# Union of MDCT Bases for Audio Coding

Emmanuel Ravelli, *Student Member, IEEE*, Gaël Richard, *Senior Member, IEEE*, and Laurent Daudet, *Member, IEEE*

***Abstract*—This paper investigates the use of sparse overcomplete decompositions for audio coding. Audio signals are decomposed over a redundant union of modified discrete cosine transform (MDCT) bases having eight different scales. This approach produces a sparser decomposition than the traditional MDCT-based orthogonal transform and allows better coding efficiency at low bitrates. Contrary to state-of-the-art low bitrate coders, which are based on pure parametric or hybrid representations, our approach is able to provide transparency. Moreover, we use a bitplane encoding approach, which provides a fine-grain scalable coder that can seamlessly operate from very low bitrates up to transparency. Objective evaluation, as well as listening tests, show that the performance of our coder is significantly better than a state-of-the-art transform coder at very low bitrates and has similar performance at high bitrates. We provide a link to test soundfiles and source code to allow better evaluation and reproducibility of the results.**

***Index Terms*—Audio coding, matching pursuit, scalable coding, signal representations, sparse representations.**

## I. Introduction

LOSSY audio coding removes statistical redundancy and perceptual irrelevancy in an audio signal to reduce the overall bitrate. Statistical redundancy is reduced by using a sparse signal representation such that the energy of the signal is concentrated in a few coefficients or parameters. Perceptual irrelevancy is exploited in audio coding by coarsely quantizing, or even removing, components that are imperceptible to the human auditory system. In this paper, we focus on the first part: we propose a new signal representation method for lossy audio coding.

When transparency or near-transparency is required, state-of-the-art audio coders are mostly transform-based and generally use the modified discrete cosine transform (MDCT). One example of such a coder is MPEG-2/4 Advanced Audio Coding (AAC) [1], [2] which is able to encode general audio at 64 kb/s per channel with near-transparent quality [3], [4]. However, MDCT-based coders are known to introduce severe artifacts at lower bitrates (see, e.g., [5]) even if numerous approaches have been proposed to reduce them (see, e.g., [6] for an efficient statistical quantization scheme). In this range, MDCT-based coders are nowadays outperformed by methods that use alternate signal representation based on parametric modeling. One of the most illustrative examples of this is MPEG-4 SinuSoidal Coding (SSC) [7], [8] which is based on a sine+transients+noise model of the signal. Formal verification tests [9] show that SSC outperforms AAC at 24 kb/s per channel. Finally, so-called hybrid coders combine transform and parametric modeling, such as MPEG-4 High Efficiency AAC (HE-AAC) [10], [11] and 3GPP AMR-WB+ [12]. HE-AAC uses an MDCT to model the low half of the spectrum and a parametric approach to model the high-frequency components. AMR-WB+ combines several transform and linear prediction techniques. These hybrid approaches also perform better than AAC at low bitrates [12], [13]. However, pure parametric or hybrid signal representation methods model only a subspace of the audio signals and thus are not able to provide transparent quality, even at high bitrates.

In this paper, we propose a new signal representation based on a union of a number of MDCT bases (typically eight) with different scales. As opposed to existing methods, this signal representation method is able to obtain transparency at high bitrates while giving better results than a transform-based approach at low bitrates, provided that the signal is sufficiently sparse. This best-of-both-worlds approach between parametric and transform coding results however in a very significant increase in the computational cost for encoding, which hinders on-the-fly encoding for communications but is acceptable for non-real-time coding applications. Our technique of encoding coefficients provides a fully embedded bitstream resulting in a scalable coder ranging from very low bitrates to transparency. This scalability property, although not the main motivation of our work, could be useful for applications such as transmission of audio over variable bandwidth networks or transcoding of audio files.

Our approach is related to and different from existing methods in several ways. First, it is related to transform coding and could be seen as a generalization of the transform approach since it is based on a simultaneous use of a union of MDCT bases. This allows us to use efficient scalable encoding techniques used in transform coding [14]–[16], while producing a sparser decomposition than the transform approach, and allows better coding efficiency at low bitrates. Second, our approach is also related to parametric coding. We model the signal as a sum of multiscale sinusoidal atoms which is closely related to multiresolution sinusoidal modeling [17]–[19], a technique used, e.g., in SSC. In parametric audio coding, the sinusoidal model tries to match the sinusoidal content of the signal as closely as possible, which is done using complex sinusoidal atoms with a precise estimate of amplitude, frequency, and

Manuscript received December 21, 2007; revised July 02, 2008. Current version published October 17, 2008. This work was supported in part by the French GIP ANR under contract ANR-06-JCJC-0027-01 DESAM and in part by EPSRC under Grant D000246/1. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Malcolm Slaney.

E. Ravelli and L. Daudet are with the Institut Jean le Rond d'Alembert-LAM, Université Pierre et Marie Curie-Paris 6, 75015 Paris, France (e-mail: ravelli@lam.jussieu.fr; daudet@lam.jussieu.fr).

G. Richard is with the TSI Department, GET-ENST (Télécom Paris), 75014 Paris, France (e-mail: gael.richard@enst.fr).

Digital Object Identifier 10.1109/TASL.2008.2004290

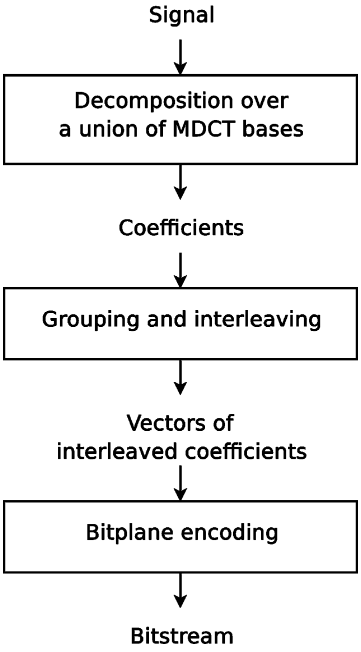

Signal
Decomposition over a union of MDCT bases
Coefficients
Grouping and interleaving
Vectors of interleaved coefficients
Bitplane encoding
Bitstream

Fig. 1. Block diagram of the proposed coder.

phase and a postprocessing stage that builds sinusoidal tracks. In our approach, we extract real sinusoidal atoms with only an amplitude parameter, and so we do not have to transmit any phase parameter. Contrary to the parametric approach, our frequencies are sampled from a limited range (the fast Fourier transform (FFT) size equals the analysis window length), which permits transmission of frequency without requantizing. Moreover, since the sinusoidal decompositions used in parametric audio coding extract, a limited number of sinusoids from the signal and model the residual as noise (plus perhaps transient modeling). Clearly, the sinusoidal decompositions only model a subspace of the signal which limits their performance at high bitrates. Our approach is more general as it models the signal entirely with sinusoidal atoms, which is feasible here at a reasonable coding cost because the set of time–frequency atoms has a limited size, and consequently the cost to encode the index of the selected atoms is not prohibitive. As a consequence, it has the possibility of providing transparency at high bitrates.

The rest of the paper describes the different building blocks of our coder, illustrated in Fig. 1. First, the signal is decomposed over a union of MDCT bases with different scales. This is described in Section II. Then, the resulting coefficients are grouped and interleaved as described in Section III. Vectors of coefficients are obtained and successively coded using a bitplane encoding approach described in Section IV. It is important to note that the proposed configuration clearly separates the signal analysis stage and the coding stage, this configuration is also referred to as out-of-loop quantization or *a posteriori* quantization in the literature. Finally, in Section V, we present objective and subjective evaluations, and conclude (Section VI) with perspectives of future work.

## II. Signal Decomposition

### A. Signal Model

The audio signal is decomposed over a union of MDCT bases with analysis window lengths defined as increasing powers of two. In practice, for a signal sampled at 44.1 kHz, we found that using eight MDCT bases with window length from 128 to 16 384 samples (i.e., from 2.9 to 370 ms) is a good tradeoff between accuracy and size of the decomposition set. Small windows are needed to model very sharp attacks while large windows are useful for modeling long stationary components. Empirical tests have shown that using more MDCT bases with window lengths larger than 16 384 does not significantly improve performance, but increases both the complexity and the coding delay. While the MDCT [20], [21] is a time–frequency orthogonal transform widely used in state-of-the-art transform-based audio coders, no work has been found in the audio coding literature related to the simultaneous use of a union of MDCT bases. However, the union of MDCT bases has already been applied with success in other contexts such as audio denoising [22].

