Audio Engineering Society

Convention Paper 6805

Presented at the 120th Convention
2006 May 20–23 Paris, France

Stack-Run Audio Coding

Marie Oger[1], Julien Bensa[3], 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*

[3] *France Télécom R&D/TECH/SSTP when this work was done. Now with the European Patent Office (EPO), The Hague, The Netherlands*

Correspondence should be addressed to Marie Oger (marie.oger@francetelecom.com)

ABSTRACT

In this paper we present an application of stack-run entropy coding to audio compression. Stack-run coding represents signed integers and zero run length by adaptive arithmetic coding using a quaternary alphabet (0, 1, +, -). We use this method to encode the scalar quantization indices representing the MDCT spectrum of perceptually weighted wideband audio signals (sampled at 16000 Hz). Noise injection and pre-echo reduction are also used to improve quality. The average quality of the proposed technique is similar to ITU-T G722.1. In addition, we compare the performance of scalar quantization with stack-run coding to the multirate lattice vector quantization of 3GPP AMR-WB+.

1. INTRODUCTION

Audio coding schemes such as MPEG-4 AAC and ITU-T G.722.1 are built upon modified discrete cosine transform (MDCT), scalar quantization of normalized MDCT coefficients and Huffman entropy coding. We propose in this work to apply a different method to represent the MDCT of audio signals: scalar quantization with stack-run coding.

Stack-run coding was introduced in [1] to encode wavelet transformed images with respect to the mean square error criterion. It is not directly applicable to audio signals. In speech and audio coding the quantization has to be performed in a perceptual domain to be efficient. An audio signal can be perceptually weighted in time or frequency domain. We chose in this work to follow a predictive transform coding approach similar to TCX coding [2], TPC coding [3]. Thus perceptual weighting is applied in time domain through a linear-predictive filter.

This paper is organized as follows. We present the principles of stack-run coding in Section 2. Then the proposed stack-run audio coding is described in Section 3. In Section 4, the reference coding method, AMR-WB+ RE_8 quantization, is presented. Exper-

imental results for stack-run audio coding are presented and discussed in Section 5 before concluding in Section 6.

2. STACK-RUN CODING

Stack-run coding [1] is originally a lossless coding method applied to wavelet image coding. In [1], the discrete wavelet transform of an input image is scalar quantized with a dead zone. The resulting integer coefficients are partionned into two groups: sequences of zeros ("runs") and non-zero integers ("stacks").

2.1. Representation of an integer sequence by a quaternary alphabet

A stack is a column of bits with the most significant bit (MSB) at the top and the less significant bit (LSB) at the bottom. This binary representation is unsigned and sign information is considered apart. In each stack the MSB is replaced by "+" if the associate coefficient is positive and "-" if it is negative. Also the binary representation of the absolute value of stack is incremented by one. For example the binary representation of +4 is "+01" instead of "+00" and the binary representation of -8 is "-001" instead of "-000".

The symbol alphabets have the following meanings:

- "0" is used to signify a bit value of 0 in encoding of stack.
- "1" is used for a bit value of 1 in stack, but it is not used for the MSB.
- "+" is used to represent the positive MSB of stack and for a bit value of 1 in representing run lengths.
- "-" is used to represent the negative MSB of stack and for a bit value of 0 in representing run lengths.

The detailed mapping and coding rules are given in [4, 1].

2.2. Mapping example

We take here the mapping example given in [4]. The integer sequence:

$$0, 0, 0, +35, +4, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, -11$$

is mapped into [4]:

$$++00100+10+-+-001-$$

On the other hand, the integer sequence:

$$0, -1, 1, 0, 0, 0, -3, 5, -6, 3, 0, -1, 0, 0, -1, 1$$

is mapped into:

$$+0-0+++00-01+11-00++0--0-0+$$

These examples show that an integer sequence is well compacted with the quaternary alphabet (0, 1, +, -) as soon as a long sequence of runs is present. The stack-run coding is more efficient when there are long runs.

2.3. Adaptive arithmetic coding

The above method of mapping an integer sequence is well adapted to arithmetic coding with a quaternay alphabet. First, the probability tables used in the arithmetic coder can be adaptive. Second, run and stack can be considered separately and therefore two probability tables can be updated independently. The sequences of symbols (0, 1, +, -) are encoded by switching between two different contexts: one for stack, another for run symbols. This context switching is illustrated in Table 1 which shows the alternance of stack and run for the integer sequence example of [4].

Table 1: Alternance between stack and run in the mapping example of [4].

