot} is the total number of bits per frame. The allocation (in bits per sample) to bit plane coding is $B = (B_{tot} - 63)/280$.

Table 1. Bit allocation for the bit plane transform audio coding.

Parameter	Number of bits
LSF	40
Step size (q)	7
Shape parameter (α)	3
Number of bit plane (K)	4
Noise injection	9
Bit plane coding	B_{tot}-63
Total	B_{tot}

4. BIT PLANE CODING OF MDCT COEFFICIENTS

4.1. Principle of bit plane coding

In the following we treat the general case of encoding of N zero-mean i.i.d. variables $X = [x_1, \ldots, x_N]$ of variance $\sigma > 0$ with respect to MSE. Note that in this work $x = \{X_{pre}\}$. After uniform scalar quantization with stepsize q, we obtain an integer sequence $\tilde{Y} = [\tilde{y}_1, \ldots, \tilde{y}_N]$, with $\tilde{y}_i = [x_i/q]$, where $[.]$ is the rounding to the nearest integer.

The integer sequence $\tilde{Y}$ is written in binary format. First, the sign and the absolute value are separated as:

$$\tilde{y}_i = a_i(-1)^{s_i} \tag{7}$$

where $a_i = |\tilde{y}_i|$ and s_i is the sign bit defined as:

$$s_i = \begin{cases} 1 & \text{if } y_i \leq 0 \\ 0 & \text{if } y_i \geq 0 \end{cases} \tag{8}$$

Then, each absolute value a_i is decomposed in binary format as

$$a_i = \mathbf{B}_{K-1}(a_i)2^{K-1} + \ldots + \mathbf{B}_1(a_i)2^1 + \mathbf{B}_0(a_i)2^0 \tag{9}$$

where $\mathbf{B}_k(a_i)$ is the k^{th} bit of the binary format of a_i and K is the number of bit planes needed for $\tilde{Y}$:

$$K = \max(\lceil log_2(\max_{i=1,\ldots,n} a_i) \rceil, 1) \tag{10}$$

where $\lceil \cdot \rceil$ is the upper integer and $log_2(0) = -\infty$. With this binary decomposition , we get bit planes:

$$\mathbf{P}_k = [\mathbf{B}_k(a_0)\, \mathbf{B}_k(a_1) \ldots \mathbf{B}_k(a_{N-1})], \;\; k = 0, \ldots, K-1 \tag{11}$$

and the sign vector:

$$\mathbf{S} = [s_0 \;\; s_1 \ldots s_{N-1}]. \tag{12}$$

In general [5, 6, 7], the sign bit $s_i, i = 1, \ldots, N$, is transmitted only if $|a_i| \neq 0$. To allow decoding for partially received coded data, s_i is transmitted as soon as one of the coded bits $\{\mathbf{B}_k(a_i)\}_{k=0,\ldots,K-1}$ is equal to one.

4.2. Model-based estimation of probabilities for entropy coding of bit planes

After uniform scalar quantization of $X = [x_1, \ldots, x_N]$ with stepsize q, we obtain an integer sequence $\tilde{Y} = [\tilde{y}_1, \ldots, \tilde{y}_N]$. Assuming the elements of X are zero-mean i.i.d. random variables of variance σ (see Eq. 2), the probability of $\tilde{y}_i$ is given by:

$$p(\tilde{y}_i) = \int_{q\tilde{y}_i - q/2}^{q\tilde{y}_i + q/2} g_{\sigma,\alpha}(x)dx \tag{13}$$

where q is the stepsize and $g_{\sigma,\alpha}(x)$ is the p.d.f. defined in Section 2.2. Without loss of generality the stepsize is normalized to q/σ and σ is normalized to 1.

We show in the following how symbol probabilities in each bit plane can be estimated based on $p(\tilde{y}_i)$, where $|\tilde{y}_i| \leq M$ with $M = 2^K - 1$ is the maximal absolute value to be coded. Note that we assume that the number of bit planes K is sent as side information to the decoder.

4.2.1. First method: adaptive arithmetic coding with model-based initialization of probability tables

In bit plane coding, successive planes $\mathbf{P}_k$ are coded in the order from MSB to LSB by an arithmetic adaptive coding [11, 10]. The probability of having the k^{th} bit in the binary decomposition of a_i in the bit plane $\mathbf{P}_k$ equals to zero is given by:

$$p(\mathbf{B}_k(a_i) = 0) = p(\tilde{y}_i) \times \delta_{\mathbf{B}_k(a_i),0} \tag{14}$$

with

$$\delta_{x,y} = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases}$$

Based on the assumption that the number of bit planes K is sent by the encoder, we can further exploit the a priori information $a_i \leq M$, so the probability of having zero in bit plane $\mathbf{P}_k$ is given by:

$$p(b_k = 0 | a_i \leq M) = \frac{p(b_k = 0, a_i \leq M)}{p(a_i \leq M)} \tag{15}$$

where b_k and M are respectively any bit in the bit plane $\mathbf{P}_k$ and the largest possible absolute values. The probability $p(a_i \leq M)$ is defined as:

$$p(a_i \leq M) = \sum_{\tilde{y}_i = -M}^{M} p(\tilde{y}_i) \tag{16}$$

It can be shown that the probability of 0 in the bit plane $\mathbf{P}_k$ is given by:

$$p(b_k = 0 | a_i \leq M) = \frac{\sum_{\tilde{y}_i=-M}^{M} p(\tilde{y}_i) \times \delta_{\mathbf{B}_k(a_i),0}}{\sum_{\tilde{y}_i=-M}^{M} p(\tilde{y}_i)} \tag{17}$$

The probability $p_M(b_k = 1)$ is then given by:

$$p(b_k = 1 | a_i \leq M) + p(b_k = 0 | a_i \leq M) = 1 \tag{18}$$

4.2.2. Second method: arithmetic coding with model-based conditional probabilities

Bit plane coding of $\mathbf{P}_k$ with $k < K - 1$ can use the knowledge of bit planes coded before, $\mathbf{P}_{K-1} \ldots \mathbf{P}_{k+1}$. The most significant bit plane (MSB) is coded with model-based initialization of probability tables as in Section 4.2.1.

We define the context for the i^{th} bit in the k^{th} bit plane as the bits on bit planes coded before $\mathbf{P}_k$. Here, for every bit plane expect the MSB ($k < K - 1$), the context $c_k(a_i)$ in $\mathbf{P}_k$ is defined as:

$$c_k(a_i) = \sum_{j=k+1}^{K-1} \mathbf{B}_j(a_i)\, 2^j \quad -M \leq \tilde{y}_i < M \quad \forall k < K \tag{19}$$

The number of contexts in $\mathbf{P}_k$ is 2^{K-k}. It can be shown that the conditional probability of having the 0 for $|\tilde{y}_i| \leq M$ with the context, c_k, is defined as:

$$p\left(b_k = 0 | c_k = c_k(a_i), a_i \leq M\right) = \frac{p\left(b_k = 0, c_k = c_k(a_i) | a_i \leq M\right)}{p\left(c_k = c_k(a_i) | a_i \leq M\right)} \tag{20}$$

We can finally derive this relationship for conditional probability:

$$p\left(b_k = 0 | c_k, a_i \leq M\right) = \frac{\sum_{\tilde{y}_i=-M}^{M} \left[p\left(\tilde{y}_i\right) \times \delta_{\mathbf{B}_k(a_i),0} \times \prod_{j=k+1}^{K-1} \delta_{\mathbf{B}_j(a_i),\mathbf{B}_j(k)} \right]}{\sum_{\tilde{y}_i=-M}^{M} \left[p\left(\tilde{y}_i\right) \times \prod_{j=k+1}^{K-1} \delta_{\mathbf{B}_j(a_i),\mathbf{B}_j(k)} \right]} \tag{21}$$

5. EXPERIMENTAL RESULTS

In this work we used the same experimental setup as in [3]. A database of 24 clean speech samples in French language (6 male and female speakers×4 sentence-pairs) and 16 clean music samples (4 types×4 samples) of 8 seconds is used for quality evaluation. These samples are sampled at 16 kHz, preprocessed by the P.341 filter of ITU-T G.191A and normalized to -26 dB_{ov} using the P.56 speech voltmeter. Two reference coders are selected: ITU-T G.722.1 at 24 and 32 kbit/s and stack-run coding from 16 to 40 kbit/s [3]

5.1. Quality results

WB-PESQ [12] is used to evaluate the quality of the proposed coder and compare it with reference coders. Only clean speech samples are used to compute the average WB-PESQ scores at various bitrate. The bit rate varies from 16 to 40 kbit/s. Our proposed coder is a bitstream scalable coder. The decoder bitrate is equal to or lower than the encoder bitrate.

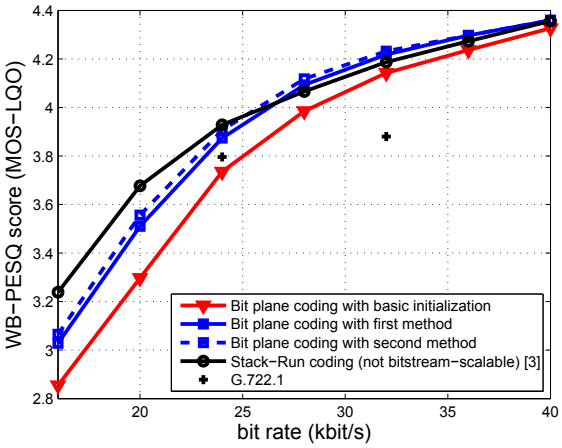

Fig. 4. Average WB-PESQ score (*without* noise injection).