The signal $f \in \mathbb{R}^N$ is decomposed as a weighted sum of functions $g_\gamma \in \mathbb{R}^N$ plus a residual of negligible energy $r$. One may note here that $N$ corresponds to the length of the signal as the analysis is performed on the whole signal; this is different from the common approach in audio coding where the analysis is performed frame-by-frame, $N$ being the frame length. The model is given by

$$f = \sum_{\gamma \in \Gamma} \alpha_\gamma g_\gamma + r \tag{1}$$

where $\alpha_\gamma$ are the weighting coefficients. The set of functions $\mathcal{D} = \{g_\gamma, \gamma \in \Gamma\}$ is called the dictionary and is a union of $M$ MDCT bases (called blocks)

$$\mathcal{D} = \bigcup_{m=0}^{M-1} \mathcal{D}_m \tag{2}$$

with

$$\mathcal{D}_m = \{g_{m,p,k} \mid 0 \le p < P_m, 0 \le k < L_m\} \tag{3}$$

where $m$ is the block index, $p$ is the frame index, $k$ is the frequency index, and $L_m$ is the half of the analysis window length of block $m$

$$L_m = L_0 2^m. \tag{4}$$

$P_m$ is the number of frames of block $m$ and the functions $g$, called atoms are defined as

$$\begin{aligned} &g_{m,p,k}(n) \\ &\quad = w_m(u)\sqrt{\frac{2}{L_m}} \cos\left[\frac{\pi}{L_m}\left(u + \frac{1+L_m}{2}\right)\left(k + \frac{1}{2}\right)\right] \end{aligned} \tag{5}$$

with

$$u = n - pL_m - T_m. \tag{6}$$

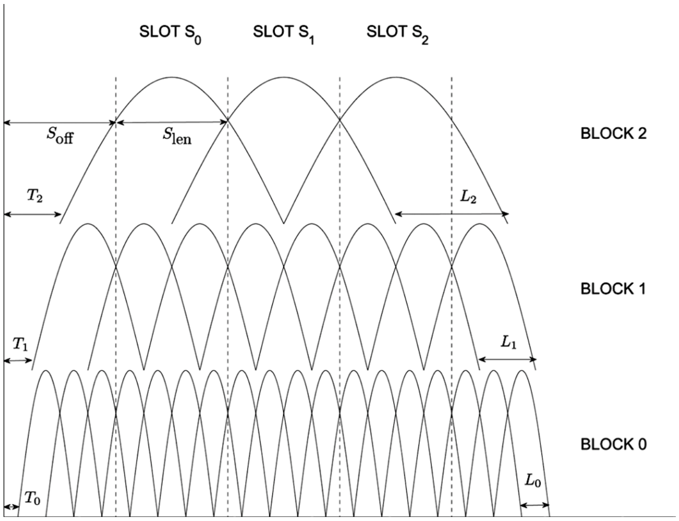


Fig. 2. Analysis windows for $M = 3$ different MDCT window sizes. Dashed vertical lines indicate timeslots.

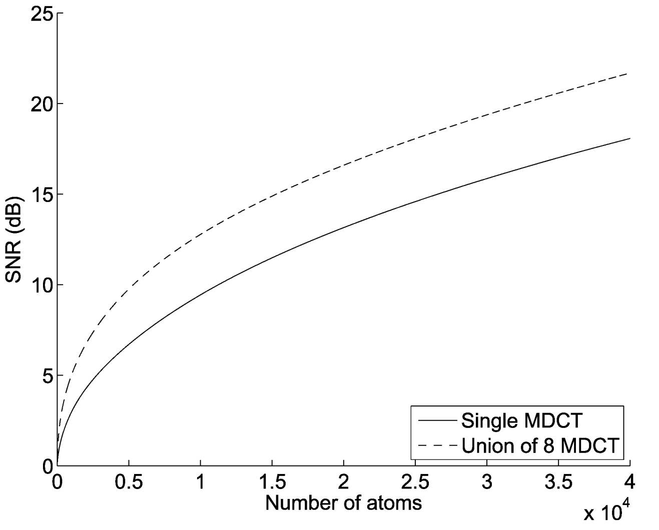


Fig. 3. Mean SNR for five signals given by MP with a single MDCT (window length 2048 samples) and a union of eight MDCT bases (window length from 128 to 16 384 samples).

$T_m$ is a time offset introduced to align the windows of different lengths (see Fig. 2)

$$T_m = \frac{L_m}{2}. \tag{7}$$

The window $w_m(u)$ defined over $u = 0, \ldots, 2L_m - 1$ must satisfy the symmetry and energy-preservation properties [21]. A window having these properties is the sine window which is used in MPEG-1 Layer 3 [23] and MPEG-2/4 AAC [2], [24]

$$w_m(u) = \sin\left[\left(u + \frac{1}{2}\right)\frac{\pi}{2L_m}\right]. \tag{8}$$

Another window is the Kaiser Bessel Derived window (KBD), used in Dolby AC-2/3 [25] and MPEG-4 AAC [2].

### *B. Decomposition Algorithm*

In the case of the orthogonal transform $(M = 1)$, $\mathcal{D}$ forms a basis of $\mathbb{R}^N$ and the atoms $\{g_\gamma\}$ are linearly independent. The decomposition of $f$ over $\mathcal{D}$ is then unique, and is easily obtained by projecting the signal on the atoms

$$\alpha_\gamma = \langle f, g_\gamma \rangle = \sum_n f(n) g_\gamma(n). \tag{9}$$

The dictionary is called overcomplete when $M > 1$: the dimension of $\mathcal{D}$ is greater than the dimension of the signal, and the decomposition of $f$ in $\mathcal{D}$ is not unique anymore. We are looking for a sparse solution, where the signal is represented by a small number of atoms. Finding an optimally sparse solution is a NP-hard problem if the dictionary is unrestricted [26]. Instead, it is possible to find a suboptimal solution using algorithms such as matching pursuit (MP) [27], basis pursuit [28], or FOCUSS [29]. We have chosen MP for its simplicity, flexibility, and rapidity. MP is an iterative descent algorithm which selects the optimal atom at each iteration (see Algorithm 1).

**Algorithm 1** Standard MP

```
input: $f; \mathcal{D} = \{g_\gamma, \gamma \in \Gamma\}$
output: $\alpha_\gamma, \gamma \in \Gamma$
  $r = f$
  $\alpha_\gamma = 0, \forall \gamma \in \Gamma$
  repeat
    $\gamma_{\text{opt}} = \text{argmax}_{\gamma \in \Gamma} |\langle r, g_\gamma \rangle|$
    $c = \langle r, g_{\gamma_{\text{opt}}} \rangle$
    $r = r - c.g_{\gamma_{\text{opt}}}$
    $\alpha_{\gamma_{\text{opt}}} = \alpha_{\gamma_{\text{opt}}} + c$
  until target signal-to-noise ratio (SNR) has been reached.
```

Our experience shows that MP produces a much sparser decomposition when using an overcomplete dictionary $(M > 1)$ instead of an orthogonal dictionary $(M = 1)$. Fig. 3 shows the mean SNR of five signals (with a variety of genres) given by MP with a single MDCT (window length 2048 samples) and a union of eight MDCT bases (window length from 128 to 16 384 samples) as a function of the number of the selected atoms. In the overcomplete case, long stationary parts are efficiently modeled by a small number of large atoms while the attacks are modeled by only a few small scale atoms. This then requires fewer atoms than the single basis case to reach the same SNR, and thus fewer atoms to encode. However, the cost of encoding the index of the selected atoms is greater in the overcomplete case due to the size of the dictionary, but efficient encoding algorithms described in the next sections are able to reduce this cost and still provide better overall coding efficiency than in the transform case at low bitrates.

Standard MP gives good results with most signals; however, it inevitably introduces pre-echo when decomposing signals containing strong attacks. This problem is illustrated in Fig. 4. An extract of a glockenspiel signal is decomposed with MP over

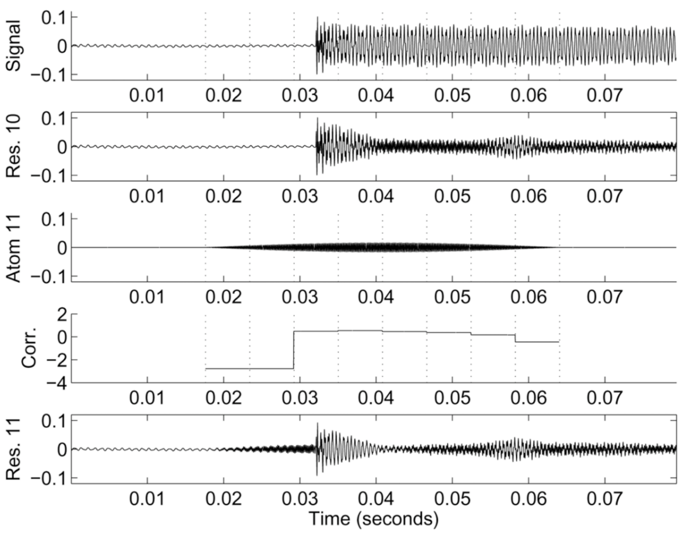


Fig. 4. From top to bottom: the original signal; the residual after ten iterations; the selected atom at iteration 11; the logarithm of the absolute value of the cross-correlation between the un-windowed atom and the original signal on subframes (dotted lines); the residual after 11 iterations.