Run	++			-+-	
Stack		00100+	10+		001-

3. STACK-RUN AUDIO CODING SCHEME

3.1. Encoder

The proposed stack-run audio encoder is presented in Figure 1. This proposed scheme employs a linear-predictive perceptual weighting filter followed by MDCT coding. It can therefore be considered as a form of predictive transform coding. The input sampling frequency is 16000 Hz, while the frame length

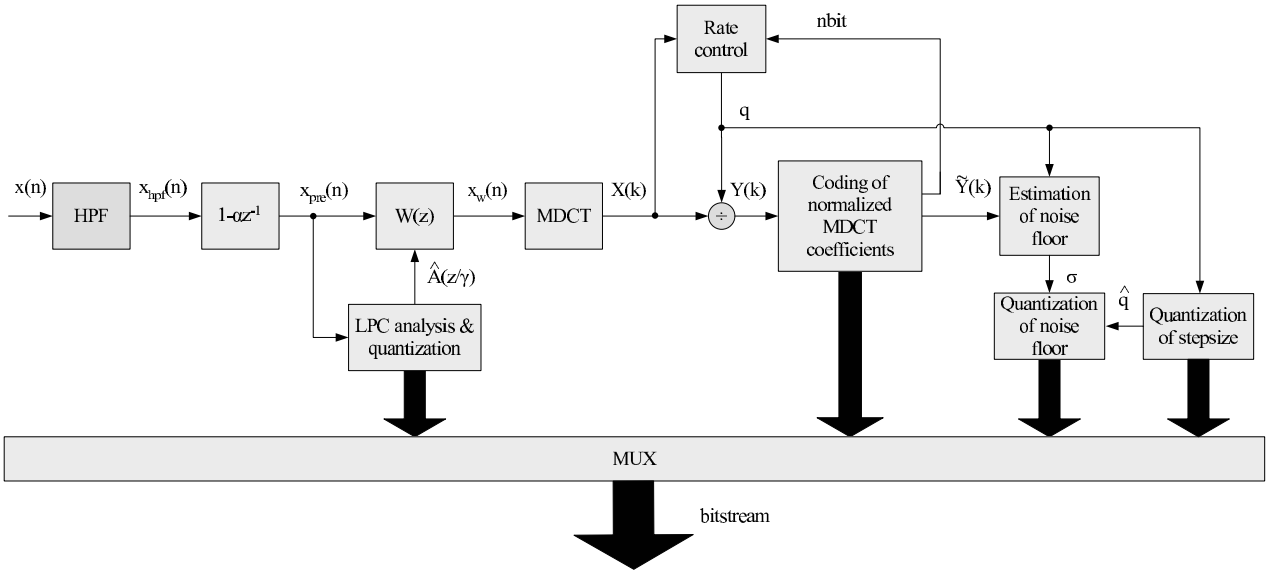

Fig. 1: Stack-run audio coding scheme.

is 20 ms with a lookahead of 25 ms. The effective bandwith of the input signal is considered to be 50-7000 Hz.

3.1.1. High-pass prefilter and preemphasis

A high-pass filter (HPF) is applied on the input signal $x(n)$ in order to remove the frequency component under 50 Hz. The resulting signal $x_{hpf}(n)$ is then preemphasized by $1 - \alpha z^{-1}$ with $\alpha = 0.75$.

3.1.2. LPC analysis & quantization

An 18th order LPC analysis is performed on the preemphasized signal $x_{pre}(n)$. The resulting LPC coefficients are quantized with 40 bits using a parametric method based on Gaussian Mixture Models (GMM) [5, 6].

LPC analysis The filter used for the LPC analysis is defined as:

$$A(z) = 1 + \sum_{i=1}^{n} a_i z^{-i} \qquad (1)$$

where a_i and n are respectively the LPC coefficients and the LPC order, here $n = 18$ [7]. The autocorrelation method with asymetric Hamming-cosine window of 30 ms and lag windowing is used. The LPC coefficients are computed with the Levinson-Durbin recursion and transformed into Line Spectrum Frequencies (LSF) for quantization. The LSF parameter actually correspond here to the modified filter $A(z/\gamma)$, with $\gamma = 0.92$, instead of $A(z)$.

Approximation of the LSF probability density function by a Gaussian mixture model We follow here the notations of [8]. The probability density function (pdf) of LSF vectors ω in dimension n can be modeled [5] by a Gaussian mixture model of order M given by

$$f(\omega|\Theta) = \sum_{i=1}^{M} \rho_i \, f_i(\omega|\theta_i), \qquad (2)$$

$$\text{where } f_i(\omega|\theta_i) = \frac{e^{-\frac{1}{2}(\omega-\mu_i)^T \mathbf{C}_i^{-1}(\omega-\mu_i)}}{\sqrt{(2\pi)^n |det(\mathbf{C}_i)|}}, \qquad (3)$$

with the following constraints: $\rho_i > 0$ and $\sum_{i=1}^{M} \rho_i = 1$. The set of GMM parameters is given by:

$$\begin{aligned} \Theta \;=\; & \{\rho_1, \cdots, \rho_M, \theta_1, \cdots, \theta_M\} \\ & \{\rho_1, \cdots, \rho_M, \mu_1, \cdots, \mu_M, \mathbf{C}_1, \cdots, \mathbf{C}_M\} \end{aligned} \qquad (4)$$

