Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

Signal Processing 86 (2006) 432–443

SIGNAL PROCESSING

www.elsevier.com/locate/sigpro

Fast implementation of an improved parametric audio coder based on a mixed dictionary

P. Vera-Candeas[a,*], N. Ruiz-Reyes[a], M. Rosa-Zurera[b], J.C. Cuevas-Martínez[a], F. López-Ferreras[b]

[a]*Electronics and Telecommunication Engineering Department, University of Jaen, Polytechnic School, 23700 Linares, Jaen, Spain*
[b]*Signal Theory and Communications Department, University of Alcala, Polytechnic School, 28871 Alcala de Henares, Madrid, Spain*

Received 29 October 2004; received in revised form 6 May 2005; accepted 30 May 2005
Available online 2 August 2005

Abstract

This paper deals with the application of adaptive signal models for representing transients and sinusoids at the same stage in a parametric audio coder. To accomplish such a goal, we search for sparse approximations by means of matching pursuit with a mixed dictionary, instead of using two different dictionaries that operate in cascade. In such sense, complex exponentials and wavelet packets are chosen for modeling the tonal and transient features of an audio signal, respectively. At each iteration of the pursuit, the mixed dictionary function that extracts the most energy from the residue is selected. This function will be either a complex exponential or a wavelet packet, depending on the characteristics of the residue at that iteration. Experimental results clearly show the objective (compression rate) and subjective (% preference) advantages of the mixed dictionary over two cascaded dictionaries. The approach proposed in this paper is successfully applied for parametric audio coding purposes, assuring better perceptual audio quality than MPEG2/4-AAC at 16 Kbits/s for most of the CD-quality one channel audio signals considered for testing.

Keywords: Matching pursuit; Overcomplete dictionary; Sparse approximation; Parametric audio coding; Wavelet packets; Complex exponentials

1. Introduction

Parametric or model-based coding of audio signals has become a popular tool for representing audio signals at very low bit rates [1–9]. All signal models assume an underlying structure to the signal in question. A wide range of audio signals intuitively fit into the three-part model of sines, transients and noise. Transients usually describe drum hits and sudden starts of many instruments, sines describe signal components that have a distinct tonality, and noise often describes the

*Corresponding author. Tel.: +34 953 646554; fax: +34 953 646508.
E-mail address: pvera@ujaen.es (P. Vera-Candeas).

0165-1684/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.sigpro.2005.05.022

rest of the signal that is neither sinusoidal nor transient. This model consists of three parts that work together and complement each other to form a complete and robust signal model, which makes highly optimized audio compression schemes possible (STN model-based audio coders).

Several approaches have been adopted in literature concerning the order in which the sinusoidal and transient models are applied. In all approaches, the noise model makes use of the residual resulting from the sinusoidal and transient models. In the coder of Levine and Smith [3], transform coding was utilized to encode the complete signal when a transient occurred, while during non-transient intervals the sinusoidal and noise models were applied in cascade [3]. In HILN [5], a pre-analysis step is performed to detect transients and determine the amplitude envelope of the audio signal in the frame. The presence of a transient is signaled to the sinusoidal model, which then analyzes the audio signal over a short frame. In the coder of Ali [1], the sinusoidal, transient, and noise models are applied in cascade. After sinusoidal coding, the sinusoidal component is subtracted from the audio signal, resulting in a first residual, which is then used as input to the transient coder. The transient component is subtracted from the first residual, resulting in a second residual, which is then coded by the noise coder.

In contrast to the coder of Ali, recent STN model-based audio coders [6–9] apply sinusoidal coding after transient coding, followed by noise coding. The reasoning behind this approach is that sinusoids are suitable functions for modeling the tonal, quasi stationary aspects of an audio signal. The presence of transients disturbs the stationarity of the audio signal, thus complicating the task of the sinusoidal model (i.e. if a sinusoidal model represents a signal onset, the attack becomes smeared in time, resulting in a pre-echo). By removing transients from the audio signal prior to sinusoidal modeling, this problem is avoided. This approach provides good quality audio coding at low bit rates (about 24 Kbits/s per channel) for most audio excerpts. Nevertheless, it has two main drawbacks:

- Since transients and sinusoids are modeled in cascade (first, transients, and then, sinusoids), mismatching problems can appear if the two components are not properly separated when a signal onset is detected. The problem of separating transients from sinusoids in transient intervals must be appropriately posed.
- Transient detection tools very often fail to detect micro-transients [3], which would not be properly represented if transients and sinusoids are modeled in cascade.