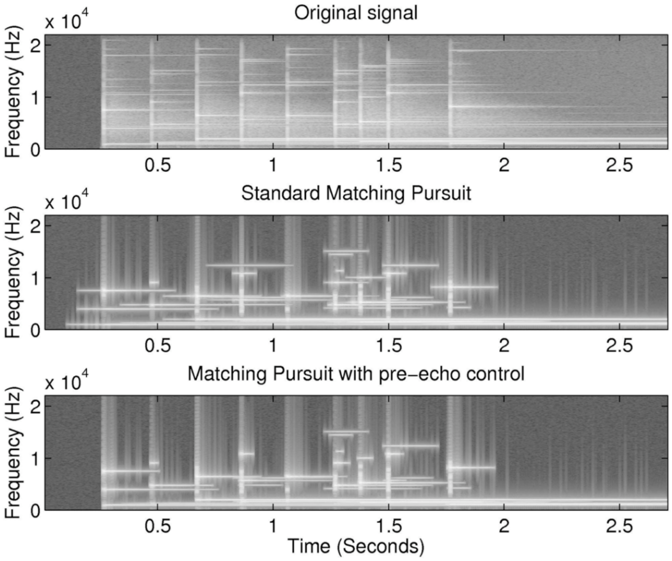


Fig. 5. From top to bottom: Spectrogram of the original signal; spectrogram of the approximated signal with the standard MP and an SNR of 20 dB; spectrogram of the approximated signal with the modified MP and an SNR of 20 dB.

a union of $M = 4$ MDCT bases (and $L_0 = 128$ samples). The second subplot shows the residual at iteration 10. The atom which is best correlated with this residual is in the third subplot. The logarithm of the absolute value of the correlation of the un-windowed function with the original signal on subframes of size 256 is in the fourth subplot. This shows that the beginning of the atom is not correlated with the signal. This results in creating energy just before the transient, which appears in the residual at iteration 11 in the fifth subplot. This energy is removed in further iterations with atoms of low energy. When coding such a decomposition at low bitrate, only the greatest energy atoms are kept which then introduces a pre-echo artifact. Gribonval pointed out this problem in [30] with MP and a Gabor dictionary. He proposed a modified MP algorithm called High-Resolution Matching Pursuit (HRMP) based on the work of Jaggi *et al.* [31]. However, this algorithm was designed for a complex Gabor dictionary and is not adapted to a union of real MDCT bases. Moreover, HRMP significantly increases the computational cost. Alternatively, we propose a simple modification of the MP algorithm that reduces pre-echo artifacts with a small additional computational cost. At each iteration, the function in $\mathcal{D}$ most correlated with the signal is chosen; then the cross-correlation of the un-windowed function with the original signal is computed in subframes of size $W$ (as in Fig. 4); if the dynamic of the cross-correlations (computed as the ratio of the cross-correlation extrema) is greater than a predefined threshold then the function is not selected and removed from the dictionary, otherwise the function is kept and subtracted from the residual (see Algorithm 2). The threshold level was chosen empirically. On the one hand, a high value only removes a few large atoms and gives almost the same result as the standard MP. On the other hand, a low value removes many large atoms but reduces the sparseness of the representation. The value 100 appears to be a satisfactory tradeoff between these two situations. The effectiveness of this pre-echo control is illustrated in Fig. 5 for an extract of a glockenspiel signal.

**Algorithm 2** MP with pre-echo control

```
input: $f$; $\mathcal{D} = \{g_\gamma, \gamma \in \Gamma\}$
output $\alpha_\gamma, \gamma \in \Gamma$
  $r = f$
  $\alpha_\gamma = 0, \forall \gamma \in \Gamma$
  $W_{\min}$ is the shortest window size
  repeat
    loop
      $\gamma_{\text{opt}} = \text{argmax}_{\gamma \in \Gamma} |< r, g_\gamma >|$
      $W = \max\left(g_{\gamma_{\text{opt}}} \text{window size}/8, W_{\min}\right)$
      $S^i f$ = $i$th W-length subframe of $f$
      $S^i g_{\gamma_{\text{opt}}}$ = $i$th W-length subframe of un-windowed. $g_{\gamma_{\text{opt}}}$
      $d_i = \left|\langle S^i f, S^i g_{\gamma_{\text{opt}}} \rangle\right|$
      if $\max_i(d_i) >= thresh * \min_i(d_i)$ then
        $\Gamma = \Gamma \setminus \{\gamma_{\text{opt}}\}$
      else
        exit loop
      end if
    end loop
    $c = \langle r, g_{\gamma_{\text{opt}}} \rangle$
    $r = r - c.g_{\gamma_{\text{opt}}}$
    $\alpha_{\gamma_{\text{opt}}} = \alpha_{\gamma_{\text{opt}}} + c$
  until target SNR has been reached.
```

### C. Implementation

The decomposition algorithm has been implemented in the Matching Pursuit ToolKit [32], a GPL/GNU library developed

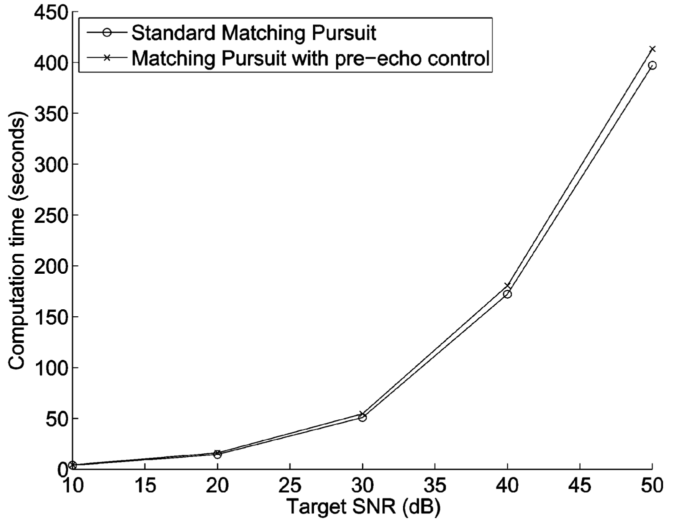


Fig. 6. Computation time to decompose the glockenspiel signal (10-s length) on a 3.0-GHz Pentium 4 computer.

by Krstulovic *et al.* and written in C++. This implementation is to date the fastest available implementation of general MP.

The pre-echo control modification adds a small complexity overhead. Our experiments show that it is around 5% slower than the standard MP. Fig. 6 shows the computation time needed to decompose a 10-s length glockenspiel signal with the eight-scale MDCT dictionary.

## III. Grouping and Interleaving

The decomposition algorithm produces a set of coefficients $\alpha_{m,p,k}$ such that

$$f = \sum_{m,p,k} \alpha_{m,p,k} g_{m,p,k} + r \tag{10}$$

with $r$ a residual of negligible energy.

Contrary to the transform coding case where the analysis is done on a frame-by-frame basis, the decomposition is here performed on the whole signal. As the coefficients are encoded using similar techniques as used in transform coding, it is necessary to group the coefficients in time segments similar to the frames of transform coding. These segments are called here timeslots (see Fig. 2).

In each timeslot, the coefficients are interleaved to produce a vector of coefficients, which is then encoded using the bitplane algorithm described in the next section. The interleaving process puts side by side the coefficients which are close in the time–frequency plane. Consequently, this will most probably cluster the coefficients of high amplitude and leave long series of zeros coefficients. It has been observed that if a coefficient has a high amplitude, then there is a high probability that neighboring coefficients in the time–frequency plane have a high amplitude too. This vector of interleaved coefficients with clusters of high-amplitude coefficients and long series of zeros, is then efficiently encoded using run length-based bitplane algorithms. This section describes how the coefficients are grouped in timeslots and then interleaved to produce a vector of coefficients per timeslot.