Fig. 4 shows the WB-PESQ scores obtained for the three coders. The bit-plane coding results in Fig . 4 are from one encoding (one bitstream at 40 kbit/s), decoded in a bitstream scalable fashion. The use of model-based probabilities improve coding performance with respect to adaptive arithmetic coding with probabilities initialized to $p(0) = p(1) = 0.5$ (basic initialization). These results suggest that the speech quality of the proposed coder using model-based initialization of symbol probabilities is equivalent to reference coders at high bitrate (0.02 MOS-listening quality objective (LQO) difference) and slightly worse at low bitrate (0.1 MOS-LQO difference). Subjective tests at 32 kbit/s have been conducted: one for speech, another for music. At 32 kbit/s the proposed coder is equivalent to reference coders in both cases (G.722.1 and stack-run coding). Informal listening confirmed the quality difference from 16 to 32 kbit/s between the proposed coder and stack-run coding, predicted by WB-PESQ. Note that WB-PESQ is revelant in this latter case, as the proposed coder and stack-run coder have very close coding structures (only MDCT quantization methods differ). Furthermore we still have to improve the noise injection of the proposed coder at low bitrate in order to compare it with the reference coders.

5.2. Complexity

The algorithmic delay of the proposed embedded coder and stack-run coding is 45 ms (20 ms for the frame, 20 ms for the MDCT and 5 ms for the lookahead), while that of G.722.1 is 40 ms. The computational complexity of G.722.1 is low which is also the case for the proposed embedded coder since rate control is automatically handled by bit plane coding. The memory requirements (in terms of data ROM) for the proposed coder consists mainly of the storage of GMM parameters for LPC quantization and MDCT computation tables.

6. CONCLUSION

In this paper we proposed an embedded speech and audio coder based on generalized Gaussian modeling and bit plane coding. This coder was compared against ITU-T G.722.1 and stack-run audio coding. The generalized Gaussian model allows to estimate efficiently symbol probability in bit planes. This model-based approach brings an improvement of 0.1-0.4 MOS-LQO compared with a baseline bit plane coder. The proposed coder reachs a performance similar to non-embedded coding such as stack-run coding or G.722.1, which is remarkable. Further work will be focused on improving quality at low bit rates to reduce the performance penalty and to handle multiple constraints (e.g. sampling frequency, frame length) of bitstream scalability.

REFERENCES

[1] A. D. Subramaniam and B. D. Rao, "PDF optimized parametric vector quantization of speech line spectral frequencies," *IEEE Trans. Speech and Audio Proc.*, vol. 11, no. 2, pp. 130–142, Mar 2003.

[2] J. Samuelsson, "Waveform quantization of speech using Gaussian mixture models," *Proc. ICASSP*, vol. 1, pp. 165–168, 2004.

[3] M. Oger, S. Ragot, and M. Antonini, "Transform audio coding with arithmetic-coded scalar quantization and model-based bit allocation," *Proc. ICASSP*, vol. 4, pp. 545–548, 2007.

[4] D. Zhao, J. Samuelsson, and M. Nilsson, "GMM-based entropy-constrained vector quantization," *Proc. ICASSP*, vol. 4, pp. 1097–1100, 2007.

[5] S. H. Park and al., "Multi-layer bit-sliced bitrate scalable audio coding," *Presented at the AES 103rd Convention*, vol. Preprint 4520, Aug 1997.

[6] C. Dunn, "Efficient audio coding with fine-grain scalability," *Presented at the AES 111th convention*, vol. Preprint 5492, Sep 2001.

[7] J. Li, "Embedded audio coding (EAC) with implicit auditory masking," *ACM Multimedia 2002*, Dec 2002.

[8] D. S. Taubman and M. W. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards and Practice*, Springer, 2001.

[9] C. Parisot, M. Antonini, and M. Barlaud, "3d scan based wavelet transform and quality control for video coding," *EURASIP*, vol. 1, pp. 521–528, Jan 2003.

[10] I. H. Witten, R. M. Neal, and J. G. Cleary, "Arithmetic coding for data compression," *ACM Communications*, Jun 1987.

[11] G. G. Langdon, "An introduction to arithmetic coding," *IBM J. Res. Develop. 28*, pp. 135–149, Mar 1984.

[12] ITU-T Rec P.862.2, *Wideband extension to Recommendation P.862 for the assessment of wideband telephone networks and speech codecs*, Nov 2005.