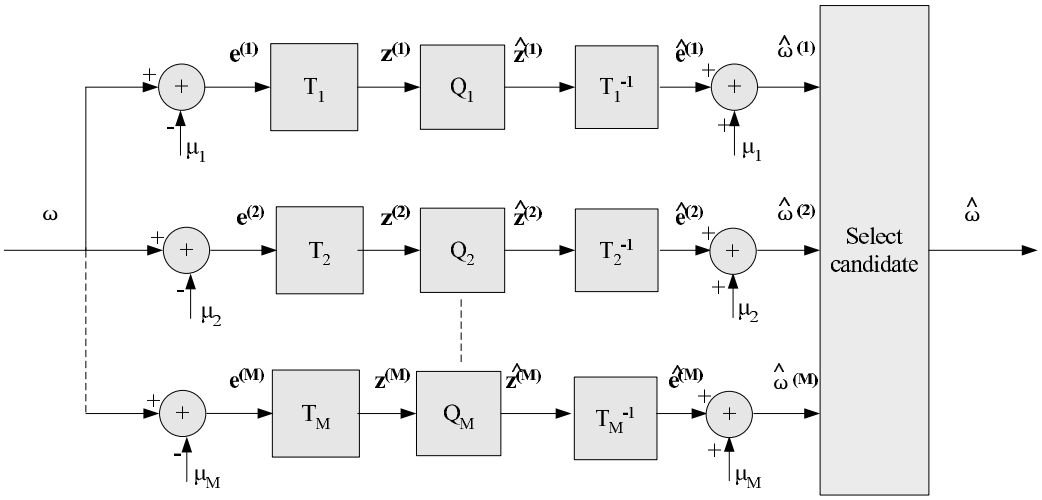

Fig. 2: Mean-removed KLT coding.

where ρ_i, μ_i and $\mathbf{C}_i$ are respectively the weight (a priori probability), the mean vector and the covariance matrix of the i-th GMM component. For a given source database, Θ is usually estimated using the E-M algorithm [9]. In this work we use a GMM order $M = 4$.

LSF quantization based on mean-removed KLT coding The LSF parameters are quantized using a GMM-based scheme illustrated in Figure 2. For an input LSF vector ω, the quantized LSF vector $\hat{\omega}$ is selected among M candidates $\widehat{\omega}^{(i)}$, with $i = 1, \cdots, M$, by minimizing a distortion criterion:

$$\hat{\omega} = \hat{\omega}^{(j)} \text{ where } j = \arg \min_{i=1,\cdots,M} d(\omega, \widehat{\omega}^{(i)}). \quad (5)$$

with M the order of the GMM.

The selection criterion d is a weighted Euclidean distance [10]. The candidate $\hat{\omega}^{(i)}$ is the representative of ω in the i-th GMM component (or cluster). The candidates $\hat{\omega}^{(i)}$ are computed as in [5] by mean-removed Karhunen-Loeve transform (KLT) coding using the parameters μ_i and $\mathbf{C}_i$ of the i-th GMM component. The transform matrix $\mathbf{T}_i$ is defined as:

$$\mathbf{T}_i = \mathbf{K}_i \ diag\left(\frac{1}{\sigma_{i1}}, \cdots, \frac{1}{\sigma_{in}}\right) \quad (6)$$

where $\mathbf{K}_i$ is the KLT matrix and σ_{ij} are the normalization factors derived from the eigenvalue decomposition of covariance matrices $\mathbf{C}_i$ [5]:

$$\mathbf{C}_i = \mathbf{K}_i \ diag(\sigma_{i1}^2, \cdots, \sigma_{in}^2) \ \mathbf{K}_i^T \quad (7)$$

the terms $\sigma_{i1}^2 \geq \cdots \geq \sigma_{in}^2$ are the eigenvalues of $\mathbf{C}_i$ and the matrix $\mathbf{K}_i$ comprises the eigenvectors of $\mathbf{C}_i$. The quantization Q_i of the source $\mathbf{z}^{(i)}$ is implemented by a corresponds to model-based Lloyd-Max quantization.

Allocation of scalar quantization levels The number of bit B_{LSF} allocated to LPC quantization is distributed among the M clusters by the algorithm presented in [5]. Inside each cluster $i = 1, \ldots, M$ the quantization Q_i is a scalar model-based Lloyd-Max quantization. The number of levels L_{ij} allocated to each element of $\mathbf{z}^{(i)}$ is optimized by a greedy allocation algorithm [6].

Model-based Lloyd-Max quantization The source $\mathbf{z} = (z_1, \cdots, z_n)$ is encoded by Lloyd-Max quantization [11] optimized for a Gaussian source model.