### A. Segmentation in Timeslots

The coefficients are grouped in subsets $\mathcal{S}_q$ called timeslots, each of which includes coefficients $\alpha_{m,p,k}$ such that the centers of the corresponding atoms are in the time support of the timeslot

$$qS_{\text{len}} + S_{\text{off}} \leq (p+1)L_m + T_m < (q+1)S_{\text{len}} + S_{\text{off}} \tag{11}$$

where $S_{\text{len}}$ is the timeslot length and $S_{\text{off}}$ is the timeslot offset (position of the first timeslot). The values are chosen such that the timeslots are aligned with the maximum window length block (see Fig. 2)

$$S_{\text{len}} = S_{\text{off}} = L_{M-1}. \tag{12}$$

Using these values, the first $P'_m - 1 = 2^{M-m-1} - 1$ frames of block $m$ are discarded and there are $P'_m$ frames of block $m$ in each timeslot. Timeslots are then simply defined as

$$\mathcal{S}_q = \left\{ \alpha_{m,p,k} \mid \text{floor}\left( \frac{p - (P'_m - 1)}{P'_m} \right) = q \right\} \tag{13}$$

where $\text{floor}(x)$ is the function that rounds $x$ to the nearest integer less than or equal to $x$. Since the first and last frames of each block are discarded in this scheme, it is necessary to pad the signal $f$ with zeros at both sides before the decomposition to avoid any problem at the edges.

To simplify notations in the following, we introduce a new frame index $p'$ such that the frame index starts at 0 in each timeslot. It is defined as

$$p' = \text{mod}\,(p - P'_m + 1, P'_m) \tag{14}$$

where $\text{mod}(x,y)$ is the remainder of the Euclidean division of $x$ by $y$.

### B. Coefficient Interleaving

To be encoded efficiently with the runlength-based bitplane encoder described in the next section, the coefficients are interleaved so that the coefficients that are close in the time–frequency plane are put side by side. The interleaving process produces a vector of coefficients $\mathbf{v} = \{v_i | i = 1, \ldots, ML_{M-1}\}$.

Fig. 7 shows the interleaving process for a simple example where $M = 3$ and $L_0 = 2$. Coefficients are indexed using the notation $XYZ$ where $X = m$, $Y = p'$, and $Z = k$. In 1), each row corresponds to one block and in each block, coefficients are grouped in frames. In 2), the frames of smallest scale (block 0) are interleaved two by two with the immediate upper frame in block 1. This first step produces two new frames of interleaved coefficients. In 3), these two frames are interleaved with the frame of largest scale (block 2) in such a way that the resulting vector has alternatively a coefficient of each block: one of block 2, followed by one of block 1, followed by one of block 0, followed by one of block 2, and so on.

The mapping process between the coefficients of a timeslot $\alpha_{m,p',k}$ and the corresponding vector values $v_i$ may also be formulated as follows. First we define a recursive function $r$ that performs a permutation of the frames

$$r(p', M-1) = p' \tag{15}$$

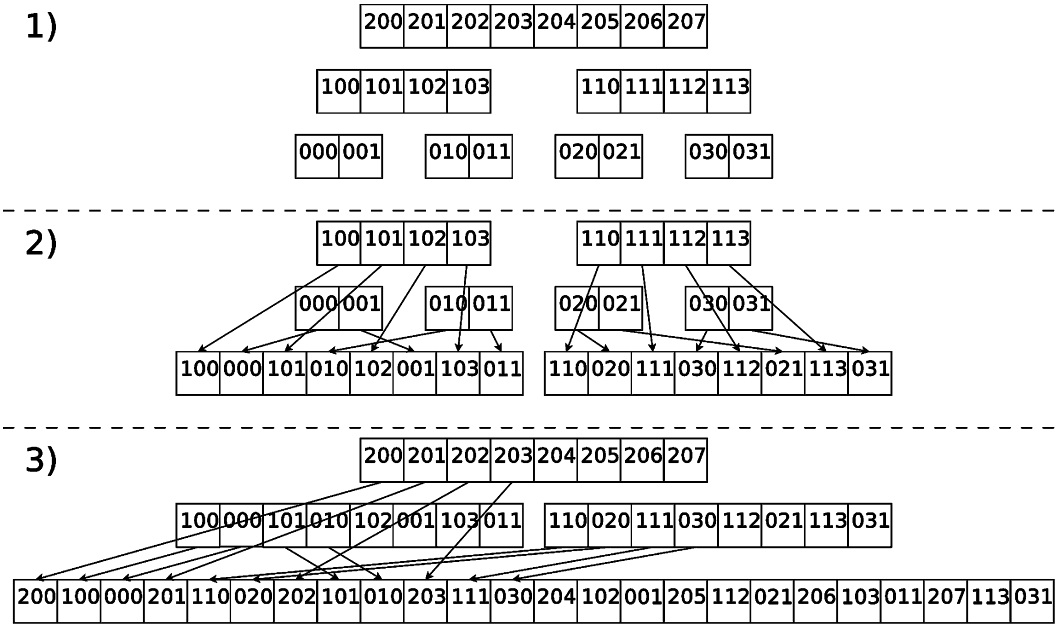


Fig. 7. Interleaving process for $M = 3, L_0 = 2$. 1) The coefficients of the three blocks are given frame by frame and block by block ($XYZ$ is the coefficient of index $m = X, p' = Y$, and $k = Z$). 2) The frames of block 0 are interleaved two by two with the immediate upper frame of block 1. 3) The resulting two frames of interleaved coefficients are interleaved with the unique frame of block 2.

and for $m < M - 1$

$$r(p', m) = \begin{cases} r\left(\frac{p'}{2}, m+1\right), & \text{if } p' \text{ is even} \\ r\left(\frac{p'-1}{2}, m+1\right) + P'_{m+1}, & \text{if } p' \text{ is odd.} \end{cases} \tag{16}$$

Then, values are mapped according to

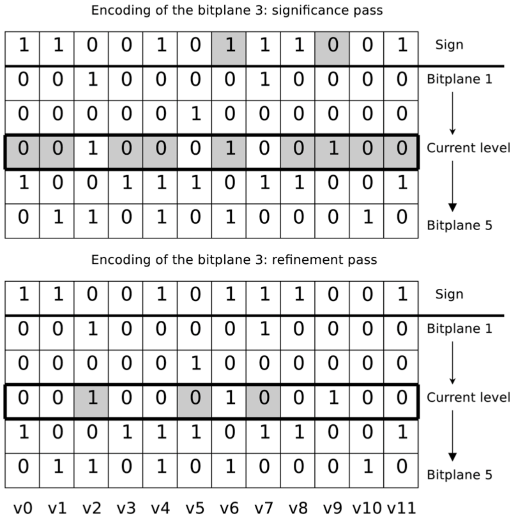
$$v_i = \alpha_{m,r(p',m),k} \tag{17}$$

with

$$i = (kP'_m + p')M + m. \tag{18}$$

## IV. Bitplane Encoding

The vector $\mathbf{v}$ of interleaved coefficients of each timeslot is encoded using bitplane encoding approaches that are similar to those used in transform coding. Though the vector length is much greater ($M$ times) than it would be in the transform coding case, many coefficients are zero and energy is concentrated in fewer coefficients than in the transform coding case. Moreover, the interleaving process often clusters coefficients of high amplitudes and leaves long series of zeros as explained in the previous section. Consequently, runlength-based encoding techniques are very efficient in this case as the long series of zeros are coded using very few bits. The runlength based bitplane encoder we use is based on an approach originally proposed in [33]. We first describe the simple bitplane encoder we used in a previous work [34]. This algorithm is the same as the one used in some wavelet-based image coders [35], [36] and also in a transform-based audio coder [16]. Then, we present a modified version of the bitplane encoder that shapes the quantization noise according to a psychoacoustic model.



Encoding of the bitplane 3: significance pass

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 1 | 0 | **1** | 1 | 1 | **0** | 0 | 1 | Sign |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | Bitplane 1 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | |
| **0** | **0** | 1 | **0** | **0** | 0 | **1** | 0 | **0** | **1** | **0** | **0** | Current level |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | Bitplane 5 |

Encoding of the bitplane 3: refinement pass

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | Sign |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | Bitplane 1 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 0 | 0 | **1** | 0 | 0 | **0** | 1 | **0** | 0 | 1 | 0 | 0 | Current level |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | Bitplane 5 |
| v0 | v1 | v2 | v3 | v4 | v5 | v6 | v7 | v8 | v9 | v10 | v11 | |

Fig. 8. One iteration of the simple bitplane encoder. The two encoding passes are shown. The bits in gray correspond to the transmitted bits.

### A. Simple Bitplane Encoder

The coefficients $v_i$ are first normalized by the amplitude of the coefficient with maximum amplitude $A = \max(abs(v_i))$. The value $A$ is quantized and transmitted. The coefficients are then represented in sign-amplitude form (as shown in Fig. 8, only the five most significant bits are shown). The $j$th most significant bit of the coefficient $v_i$ is given by $b_{i,j} = mod(floor(abs(v_i) * 2^j/A), 2)$. The vector of bits of same significance (or level) $j$ is the $j$th bitplane $B_j = \{b_{i,j}\}$. A coefficient $v_i$ is said to be significant at level $j$ if $abs(v_i)/A \geq 2^{-j}$. The significance of each coefficient is stored in a vector $z_i$ ($z_i = 1$ if the coefficient is significant, $z_i = 0$ if not).