At the sight of these problems, we propose modeling transients and sinusoids at the same stage of the encoder. To accomplish such a goal, we search for sparse approximations by means of matching pursuit with a mixed dictionary. The dictionary must be defined from two types of functions: (1) functions that match well to sharp transitions in the signal; (2) functions that represent the tonal, quasi stationary aspects of an audio signal. In such sense, complex exponentials and wavelets are chosen for modeling the tonal and transient features of an audio signal, respectively. At each iteration of the pursuit, the dictionary element that extracts the most energy from the residue is selected. Depending on the characteristics of the residue at that iteration, this function will be either a complex exponential or a wavelet.

The use of a mixed dictionary composed of complex exponentials and wavelets not only provides an efficient representation of the audio signal, but also a better subjective quality of the decoded audio signals, as will be assessed in the experimental results.

2. Sparse approximations

2.1. Principles of atomic modeling

Atomic signal representations have been of ongoing interest due to their profitable properties in order to obtain compact time–frequency decompositions. The fundamental notion of atomic modeling is that a signal can be decomposed into elementary waveforms (atoms). The set of

possible atoms is known as dictionary and is chosen to be adapted to the time–frequency behavior of the signal [10].

Let $\mathscr{H}$ be a Hilbert space. The dictionary $\mathscr{D}$ is then defined as a family of functions $\mathscr{D} = \{\mathbf{g}_i;\ i = 0, 1, \ldots, M\}$ in $\mathscr{H}$, such as $\|\mathbf{g}_i\| = 1, \forall i$, where M is the dictionary number of functions. The dictionary is overcomplete if $M > N$, where N is the signal length. A signal model of the form

$$\mathbf{x} \approx \sum_{i=1}^{M} \alpha_i \mathbf{g}_i \tag{1}$$

constitutes a sparse approximation when $\alpha_i \neq 0$ only for a low number of K values ($K \ll M$).

A signal can be approximated with few atoms when an overcomplete set of atoms adapted to the time–frequency behavior of the signal is defined. However, the price to pay for overcompleteness is an increase in both the complexity and the "overhead" cost to encode the indexes of retained coefficients.

There is a wide variety of approaches for deriving overcomplete signal expansions, which differ on the method of selecting atoms from the dictionary. Such approaches can be roughly grouped into two categories: (a) parallel methods, such as the method of frames, basis pursuit [11], and FOCUSS [12], in which computation of the various expansion components is coupled; (b) sequential methods, such as matching pursuit [13] and its variations, in which models are computed one component at a time and derive sparse approximate solutions according to sub-optimal criteria.

Since sparse approximate solutions are of interest for compact signal modeling, we have chosen matching pursuit for deriving overcomplete signal expansions in the proposed audio compression scheme.

2.2. Matching pursuit

Matching pursuit was introduced by Mallat and Zhang [13]. The problem of choosing K functions $\mathbf{g}_i$ that best approximate the analyzed signal $\mathbf{x}$ is computationally very complex [14]. Matching pursuit is an iterative algorithm that offers suboptimal solutions for decomposing a signal $\mathbf{x}$ in terms of unit-norm expansion functions $\mathbf{g}_i$ chosen from an overcomplete dictionary $\mathscr{D}$, where the l^2 norm is used as the approximation metric. When a well-designed dictionary is used in matching pursuit, compact adaptive signal decompositions are achieved.

At the first iteration, the function (or atom) $\mathbf{g}_i$ which gives the highest inner product with the analyzed signal $\mathbf{x}$ is chosen. The contribution of this function is then subtracted from the signal and the process is repeated on the residual. At the mth iteration, it follows:

$$\mathbf{r}^{m+1} = \mathbf{r}^m - \alpha_{i(m)}\mathbf{g}_{i(m)} \quad m \geqslant 1, \tag{2}$$

where $\alpha_{i(m)}$ is the weight associated to the optimum atom $\mathbf{g}_{i(m)}$ at the mth iteration and $\mathbf{r}^1$ is initialized to $\mathbf{x}$.