A model-based approach is used to circumvent the costly training of stochastic codebooks using a database. The decision thresholds t_{ij}, and the reconstruction levels s_{ij} of the quantizer are found by the

following iterative process until convergence of s_{ij}:

$$t_{ij} = \frac{1}{2}\left(s_{ij} + s_{i-1,j}\right) \qquad i = 2, \ldots, L_{ij} \tag{8}$$

with $t_{i0} = -\infty$ and $t_{i,L_{ij}+1} = +\infty$

$$s_{ij} = \frac{\int_{t_{ij}}^{t_{i,j+1}} z\; g(z)\, dz}{\int_{t_{ij}}^{t_{i,j+1}} g(z)\, dz} \qquad i = 1, \ldots, L_{ij} \tag{9}$$

where L_{ij} and $g(z)$ are respectively the allocated number of levels and the probability density function for a Gaussian source model defined as:

$$g(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-z^2}{2\sigma^2}} \tag{10}$$

3.1.3. Perceptual weighting filter

The perceptual weighting filter $W(z)$ in Figure 1 is defined as:

$$W(z) = \frac{\hat{A}(z/\gamma)}{1 - \beta z^{-1}} \tag{11}$$

where $\beta = 0.75$ is the tilt parameter and $\gamma = 0.92$. The coefficients of $W(z)$ are updated every 5 ms by interpolating the LSF parameters.

3.1.4. MDCT analysis

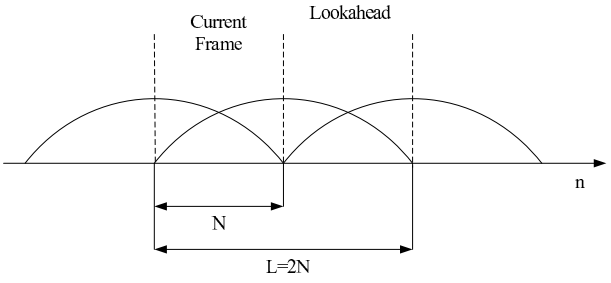

Fig. 3: MDCT windowing.

For the given signal $x_w(n)$ of $L = 2N$ samples with N samples for the current frame and N samples in the lookahead, the MDCT computes a spectrum $X(k)$ of N coefficients. We use $N = 320$ samples (20 ms). Since the sampling frequency is 16000 Hz, each MDCT coefficient corresponds to a band of 25 Hz. The MDCT is defined as follows [12, 13]:

$$X(k) = \frac{\sqrt{2}}{L} \sum_{n=0}^{L-1} sin\left(\frac{\pi}{L}(n+0.5)\right) \times$$
$$cos\left(\frac{\pi}{N}(n + N/2 + 0.5)(k+0.5)\right) x_w(n) \tag{12}$$

where $k = 0, \ldots, N-1$. The related buffering and windowing are depicted in Figure 3. The MDCT is implemented using the fast algorithm of [14] which is based on a complex FFT.

3.1.5. Coding of normalized MDCT coefficients

Only the first 280 coefficients corresponding to the 0-7000 Hz band are coded; the last 40 coefficients are discarded. The MDCT coefficients $X(k)$ are normalized by a step size q and the resulting spectrum $Y(k)$ is encoded by scalar quantization with stack-run coding. For a given spectrum $Y(k)$ the spectrum $\tilde{Y}(k)$ after scalar quantization is defined as:

$$\tilde{Y}(k) = \left[\frac{Y(k)}{q}\right] \tag{13}$$

where [.] represents the rounding to the nearest integer and q is the step size.

Then the integer sequence $\tilde{Y}(k)$, $k = 0, \ldots, 279$ is encoded by stack-run coding presented in Section 2.

3.1.6. Rate control and step size estimation

The rate control consists of finding the appropriate step size q that the number of bits *nbit* used for stack-run coding matches the allocated bit budget. A bisection search algorithm is implemented in a way similar to the rate control of AMR-WB+ coding. See Section 4.2 for more details.

3.1.7. Noise floor estimation

The spectrum $\tilde{Y}(k)$ is divided in 35 subbands of 8 coefficients. The noise floor σ is estimated as the global r.m.s. of all subbands of $\tilde{Y}(k)$ above 1 kHz that are quantized to zero.

3.1.8. Quantization of step size and noise floor

The step size is quantized in log domain (with steps of 0.71 dB) with 7 bits. For a given step size q the quantized step size $\hat{q}$ is given by

$$\hat{q} = 10^{(i_q - 44)/28}, \tag{14}$$

where

$$i_q = 28\,[\,log_{10}(q)\,], \tag{15}$$

and [.] represents the rounding to the nearest integer and i_q is restricted to $-44 \leq i_q < 83$.