The basic principle of bitplane encoding is to send successively each bitplane starting from the most significant bitplane. This is generally done using a scheme in two passes: the significance pass and the refinement pass. The significant pass transmits the subset of the bitplane $B_j$ corresponding to the $j$th most significant bits of the coefficients that are not yet significant $BS_j = \{b_{i,j} | z_i = 0\}$. The significance pass also transmits the sign of the new significant coefficients. The refinement pass transmits the subset of the bitplane $B_j$ corresponding to the $j$th most significant bits of already significant coefficients $BR_j = \{b_{i,j} | z_i = 1\}$. Many existing bitplane encoding algorithms differ essentially in the way they perform the significance pass. In our case, we use an approach based on adaptive Golomb codes. Here, the significance pass does not transmit directly the bits in $BS$ but instead transmits the number of zeros between ones using adaptive Golomb codes. The parameter $k$ of the Golomb coder is initialized to a fixed value $k_{\text{init}}$ before encoding each bitplane. Then, the bits are encoded using the following simple algorithm: if a sequence of $2^k$ zeros is found in $BS$, a bit 0 is transmitted and $k$ is updated $k \leftarrow k + 1$; if not, a number of zeros inferior to $2^k$ followed by a one remains, this number of zeros is transmitted on $k$ bits preceded by the bit 1 and $k$ is updated $k \leftarrow k - 1$. Each one found in $BS$ corresponds to a new significant coefficient; consequently, the sign of this coefficient is also transmitted. This process is repeated until the end of $BS$ is reached. The complete algorithm of the simple bitplane encoder is detailed in Algorithm 3.

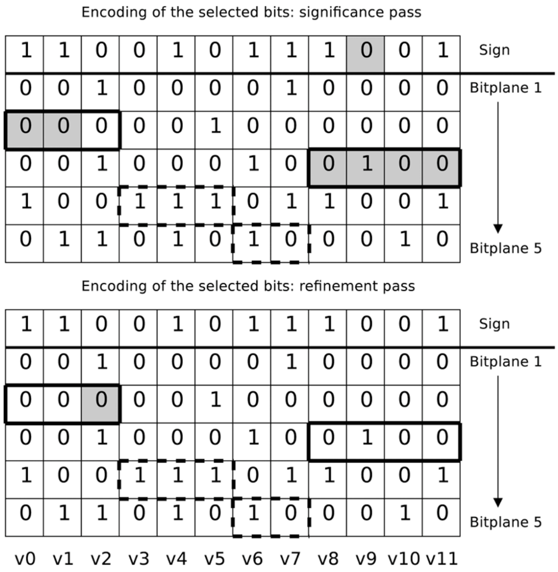
**Algorithm 3** Simple bitplane encoder

```
input: the coefficients vector $\mathbf{v}$
output: the bitstream
  quantize and code max amplitude $A = \max(\text{abs}(v_i))$
  $z_i = 0$ for all $i$
  $\nu = 1$
  repeat
    $b_i = \text{mod}(\text{floor}(\text{abs}(v_i) * 2^\nu / A), 2)$
    $BS = \{b_i | z_i = 0\}$
    $BR = \{b_i | z_i = 1\}$
    $k = k_{\text{init}}$
    repeat
      if sequence of $2^k$ zeros in $B_S$ then
        emit the bit 0
        $k = k + 1$
      else
        emit the bit 1
        emit $k$ bits: number of zeros followed by a one
        emit 1 bit: sign of corresponding coefficient
        $k = k - 1$
      end if
      move to the next bits in $BS$
    until the end of $BS$
    emit the sequence of bits in $BR$
    $z_i = 1$ for all $i$ such that $b_i \in BS$ and $b_i = 1$
    $\nu = \nu + 1$
  until bit budget spent or $\nu > \nu_{\max}$.
```

### B. *Psychoacoustic Bitplane Encoder*

The simple bitplane encoder sends the coefficients in decreasing order of amplitude. However, the coefficients with the highest amplitude are not necessarily the most perceptually relevant coefficients. Indeed, some components are masked and some others are below the absolute threshold of hearing. It is therefore preferable to send first the most perceptually relevant coefficients using a psychoacoustic model. However, existing psychoacoustic approaches as used in transform coding cannot be easily applied to union of MDCT bases representations. This is due to two main reasons. The first reason is that the psychoacoustic models used in transform coding are designed for a fixed resolution representation and are not adapted to a multiresolution representation where time-localized components (short window atoms) and frequency-localized components (long window atoms) are superimposed. We thus propose a suboptimal approach where a masking threshold is computed for each MDCT as if they were independent MDCTs. In each frame of each block, a spectral analysis is performed and the Johnston model [37] is used to compute a mask for the corresponding frame of coefficients. The second reason is that there are more components in the overcomplete case and thus more masking values, it is then more costly to send the psychoacoustic masking threshold to the decoder. Instead, we propose a suboptimal approach inspired from [15] where the mask is computed on the partially coded coefficients. Consequently, there is no need to transmit the mask to the decoder, as it is computed the same way by the decoder on the partially decoded coefficients.

Here, we propose a modification of the simple bitplane encoder which shapes the quantization noise according to the psychoacoustic model. The complete algorithm is detailed in Algorithm 4. At each iteration of the algorithm, only a subset of the current bitplane is sent. A masking threshold is used to select the bits which are transmitted. First, the masking threshold is initialized to the absolute threshold of hearing (ATH). Second, the mask is updated every time $U$ bits have been added to the bitstream. To update the mask, a synthesized signal is first reconstructed from the partially coded coefficients. Then, a masking threshold is computed in each frame of each block as mentioned above. The psychoacoustic model gives a masking threshold value $th_i$ for each coefficient, and finally, a simple rule decides which bits are transmitted: the difference gap $g_i$ between the masking threshold (converted to the bitplane scale) and the current bitplane level is computed: $g_i = -\text{log2}(\sqrt{th_i}/A) - \nu_i$. Then, we select the bits whose gaps are greater than the mean of the gap: $j$ such that $g_j \geq \text{mean}(g_i)$. Fig. 9 shows one iteration of the algorithm. Contrary to the simple bitplane encoder, the current level is different for each coefficient because only a subset of the bitplane is sent at each iteration. The gap is computed for each coefficient and the coefficients whose gap value



Encoding of the selected bits: significance pass

| v0 | v1 | v2 | v3 | v4 | v5 | v6 | v7 | v8 | v9 | v10 | v11 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | Sign |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | Bitplane 1 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | Bitplane 5 |

Encoding of the selected bits: refinement pass

| v0 | v1 | v2 | v3 | v4 | v5 | v6 | v7 | v8 | v9 | v10 | v11 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | Sign |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | Bitplane 1 |
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | |
| 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | |
| 0 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | Bitplane 5 |

Fig. 9. One iteration of the psychoacoustic bitplane encoder. Plain rectangles correspond to the level of the selected coefficients. Dotted rectangles correspond to the level of the nonselected coefficients. The two encoding passes are shown. The bits in gray correspond to the transmitted bits.

is above the mean of the gap are selected, here the coefficients 0,1,2,8,9,10,11. Finally, the selected bits are encoded using the same two-pass scheme as in the simple bitplane encoder.

**Algorithm 4** Psychoacoustic bitplane encoder

```
input: the coefficients vector v
output: the bitstream
  quantize and code max amplitude $A = \max(\mathrm{abs}(v_i))$
  $z_i = 0$ and $\nu_i = 1$ for all $i$
  $th_i$ initialized to the ATH
  repeat
    if $U$ new bits have been added to the bitstream then
      update mask $th_i$
    end if
    $g_i = -\mathrm{log2}(\sqrt{th_i}/A) - \nu_i$
    $s_i = g_i \geq \mathrm{mean}(g_i)$
    $b_i = \mathrm{mod}(\mathrm{floor}(\mathrm{abs}(v_i) * 2^{\nu_i}/A), 2)$
    $B_S = \{b_i | z_i = 0 \text{ and } s_i = 1\}$
    $B_R = \{b_i | z_i = 1 \text{ and } s_i = 1\}$
    $k = k_{\mathrm{init}}$
    repeat
      if sequence of $2^k$ zeros in $B_S$then
        emit the bit 0
        $k = k + 1$
      else
        emit the bit 1
        emit $k$ bits: number of zeros followed by a one
        emit 1 bit: sign of corresponding coefficient
        $k = k - 1$
      end if
      move to the next bits in $B_S$
    until the end of $B_S$
    emit the sequence of bits in $B_R$
    $z_i = 1$ for all $i$ such that $b_i \in B_S$ and $b_i = 1$
    $\nu_i = \nu_i + 1$ for all $i$ such that $s_i = 1$
  until bit budget spent or $\nu_i > \nu_{\max}$.
```

TABLE I
TESTING MATERIAL

| Test Item | Description |
|---|---|
| harp | Harpsichord |
| bagp | Bagpipes |
| gloc | Glockenspiel |
| orch | Orchestral piece |
| popm | Contemporary pop music |

## V. EVALUATION

### A. Test Material

Five signals were used, from the dataset of [13] (see Table I). This subset was chosen to include critical and rather varied material (from mono-instrumental to complex polyphonic signals). The length of each signal is approximately 10 s. We resampled the signals from 48 to 44.1 kHz and kept only the left channel.