Computing the orthogonal projections of $\mathbf{r}^m$ on elements $\mathbf{g}_i \in \mathscr{D}$, the weight associated to each dictionary element at the mth iteration is computed as

$$\boldsymbol{\alpha}_i^m = \langle \mathbf{r}^m, \mathbf{g}_i \rangle. \tag{3}$$

The optimum atom $\mathbf{g}_{i(m)}$ at the mth iteration is obtained by minimizing residual energy:

$$\mathbf{g}_{i(m)} = \arg\min_{\mathbf{g}_i \in D} \|\mathbf{r}^{m+1}\|^2 = \arg\max_{\mathbf{g}_i \in D} |\boldsymbol{\alpha}_i^m|. \tag{4}$$

The computation of correlations $\langle \mathbf{r}^m, \mathbf{g}_i \rangle$ for all vectors $\mathbf{g}_i$ at each iteration implies a high computational effort, which can be substantially reduced using an updating procedure derived from (2). The correlation updating procedure is performed as follows [13]:

$$\langle \mathbf{r}^{m+1}, \mathbf{g}_i \rangle = \langle \mathbf{r}^m, \mathbf{g}_i \rangle - \alpha_{i(m)} \langle \mathbf{g}_{i(m)}, \mathbf{g}_i \rangle. \tag{5}$$

Correlations $\langle \mathbf{g}_{i(m)}, \mathbf{g}_i \rangle$ can be pre-calculated and stored, once the overcomplete set $\mathscr{D}$ has been determined. Therefore, it is only necessary to compute once the correlations with the explicit formula, at the first iteration.

3. Matching pursuit with a mixed dictionary based on sines + wavelets

The operation of matching pursuit with a mixed dictionary composed of sinusoids and wavelets is described in this section. The mixed dictionary $\mathscr{D}$ is obtained by merging a dictionary of complex exponentials $\mathscr{D}_{\mathrm{e}}$ with a dictionary of wavelets $\mathscr{D}_{\mathrm{w}}$ ($\mathscr{D}=\mathscr{D}_{\mathrm{e}}\cup\mathscr{D}_{\mathrm{w}}$). Let us denote $\mathbf{e}_i$ and $\mathbf{w}_i$ the elements of the two merged dictionaries, respectively.

At each iteration, the algorithm can choose either a complex exponential or a wavelet function, and the update procedure depends on what type of function has been chosen. The algorithm will choose the function which extracts the highest amount of energy from the current residue.

3.1. Principles of the wavelet packet decomposition

Here, we are going to discuss some properties of the wavelet packet (WP) decomposition, which our wavelet-based dictionary $\mathscr{D}_{\mathrm{w}}$ is derived from. First of all, we restrict the wavelet-based dictionary to orthonormal wavelets in order to speed up the correlation updating procedure, as can be seen later. The dictionary $\mathscr{D}_{\mathrm{w}}$ is made up of those functions which give rise to the P-depth full wavelet packet decomposition. Note that $M_{\mathrm{w}}=P\cdot N$ is the size of the dictionary $\mathscr{D}_{\mathrm{w}}$. The inner products of the signal with the wavelet-based atoms in set $\mathscr{D}_{\mathrm{w}}$ lead to all the wavelet coefficients that can be considered in the P-depth full WP tree. These coefficients can be identified using three indexes, $\{s,p,k\}$, which indicate the subband at a given decomposition depth, the decomposition depth and the delay, respectively. The wavelet-based atoms can be expressed as follows:

$$w_{\{s,p,k\}}[n]=w_{\{s,p\}}[n-2^p k]. \tag{6}$$

Sequence $w_{\{s,p\}}[n]$ is the time-domain version of $W_{\{s,p\}}(z)$, which can be built directly from $G_0(z)$ and $G_1(z)$, the transfer functions of the low pass and high pass synthesis filters, respectively. These filters implement the inverse WP transform. Therefore, the function $W_{\{s,p\}}(z)$ can be expressed as follows:

$$W_{\{s,p\}}(z)=\prod_{d=0}^{p-1} G_{((\lfloor s/2^d\rfloor))_2}(z^{2^d}), \tag{7}$$

where $((l))_L$ denotes (l modulo L).