The noise floor is quantized in linear domain with 3 bits relatively to $\hat{q}$. For a given noise floor σ the quantized noise floor $\hat{\sigma}$ is given by:

$$\hat{\sigma} = \frac{\hat{q}}{10}\left(8 - i_\sigma\right), \tag{16}$$

where

$$i_\sigma = \left\lfloor 8 - \frac{10\sigma}{\hat{q}} \right\rfloor \tag{17}$$

and i_σ is restricted to $0 \leq i_\sigma < 7$.

3.2. Bit allocation

The bit allocation to the parameters of the stack-run audio coder is described in Table 2 where B_{tot} is the total number of bits per frame. For instance at 24 kbit/s , $B_{tot} = 480$ bits.

Table 2: Bit allocation for the coding scheme

Parameter	Number of bits
LSF ω	$B_{LSF} = 40$
Step size q	7
Noise floor σ	3
Stack-run coding	B_{tot}-50
Total	B_{tot}

3.3. Decoder

The stack-run audio decoder is presented in Figure 4. The parameters are decoded to obtain $\hat{\omega}$, $\hat{q}$, $\hat{\sigma}$. The quantized LSF $\hat{\omega}$ are interpolated in every 5 ms and converted to LPC coefficients. To improve quality two methods are applied at the decoder: noise injection and pre-echo reduction.

3.3.1. Noise injection and spectrum denormalization

The spectrum $\tilde{Y}(k)$ reconstructed after stack-run decoding is divided into 35 subbands of 8 coefficients. A noise with amplitude $\hat{\sigma}/\hat{q}$ and random signs is injected in all subbands above 1 kHz of $\tilde{Y}(k)$ that are decoded to zero. The reconstructed spectrum $\hat{X}(k)$ is then given by:

$$\hat{X}(k) = \hat{q}\ \hat{Y}(k) \tag{18}$$

where $\hat{Y}(k)$ is the reconstructed $\tilde{Y}(k)$ added with noise.

3.3.2. Inverse MDCT and overlap-add with pre-echo reduction

The spectrum $\hat{X}(k)$ is transformed in time domain using the inverse MDCT and overlap-add algorithm described in [14]. A pre-echo reduction is applied on the resulting signal. Pre-echos are detected by comparing the energies of the last portion of the previous MDCT window and the last portion of the current MDCT window. If these energies are significantly different, the shape of the first portion of the current MDCT window is modified.

3.3.3. Inverse perceptual filtering

The inverse perceptual filter $W(z)^{-1}$ is defined as:

$$W(z)^{-1} = \frac{1 - \beta z^{-1}}{\hat{A}(z/\gamma)} \tag{19}$$

The role of $W(z)^{-1}$ is to shape the coding noise introduced in the MDCT domain. The response of $W(z)^{-1}$ is similar to a short-term masking curve. The coefficients of $W(z)^{-1}$ are updated every 5 ms by LSF interpolation.

3.3.4. Inverse preemphasis filtering

The inverse preemphasis filter $1/\left(1 - \alpha z^{-1}\right)$ is applied on the signal $\tilde{x}(n)$ to find the synthesis $\hat{x}(n)$.

4. REFERENCE CODING METHOD: AMR-WB+ $\mathbf{RE_8}$ QUANTIZATION

The stack-run audio coder shown in Figures 1 and 4 is similar to the TCX coder of 3GPP AMR-WB+ [15]. The main difference lies in the type of transform and the quantization of normalized transform coefficients. The audio coder proposed here is based on stack-run coding of MDCT coefficients, while the TCX coder of [15] represents FFT coefficients by multirate RE_8 (lattice) vector quantization. This similarity is exploited in this work to evaluate the performance of stack-run coding using the multirate RE_8 vector quantization as a yardstick. In case of RE_8 quantization, the proposed audio coder is modified by replacing stack-run coding by multirate RE_8 vector quantization and the rate control is slightly adapted. We provide here for the sake of completeness a short description of this reference coding method. Note that in this work we do not use