### B. Coder Configuration

To evaluate the coder, the following parameter values have been chosen.

*1) Decomposition:* The coder tested is based on a union of $M = 8$ bases, with the half of the shortest window length being $L_0 = 64$. Two analysis windows have been tested: the sine window and the KBD window introduced in Section II. Since there were no noticeable differences in the results using one or the other, we used the sine window (8). The signal is padded with one second of zeros on both sides. The parameter $thresh$ in the modified MP is set to 100 and the target SNR is set to 80 dB. We found it necessary to use such a high value to obtain relevant objective measurements at high bitrates (above 128 kb/s). However, in practice, it is sufficient to use a target SNR between 30 and 50 dB for low to medium bitrates.

*2) Coding:* The maximum amplitude value is quantized and coded using 16 bits. The parameter $k_{\mathrm{init}}$ is set to 2. The bit budget for each timeslot is constant and is set according to the target bitrate. The maximum level $v_{\max}$ is set to 30. The number of bits $U$ necessary to update the mask is set to 1 kb. The critical bands for the Johnston model have a length of approximately 0.3 barks. The ATH formula is the one provided in [38], and the spreading function formula is the one provided in MPEG-4 specifications [2].

### C. Objective Evaluation

We use PEMO-Q software [39] as an objective measure to evaluate our coder. This software provides two values, the perceptual similarity measure (PSM) which is restricted to the in-

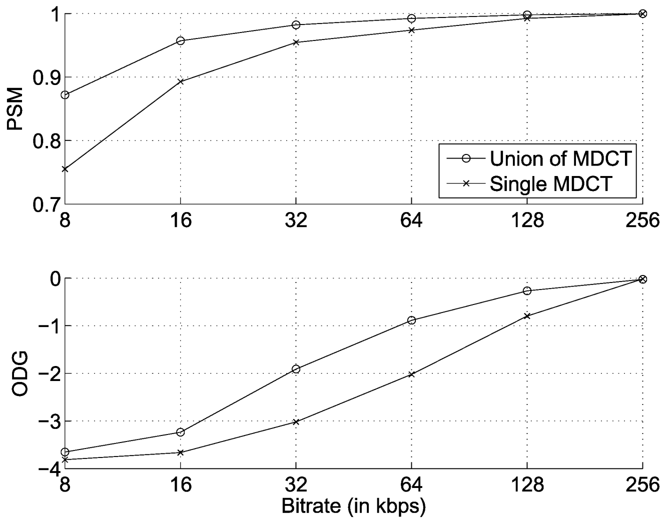


Fig. 10. Mean of PSM and ODG values for five signals and two coders based on a single MDCT and a union of eight MDCTs.

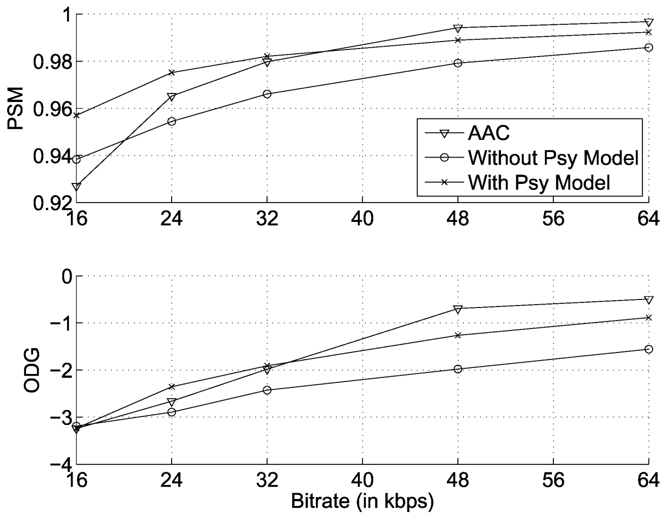


Fig. 11. Mean of PSM and ODG values for five signals and three coders: iTunes's AAC, our coder without psychoacoustic model, our coder with psychoacoustic model.

terval $[-1, 1]$ (1 indicates transparency) and the objective difference grade (ODG) which is on the same scale as the subjective difference grade (0 indicates transparency). It is important to note that PEMO-Q was optimized and validated with high-bitrate audio coders. Consequently, the results at low bitrates may be less relevant.

In our first experiment, our coder is compared to a similar transform-based coder. This makes it possible to assess the usefulness of using an overcomplete basis, all other things (quantization, coding) being equal. The reference coder uses only $M = 1$ MDCT basis with $L_0 = 1024$ samples and exactly the same coding algorithm. This experiment is an extension of prior work where no psychoacoustic model was used [34]. Note that the results of the transform coder may have been improved by using a time-varying MDCT with a switching block size at transients. The mean of the PSM and ODG values over the five signals are in Fig. 10. The union of MDCT approach gives significantly better results at low bitrates while performing similarly at high bitrates.

In our second experiment, two versions of our coder are compared with a reference state-of-the-art coder. The two versions of our coder are the coder without the psychoacoustic model which is then the same as in [34] and the coder with the psychoacoustic model. The reference coder is the iTunes 7 AAC encoder [40] as it is freely available and a fully AAC compatible encoder. The mean of the PSM and ODG over the five signals are in Fig. 11. The results show that the psychoacoustic model adds a significant gain to the results of the previous coder, and gives a coder which is competitive with the iTunes AAC encoder.

### D. Subjective Evaluation

Two versions of our coder are subjectively evaluated with a MUSHRA listening test [41], with scores that range from 0 (extremely poor quality) to 100 (transparent). Ideally, an ABC test would also be needed to test transparency at high bitrates. However, performing a listening test is very time consuming, and we have preferred to test the performance of our coder at medium and low bitrates only. A total of 20 listeners evaluated eight versions of the five test signals of Table I: a hidden reference, a 3.5-kHz low-pass anchor, two signals coded with iTunes AAC at 24 and 48 kb/s, two signals coded with our coder and the simple bitplane encoder (noted `codec_A`) at 24 and 48 kb/s, and two signals coded with our coder and the psychoacoustic bitplane encoder (noted `codec_B`) at 24 and 48 kb/s. These 40 sound files and the source code of the codec are available online.[1] The participants were post-screened according to the score they attributed to the reference: the data for listeners who gave a score under 90 (which corresponds to the lowest mark for the category excellent quality) was discarded. A total of 12 listeners were kept, mostly graduate students working in the field of acoustics or audio processing. Fig. 12 presents the scores for each five signals, with the overall scores shown at the bottom.

The first goal of this listening test is to show the benefit of the psychoacoustic bitplane encoder (`codec_B`) compared to the simple bitplane encoder (`codec_A`). The results are very signal dependent. Simple signals (bagp, harp, and gloc) are very sparse in the time–frequency domain; in this case there is no gain when using the psychoacoustic bitplane encoder. For harp and gloc signals, the psychoacoustic approach even slightly degrades the performance, though scores still remain high. For more complex, polyphonic signals, the psychoacoustic approach significantly increases the scores. This is due to the presence of a very large number of time–frequency components that mask each other.