The only required correlations to implement matching pursuits are $\langle\mathbf{x},\mathbf{w}_{\{s,p,k\}}\rangle$ and $\langle\mathbf{w}_{\{s_1,p_1,k_1\}},\mathbf{w}_{\{s_2,p_2,k_2\}}\rangle$, according to expression (5). The first ones are obtained from the WP transform of $x[n]$. Instead, cross-correlations between atoms, which must be pre-calculated and stored, are computed taking into account that only atoms with heritage relation ($s_2=\lfloor s_1/(2^{p_1-p_2})\rfloor$) have to be considered when wavelet-based dictionaries built from orthonormal wavelets are used. The cross-correlations result in [15]

$$\langle w_{\{s_1,p_1,k_1\}}[n],w_{\{s_2,p_2,k_2\}}[n]\rangle = \begin{cases} \delta[k_2-k_1] & s_1=s_2,p_1=p_2,\\ 0 & s_2\neq\left\lfloor\dfrac{s_1}{2^{p_1-p_2}}\right\rfloor,\\ w_{\{s,p\}}[k_2-2^p k_1] & s_2=\left\lfloor\dfrac{s_1}{2^{p_1-p_2}}\right\rfloor, \end{cases} \tag{8}$$

where $p_1\geqslant p_2$ is supposed, $p=p_1-p_2$ and $s=((s_1))_{2^p}$. Therefore, according to (8), the iterative procedure to update correlations requires impulsive responses of the synthesis WP tree branches to be stored.

3.2. Properties of complex exponentials

Using a dictionary of complex exponentials $\mathscr{D}_{\mathrm{e}}$, only the frequency of each exponential function must be determined, which involves a significant reduction of the dictionary size [16]. As stated below, the projection onto the selected complex exponential contains the information of the phase. Furthermore, each sinusoidal function is a linear combination of two conjugated complex exponentials.

The functions that belong to the considered set can be expressed as follows:

$$\mathrm{e}_i[n]=\frac{1}{\sqrt{N}}\mathrm{e}^{\mathrm{j}(2\pi i/2L)n},\quad i=0,\ldots,L-1$$
$$n=0,\ldots,N-1. \tag{9}$$

The constant $1/\sqrt{N}$ is selected to obtain unit-norm functions, N is the length of the analysis frame, and $M_e = L$ the number of frequencies within the dictionary.

Due to the complex nature of these atoms, the residual at each iteration of the pursuit is calculated according to (10)

$$\begin{aligned}\mathbf{r}^{m+1} &= \mathbf{r}^m - \alpha_{i(m)}\mathbf{e}_{i(m)} - \alpha^*_{i(m)}\mathbf{e}^*_{i(m)} \\ &= \mathbf{r}^m - 2Re\{\alpha_{i(m)}\mathbf{e}_{i(m)}\}. \end{aligned} \tag{10}$$

Now, a conjugate sub-space is searched at each iteration of the algorithm [10], which does not change the principles of the algorithm, so that the atom which minimizes the residual energy is chosen at each iteration. The correlation update between the atoms $\mathbf{e}_i \in \mathscr{D}_e$ and the residue at each iteration is performed as follows:

$$\langle \mathbf{r}^{m+1}, \mathbf{e}_i \rangle = \langle \mathbf{r}^m, \mathbf{e}_i \rangle - \alpha_{i(m)} \langle \mathbf{e}_{i(m)}, \mathbf{e}_i \rangle - \alpha^*_{i(m)} \langle \mathbf{e}^*_{i(m)}, \mathbf{e}_i \rangle. \tag{11}$$

Owing to the nature of the atoms $\mathbf{e}_i \in \mathscr{D}_e$ (complex exponentials), the correlations required to implement matching pursuit can be efficiently computed by applying the fast Fourier transform (FFT). Thus, the initial correlations between the signal $\mathbf{x}$ and the atoms $\mathbf{e}_i \in \mathscr{D}_e$ are expressed as

$$\langle \mathbf{x}, \mathbf{e}_i \rangle = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x[n] \mathrm{e}^{-\mathrm{j}(2\pi i/2L)n} = \frac{1}{\sqrt{N}}\ X[i], \tag{12}$$

where $X[i]$ is the $2L$-length DFT of the input signal $x[n]$, and $L > N$ in order to assemble an over-complete dictionary. Note that $X[i]$ has complex nature, which implies both amplitude and phase information are included in this value. The initial correlations in (12) can be computed by applying the FFT algorithm, which implies that the signal $x[n]$ must be zero-padded for implementing the $2L$-length FFT.