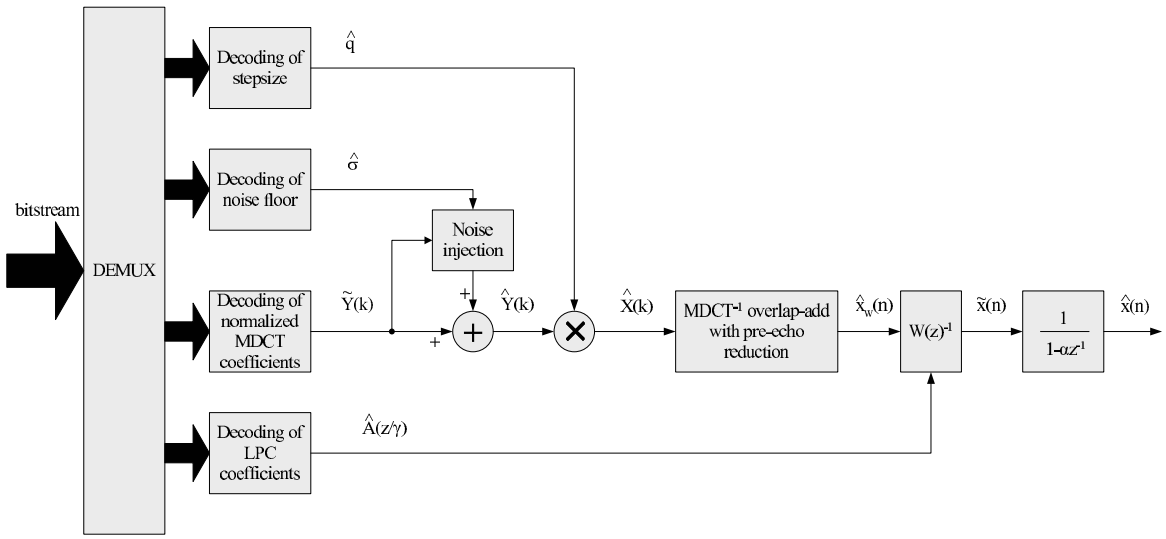

Fig. 4: Stack-run audio decoding scheme.

the spectrum pre- and de-shaping (a.k.a. adaptive low-frequency pre-/de-emphasis) described in [15].

4.1. Multirate RE_8 quantization

To be able to use multirate RE_8 vector quantization, the spectrum $Y(k)$ in Figure 1 is divided into 35 subbands of 8 MDCT coefficients. Each subband $j = 1, \ldots, 35$ is encoded by the self-scalable RE_8 quantizer described in [16]. We recall below the definition of the (infinite) RE_8 lattice:

$$RE_8 = 2D_8 \cup \{2D_8 + (1, 1, \cdots, 1)\} \qquad (20)$$

where

$$D_8 = \left\{(y_1, \ldots, y_8) \in \mathbb{Z}^8 | y_1 + \ldots + y_8 \ \ even\right\} \quad (21)$$

The lattice RE_8 comprises all vectors $\mathbf{y} = (y_1, \ldots, y_8)$ of dimension 8 which verify the following properties:

- The elements $y_{i=1,\ldots,8}$ are integers
- The sum $y_1 + \ldots + y_8$ is a multiple of 4
- The elements y_i are either all even or all odd

After self-scalable RE_8 quantization, the j-th subband of $Y(k)$ is represented by a integer codebook number n_j restricted in $\{0, 2, 3, 4, 5, \cdots\}$ and an index I_j of $4n_j$ bits. Unary coding is used to map the codebook number n_j into binary format using the following mapping rule:

$$\begin{array}{ccc} 0 & \rightarrow & 0 \\ 2 & \rightarrow & 10 \\ 3 & \rightarrow & 110 \\ 4 & \rightarrow & 1110 \\ & \vdots & \end{array}$$

As a result in the j-th subband multirate RE_8 quantization consumes 1 bit if $n_j = 0$ and $5n_j$ bits otherwise.

4.2. Rate control and step size estimation

The rate control implemented in the TCX coder of 3GPP AMR-WB+ consists of estimating the optimal step size q to match a given target bit budget B_{RE_8}. Here $B_{RE_8} = B_{tot} - 50$ bits. The underlying algorithm is based on the principle of reverse water-filling [17]. The algorithm starts by estimating the codebook number $\tilde{n}_j$ of each subband based on the subband energy E_j assuming a step size $q = 1$:

$$\tilde{n}_j = \frac{1}{2} log_2(E_j/\varepsilon), \qquad (22)$$

where $\varepsilon = 2$ is a calibration factor. Then a bisection search algorithm is used to find the optimal "water

level" λ so that

$$\sum_j max(0, 5(\tilde{n}_j - \lambda)) \simeq B_{RE8}. \quad (23)$$

Finally the step size is estimated as:

$$q = 2^{\lambda/2}. \quad (24)$$

Because the rate control in AMR-WB+ is based on *estimated* bit consumptions $max(0, 5(\tilde{n}_j - \lambda))$, the rate control may also set to zero some selected subbands prior to multiplexing, so as to verify the bit budget constraint.

5. EXPERIMENTAL RESULTS FOR STACK-RUN AUDIO CODING

5.1. Experimental setup

Table 3: Samples used for subjective and objective quality evaluation.

Sample	Description	Length (frames)
t5	instrumental (harpsichord)	454
t7	female speech (French)	636
t9	song (Tracy Chapman)	691
t22	male speech (German)	584

A wideband speech and music database of 22 min in which silence segments are removed is used to train LPC quantization. This database is preprocessed by a P341 filter of ITU-T G.191A (Software Tool Library). A similar test database is also constructed consisting of 15 min of audio material. In addition several wideband test samples have been selected to conduct objective and subjective quality evaluation. These samples, which are preprocessed by the P341 filter, are described in Table 3.