The second goal of this listening test was to compare our proposed coders with a reference pure transform coder, the iTunes AAC coder. These results are also very signal dependent: at low bitrates, our approach gives much better results for monophonic signals (bagp, harp, and gloc) and slightly worse results for polyphonic signals (orch and popm). At high bitrates, our approach with the psychoacoustic model gives the same or slightly

[1] The sound files used in the listening tests and the GNU/GPL source code of the codec needed to reproduce the results are available online at the following address: http://www.emmanuel-ravelli.com/taslp08

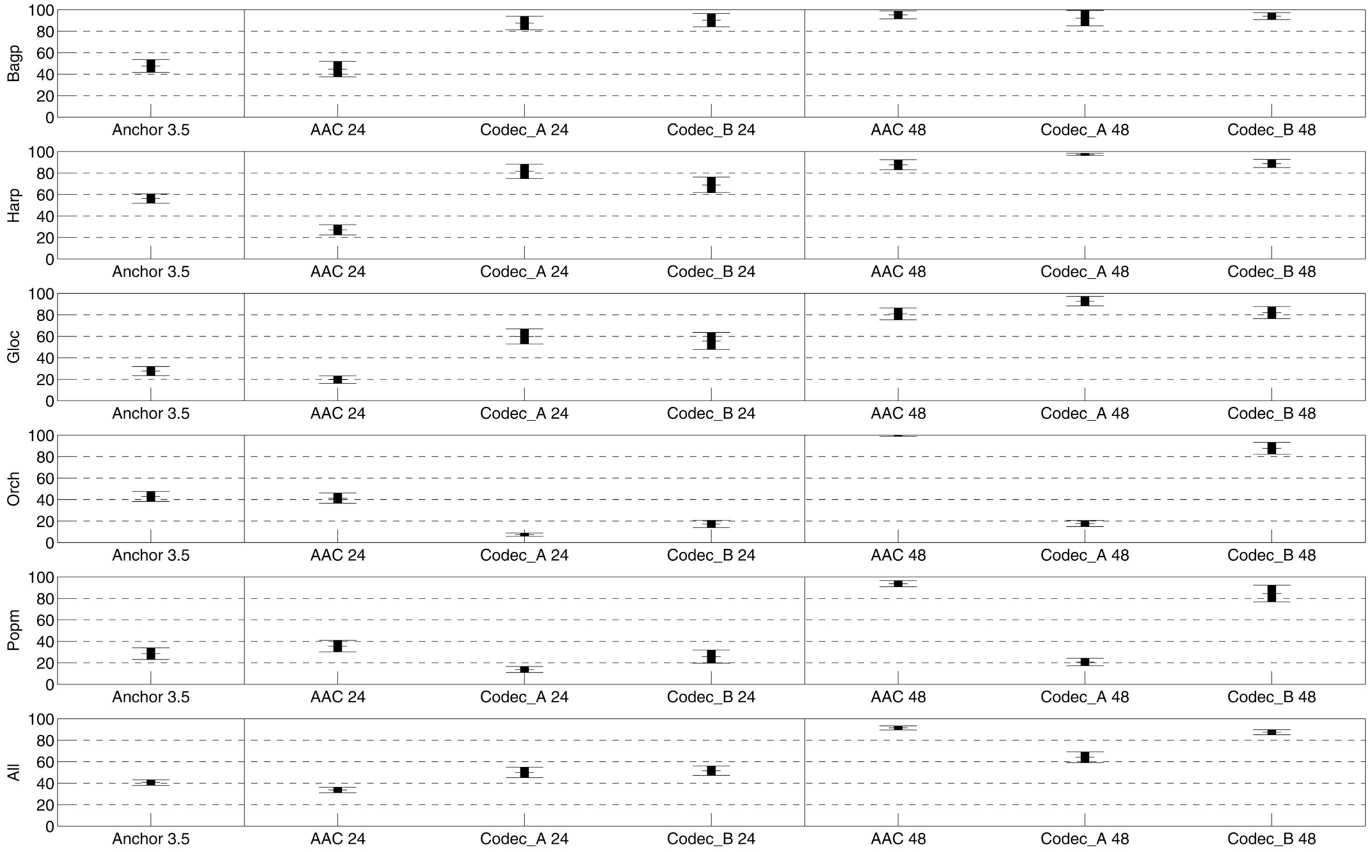


Fig. 12. MUSHRA listening test results. From left to right: 3.5-kHz low-pass anchor, iTunes AAC at 24 kb/s, our coder with the simple bitplane encoder (Codec A) at 24 kb/s, our coder with the psychoacoustic bitplane encoder (Codec B) at 24 kb/s, iTunes AAC at 48 kb/s, our coder with the simple bitplane encoder (Codec A) at 48 kb/s, our coder with the psychoacoustic bitplane encoder (Codec B) at 48 kb/s. From top to bottom: Bagpipe, Glockenspiel, Harpsichord, Orchestra, Pop Music, All signals.

worse results, except for the case of polyphonic signals and the simple bitplane encoder, where the poor results show that a psychoacoustic approach is necessary.

Overall, these results show that our approach is competitive with a state-of-the-art pure transform coder, especially since we did not spend much effort in the optimization of encoding parameters as opposed to the highly-optimized AAC coders. It is also interesting to note here that, as opposed to the reference coder, our approach provides fine-grain scalability, a property which is generally provided at a cost in terms of performance. An example of a fine-grain scalable coder is MPEG-4 BSAC. Formal listening tests [42] show that at low bitrates BSAC requires a 12.5% bitrate overhead compared to AAC for the same quality.

## VI. Conclusion

This paper proposes a new signal representation for audio coding. Contrary to existing methods, this approach provides transparency at high bitrates and competitive results at low bitrates. Listening tests that compare our proposed coder with a state-of-the-art transform-based coder show that our approach provides much better quality for monophonic signals and similar or only slightly worse results for polyphonic signals. However, it should be emphasized that the main goal of this paper is to show that the concept of sparse overcomplete representations over a union of MDCT bases is a viable approach for audio coding. Our coder has also been designed with the extra constraint of fine-grain scalability, although this is by no means a requirement for our concept.

However, these results are obtained at a cost in terms of computation time. Indeed, the decomposition algorithm is much slower than what is needed for a transform (which approximately corresponds to the first iteration of the Matching Pursuit algorithm). Moreover, our approach uses much longer analysis window lengths (up to 16 384 samples at 44.1 kHz) than those used in an AAC coder (2048 samples) which results in much greater delay time. The delay of our coder is equal to the maximum window length, i.e., 371.5 ms. These two issues prevent the use of this approach in applications that require real-time/low-delay encoding. The computation times are high but not excessive (typically between 5 and 200 times the duration of the file, depending on the signal and the target bitrate), and are acceptable for offline sound databases or personal music collections.

While the proposed coder is competitive with state-of-the-art transform coders, there is still much room for improvement. One possibility is to use more efficient decomposition algorithms such as orthogonal Matching Pursuit [26] or Basis Pursuit [28]. We can also improve the coding part: a better psychoacoustic model, a better integration of the psychoacoustic model into the bitplane encoder and a fixed bitrate coding scheme instead of bitplane encoding. We are also currently working on the application of such decompositions to audio indexing tasks. We believe that these representations can be very useful for transform domain applications such as beat tracking, tonality estimation, and audio classification.

## ACKNOWLEDGMENT

The authors would like to thank the members of LAM and TSI for participating in the listening tests, N. Moreau for helpful discussions, and B. Sturm and J. Benenson for proof reading. The authors would also like to thank R. Gribonval and S. Krstulovic for their help in implementing the contributions to the Matching Pursuit ToolKit.

## REFERENCES

[1] M. Bosi, K. Brandenburg, S. Quackenbush, L. Fielder, K. Akagiri, H. Fuchs, and M. Dietz, "ISO/IEC MPEG-2 advanced audio coding," *J. Audio Eng. Soc.*, vol. 45, no. 10, pp. 789–814, Oct. 1997.

[2] Information TechnologyCoding of Audio-Visual ObjectsPart 3: Audio, ISO/IEC, JTC1/SC29/WG11 MPEG, 2001, IS14496-3.

[3] Report on the Formal Subjective Listening Tests of MPEG-2 NBC Multichannel Audio Coding, ISO/IEC, JTC1/SC29/WG11 MPEG, 1996, Tech. Rep. N1419.

[4] Report on the MPEG-2 AAC Stereo Verification Tests, ISO/IEC, JTC1/SC29/WG11 MPEG, 1998, Tech. Rep. N2006.

[5] L. Daudet and M. Sandler, "MDCT analysis of sinusoids: Exact results and applications to coding artifacts reduction," *IEEE Trans. Speech Audio Process.*, vol. 12, no. 3, pp. 302–312, May 2004.

[6] O. Derrien, P. Duhamel, M. Charbit, and G. Richard, "A new quantization optimization algorithm for the MPEG advanced audio coder using a statistical subband model of the quantization noise," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 4, pp. 1328–1339, Jul. 2006.

[7] A. d. Brinker, E. Schuijers, and A. Oomen, "Parametric coding for high-quality audio," in *Proc. 112th AES Convention*, 2002, paper 5554.

[8] E. Schuijers, W. Oomen, B. d. Brinker, and J. Breebaart, "Advances in parametric coding for high-quality audio," in *Proc. 114th AES Convention*, 2003, paper 5852.

[9] Report on the Verification Tests of MPEG-4 Parametric Coding for High Quality Audio, ISO/IEC, JTC1/SC29/WG11 MPEG, 2004, Tech. Rep. N6675.

[10] M. Dietz, L. Liljeryd, K. Kjrling, and O. Kunz, "Spectral band replication, a novel approach in audio coding," in *Proc. 112th AES Convention*, 2002, paper 5553.