5.2. Spectral distortion for LPC quantization

The performance of the LSF quantization is evaluated with the spectral distortion [7] defined as :

$$SD = \sqrt{\frac{1}{2\pi}\int_{-\pi}^{\pi}\left[10\log_{10}\frac{1}{|A(\frac{e^{j\omega}}{\gamma})|^2} - 10\log_{10}\frac{1}{|\hat{A}(\frac{e^{j\omega}}{\gamma})|^2}\right]^2 d\omega}$$

where $1/|A(e^{j\omega}/\gamma)|^2$ and $1/|\hat{A}(e^{j\omega}/\gamma)|^2$ are respectively the original LPC spectrum and the quantized LPC spectrum. The spectral distortion (SD) for the quantization of $A(z/\gamma)$ with 40 bits is presented in Table 4. The average SD is around 1.20 dB and the amount of outliers is limited. This LPC quantization is not exactly transparent but the related spectral distortion is quite acceptable.

Table 4: Spectral distortion for the quantization of $A(z/\gamma)$ with 40 bits.

avg. SD (dB)	$SD \geq 2$ dB (%)	$SD \geq 4$ dB (%)
1.20	15.17	1.04

5.3. Objective quality results

The objective criterion used to compare stack-run coding and RE_8 vector quantization is the segmental signal-to-noise ratio in the weighted signal domain ($segSNR_w$) between $x_w(k)$ and $\hat{x}_w(k)$. This criterion allows to evaluate the intrinsic MDCT coding quality without the influence of LPC quantization.

The $segSNR_w$ is defined as:

$$segSNR_w = \frac{1}{N_{seg}} \sum_{i_{seg}=0}^{N_{seg}-1} SNR_{i_{seg}} \quad (25)$$

where

$$SNR_{i_{seg}} = 10\log_{10}\left[\frac{\sum_{seg.\, i_{seg}} x_w^2(n)}{\sum_{seg.\, i_{seg}} (x_w(n) - \hat{x}_w(n))^2}\right] \quad (26)$$

The segment length used to compute the $segSNR_w$ is 80 samples (5 ms). Note that noise injection and pre-echo reduction are disabled when evaluating the $segSNR_w$. Indeed, the $segSNR_w$ is used to compare the waveform-matching performance of stack-run coding and RE_8 vector quantization, while noise injection and pre-echo reduction change the waveform and would introduce a biais in the $segSNR_w$ evaluation.

Figures 5 and 6 show the $segSNR_w$ for the tested music and clean speech sequences. The bit rate goes from 16 to 48 kbit/s with a step of 8 kbit/s. At low

bit rates (around 16 kbit/s) we can see that for music the stack-run coding is better than or equivalent to RE_8 vector quantization. For clean speech stack-run coding is equivalent or inferior to RE_8 vector quantization. These results *at low bit rates* are well in line with subjective listening. They can be explained by the fact that there is usually more harmonic structure in music which makes stack-run coding more efficient. Indeed stack-run coding can be viewed as a sophisticated combination of run-length coding and adaptive arithmetic coding. The presence of long sequences of zeros (runs) will improve the performance of stack-run coding.

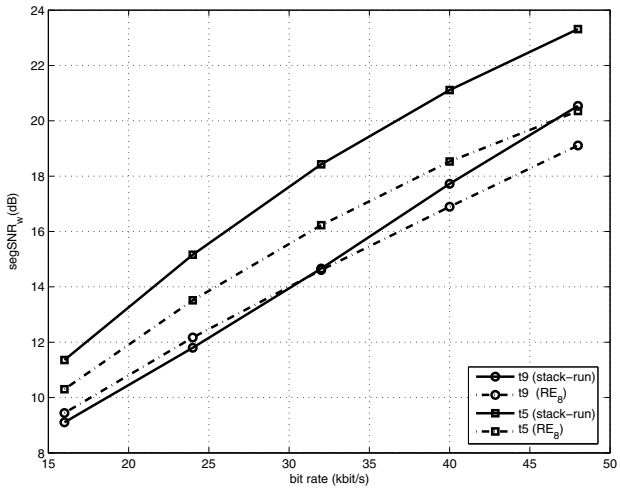

Fig. 5: $segSNR_w$ for music.

The $segSNR_w$ criterion is relevant mostly at low bit rates. At 32 kbit/s and above the actual perceptual difference between stack-run coding and RE_8 vector quantization is very small. For this range of bit rates the performance differences shown in Figures 5 and 6 are not perceptually significant.

5.4. Subjective quality

Informal expert subjective tests at 16, 24 and 32 kbit/s have been conducted with the test samples presented in Table 3. Results show that the quality of stack-run coding and RE_8 vector quantization at 16 kbit/s depend quite a lot on the tested sample. In general, stack-run coding is equivalent to or better than the RE_8 quantization for music and slightly worse for speech. At 24 and 32 kbit/s the quality of

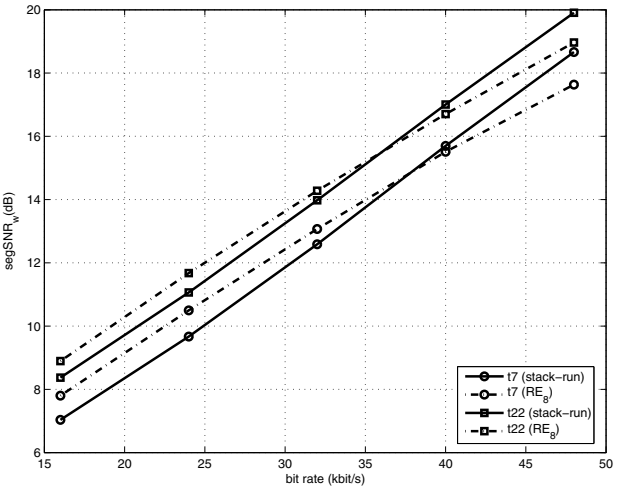

Fig. 6: $segSNR_w$ for speech.

the two methods is comparable. Furthermore informal tests showed that the quality of the proposed stack-run audio coder at 24 and 32 kbit/s is similar to ITU-T G722.1 for music and slightly worse for speech. Formal MUSHRA tests are ongoing to confirm these conclusions.

6. CONCLUSION

In this paper we proposed a new audio coding scheme based on scalar quantization with stack-run coding for wideband signals sampled at 16000 Hz. We chose to apply a perceptual weighting in time domain. However, the coding principle would be the same with a perceptual weighting in frequency domain. The current experimental results are promising. In particular, the performance of stack-run coding for most music signals is equivalent or better than AMR-WB+ RE_8 quantization and ITU-T G722.1. Current work is focused on improving quality in particular on speech signals.

7. REFERENCES

[1] M. J. Tsai, J. D. Villasenor, and F. Chen, "Stack-run image coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 5, pp. 519–521, Oct. 1996.

[2] R. Lefebvre, R. Salami, C. Laflamme, and J. P. Adoul, "High quality coding of wideband audio signals using transform coded excitation (TCX)," *IEEE Acous. Speech and Signal Proc.*, vol. 1, pp. 193–196, 1994.

[3] J. H. Chen and D. Wang, "Transform predictive coding of wideband speech signals," *IEEE Acous. Speech and Signal Proc.*, vol. 1, pp. 275–278, 1996.

[4] M. J. Tsai, J. D. Villasenor, and F. Chen, "Stack-run coding for low bit rate image communication," *in Proc. ICIP*, Oct. 1996.

[5] 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.

[6] M. Oger, S. Ragot, and R. Lefebvre, "Companded lattice VQ for efficient parametric LPC quantization," *in Proc. Eusipco*, 2004.

[7] K. K. Paliwal and W. B. Kleijn, *Quantization of LPC Parameters*, pp. 433–466, in *Speech Coding and Synthesis*, W.B. Kleijn and K.K. Paliwal eds., Elsevier Science, 1995.

[8] P. Hedelin and J. Skoglund, "Vector quantization based on Gaussian mixture models," *IEEE Trans. Speech and Audio Proc.*, vol. 8, no. 4, pp. 385–401, Jul. 2000.

[9] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *JRSSB*, vol. 39, no. 1, pp. 1–38, 1977.

[10] S. Ragot, H. Lahdili, and R. Lefebvre, "Wideband LSF quantization by generalized Voronoi codes," in *Proc. Eurospeech*, Sep. 2001, pp. 2319–2322.

[11] N. S. Jayant and P. Noll, *Digital Coding of Waveforms*, Prentice-Hall,Inc, 1984.

[12] J. Princen, A. W. Johnston, and A. Bradley, "Subband/transform coding using filter bank designs based on time domain cancellation," *in Proc.ICASSP*, pp. 2161–2164, 1987.

[13] H. S. Malvar, "Lapped transforms for efficient transform/subbband coding," *IEEE Trans. on Acoustics, Speech and Signal Processing, vol. 38, no. 6*, pp. 969–978, 1990.

[14] P. Duhamel, Y. Mahieux, and J. P. Petit, "A fast algorithm for the implementation of filter bank based on time domain aliasing cancellation," *in Proc. ICASSP*, pp. 2209–2212, May 1991.

[15] 3GPP TS 26.290, "Audio codec processing functions; extended adaptive multi-rate - wideband (AMR-WB+) codec; transcoding functions," 2005.

[16] S. Ragot, B. Bessette, and R. Lefebvre, "Low-complexity multi-rate lattice vector quantization with application to wideband TCX speech coding at 32 kbit/s," in *ICASSP*, 2004.

[17] S. Ragot, *New techniques of algebraic vector quantization based on Voronoi coding - Application to AMR-WB+ coding (in French)*, Ph.D. thesis, Department of Electrical and Computer Engineering, University of Sherbrooke, QC, Canada, May 2003.