[11] M. Wolters, K. Kjorling, D. Homm, and H. Purnhagen, "A closer look into MPEG-4 high efficiency AAC," in *Proc. 115th AES Convention*, 2003, paper 5871.

[12] J. Makinen, B. Bessette, S. Bruhn, P. Ojala, R. Salami, and A. Taleb, "Amr-Wb+: A new audio coding standard for 3rd generation mobile audio services," in *Proc. Int. Conf. Acoust., Speech, Signal. Process.*, Mar. 2005, vol. 2, pp. 1109–1112.

[13] Report on Informal MPEG-4 Extension 1 (Bandwidth Extension) Verification Tests, ISO/IEC, JTC1/SC29/WG11 MPEG, 2003, Tech. Rep. N5571.

[14] S.-H. Park, Y.-B. Kim, and Y.-S. Seo, "Multi-layer bit-sliced bit rate scalable audio coding," in *Proc. 103rd AES Convention*, 1997, preprint 4520.

[15] J. Li, "Embedded audio coding (EAC) with implicit auditory masking," in *Proc. ACM Multimedia*, 2002, pp. 592–601.

[16] C. Dunn, "Scalable bitplane runlength coding," in *Proc. 120th Audio Eng. Soc. Convention*, 2006, paper 6749.

[17] S. Levine and J. O. Smith, "A sines + transient + noise audio representation for data compression and time/pitch scale modifications," in *Proc. 115th AES Convention*, Sep. 1998, paper 4781.

[18] A. O. d. Brinker, "Sinusoids plus noise modeling for audio signals," *Proc. AES 17th Int. Conf.*, pp. 226–232, Sep. 1999.

[19] M. Goodwin, "Multiscale overlap-add sinusoidal modeling using matching pursuit and refinements," in *Proc. Workshop Applicat. Signal Process,. Audio Acoust.*, Oct. 2001, pp. 207–210.

[20] J. Princen, A. Johnson, and A. Bradley, "Subband/transform coding using filter bank designs based on time domain aliasing cancellation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Apr. 1987, vol. 12, pp. 2161–2164.

[21] H. S. Malvar, "Lapped transforms for efficient transform/subband coding," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 38, no. 6, pp. 969–978, Jun. 1990.

[22] C. Fvotte and S. J. Godsill, "Sparse linear regression in unions of bases via Bayesian variable selection," *IEEE Signal Process. Lett.*, vol. 7, no. 7, pp. 441–444, Jul. 2006.

[23] Information TechnologyCoding of Moving Pictures and Associated Audio for Digital Storage Media at up to About 1.5 Mbit/sPart 3: Audio, ISO/IEC, JTC1/SC29/WG11 MPEG, 1992, IS11172-3.

[24] Information TechnologyGeneric Coding of Moving Pictures and Associated Audio InformationPart 3: Audio, ISO/IEC, JTC1/SC29/WG11 MPEG, 1998, IS13818-3.

[25] Digital Audio Compression Standard (AC-3) United States Advanced Television Systems Committee (ATSC),, 1995, Doc. A/52.

[26] G. Davis, S. Mallat, and M. Avellaneda, "Greedy adaptive approximation," *J. Constr. Approx.*, vol. 13, pp. 57–98, 1997.

[27] S. Mallat and Z. Zhang, "Matching pursuits with time–frequency dictionaries," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.

[28] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Scientific Comput.*, vol. 20, no. 1, pp. 33–61, 1999.

[29] B. D. Rao and K. Kreutz-Delgado, "An affine scaling methodology for best basis selection," *IEEE Trans. Signal Process.*, vol. 47, no. 1, pp. 187–200, Jan. 1999.

[30] R. Gribonval, E. Bacry, S. Mallat, P. Depalle, and X. Rodet, "Analysis of sound signals with high resolution matching pursuit," in *Proc. Int. Symp. Time-Frequency Time-Scale Anal.*, 1996, pp. 125–128.

[31] S. Jaggi, W. Carl, S. Mallat, and A. Willsky, High resolution pursuit for feature extraction, MIT, 1995, Tech. Rep..

[32] S. G. R. Krstulovic, "Mptk: Matching pursuit made tractable," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2006, vol. 3, pp. 496–499.

[33] G. G. Langdon, "An adaptive run-length encoding algorithm," *IBM Tech. Discl. Bull.*, vol. 26, pp. 3783–3785, 1983.

[34] E. Ravelli, G. Richard, and L. Daudet, "Extending fine-grain scalable audio coding to very low bitrates using overcomplete dictionaries," in *Proc. IEEE Workshop Applicat. Signal Process. Audio Acoust. (WASPAA'07)*, Oct. 2007, pp. 195–198.

[35] E. Ordentlich, M. Weinberger, and G. Seroussi, "A low-complexity modeling approach for embedded coding of wavelet coefficients," in *Proc. Data Compression Conf.*, Mar. 1998, pp. 408–417.

[36] H. S. Malvar, "Fast progressive wavelet coding," in *Proc. Data Compression Conf.*, 1999, pp. 336–343.

[37] J. D. Johnston, "Transform coding of audio signals using perceptual noise criteria," *IEEE J. Sel. Areas Commun.*, vol. 6, no. 2, pp. 314–323, Feb. 1988.

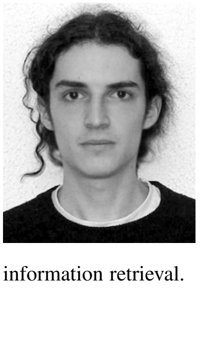
[38] T. Painter and A. Spanias*, Perceptual Coding of Digital Audio*, vol. 88, no. 4, pp. 451–515, 2000.

[39] R. Hubert, B. Kollmeier, and N. 2006, "PEMO-Qa new method for objective audio quality assessment using a model of auditory perception," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 6, pp. 1902–1911, Nov..

[40] Apple iTunes 7 Webpage, [Online]. Available: http://www.apple.com/fr/itunes/download/

[41] ITU-R BS.1534-1: Method for the subjective assessment of intermediate quality levels of coding systems, ITU, 2003.

[42] Report on the MPEG-4 Audio Version 2 Verification Test, ISO/IEC, JTC1/SC29/WG11 MPEG, 1999, Tech. Rep. N3075.



**Emmanuel Ravelli** (S'08) received the B.S. degree in electronics and signal processing engineering from the École Nationale Supérieure d'Electrotechnique, d'Electronique, d'Informatique, d'Hydraulique de Toulouse, Toulouse, France, in 2004 and the M.Sc. degree in signal processing from the University Paul Sabatier, Toulouse, in 2005. He is currently pursuing the Ph.D. degree at the Pierre et Marie Curie University—Paris 6, Paris, France.

His main research interests are digital audio signal processing, audio coding, and music information retrieval.

**Gaël Richard** (M'02–SM'06) received the State Engineering degree from the École Nationale Supérieure des Télécommunications (ENST), Paris, France, in 1990, the Ph.D. degree in speech synthesis from LIMSI-CNRS, University of Paris-XI, in 1994, and the Habilitationà Diriger des Recherches degree from the University of Paris XI in September 2001.

After the Ph.D, degree, he spent two years at the CAIP Center, Rutgers University, Piscataway, NJ, in the Speech Processing Group of Prof. J. Flanagan, where he explored innovative approaches for speech production. From 1997 to 2001, he successively worked for Matra Nortel Communications, Bois d'Arcy, France, and for Philips Consumer Comunications, Montrouge, France. In particular, he was the Project Manager of several large-scale European projects in the field of audio and multimodal signal processing. In September 2001, he joined the Department of Signal and Image Processing of TELECOM ParisTech (formerly ENST), where he is now a Full Professor in audio signal processing and Head of the Audio, Acoustics, and Waves research group. He is coauthor of over 70 papers and inventor of a number of patents. He is also an expert of the European commission in the field of audio signal processing and man/machine interfaces.

Prof. Richard is an Associate Editor of the IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING.

**Laurent Daudet** (M'04) received the M.S. degree in statistical and nonlinear physics from the École Normale Supérieure, Paris, France, in 1997 and the Ph.D. degree in mathematical modeling from the Université de Provence, Marseille, France, in audio signal representations, in 2000.

In 2001 and 2002, he was a Marie Curie Postdoctoral Fellow with the Centre for Digital Music at Queen Mary, University of London, London, U.K. Since 2002, he has been working as an Assistant Professor at the University Pierre et Marie Curie—Paris 6, Paris, France, where he joined the Laboratoire d'Acoustique Musicale, now part of Institut Jean Le Rond D'Alembert. His research interests include audio coding, time–frequency and time-scale transforms, and sparse representations for audio.