Likewise, the cross-correlations between atoms $\mathbf{e}_i \in \mathscr{D}_e$ can be expressed as

$$\begin{aligned}\langle \mathbf{e}_{i(m)}, \mathbf{e}_i \rangle &= \frac{1}{N} \sum_{n=0}^{N-1} \mathrm{e}^{-\mathrm{j}(2\pi(i-i(m))/2L)n} \\ &= \frac{1}{N}\, U[((i - i(m)))_{2L}], \end{aligned} \tag{13}$$

$$\begin{aligned}\langle \mathbf{e}^*_{i(m)}, \mathbf{e}_i \rangle &= \frac{1}{N} \sum_{n=0}^{N-1} \mathrm{e}^{-\mathrm{j}(2\pi(i+i(m))/2L)n} \\ &= \frac{1}{N}\, U[((i + i(m)))_{2L}], \end{aligned} \tag{14}$$

where $U(i)$ is the $2L$-length DFT of the unit function $u[n]$. From (13) and (14), it is deduced that the cross-correlations between atoms $\mathbf{e}_i \in \mathscr{D}_e$ can also be calculated using the FFT algorithm. In this case, the $2L$-length FFT is applied to the unit function $u[n]$. This transform can be pre-computed and memory-stored for achieving a fast correlation updating.

Therefore, the use of matching pursuits with a dictionary composed of complex exponentials involves: (1) The initial correlations, that can be obtained by a $2L$-length FFT; (2) The cross-correlations between atoms, that only require a $2L$-length vector to be memory-stored.

3.3. Implementation of matching pursuit with the mixed dictionary

When dealing with a mixed dictionary composed of complex exponentials and wavelets ($\mathscr{D} = \mathscr{D}_e \cup \mathscr{D}_w$), matching pursuit must compute the weights $\{\boldsymbol{\alpha}^m_i, \boldsymbol{\alpha}^m_{\{s,p,k\}}\}$ corresponding to all dictionary elements $\{\mathbf{e}_i, \mathbf{w}_{\{s,p,k\}}\}$ at each iteration. At the first iteration, these weights are the correlations between the input signal and all the dictionary elements. The weights are computed by the Fourier transform or the wavelet packet transform, depending on whether they correspond to complex exponential or wavelet functions, respectively. Once the weights have been calculated, the algorithm chooses the optimum atom (the one that minimizes the residual energy), which can be either a complex exponential or a wavelet function.

Next, the correlation updating procedure must be implemented, which implies that the cross-correlations between atoms must be pre-computed and stored in advance. We have already presented cross-correlations between atoms of the same nature. Now, a detailed discussion is necessary about cross-correlations between complex exponentials and wavelets. The computation of correlations between atoms of different nature depends

on the type of atom selected at each iteration, giving rise to two different situations.

3.3.1. Correlation between a complex exponential and a wavelet function when the complex exponential is chosen

In this case, although the optimum atom is complex, the function subtracted from the signal is real, as can be seen in Eq. (10). The correlation updating procedure has to pre-compute the cross-correlations between the optimum atom $\mathbf{e}_{i(m)}$ and all the wavelet functions $\mathbf{w}_{\{s,p,k\}} \in \mathscr{D}_{\mathrm{w}}$, and is implemented following the scheme of expression (11). This computation can be expressed by the DFT due to the nature of complex exponentials:

$$\langle e_{i(m)}[n], w_{\{s,p,k\}}[n]\rangle = \sum_{n=0}^{N-1} \frac{1}{\sqrt{N}} \mathrm{e}^{\mathrm{j}(2\pi i(m)/2L)n} w_{\{s,p,k\}}[n] = \frac{1}{\sqrt{N}} W^{*}_{\{s,p,k\}}[i(m)], \tag{15}$$

where $W_{\{s,p,k\}}[i(m)]$ is the value of the $2L$-length DFT of $w_{\{s,p,k\}}[n]$ at the normalized frequency $i(m)/2L$. Therefore, the $2L$-length DFT of each wavelet function $w_{\{s,p,k\}}[n]$ has to be memory-stored for the correlation updating procedure. Since the size of the wavelet-based dictionary $\mathscr{D}_{\mathrm{w}}$ is $M_{\mathrm{w}} = N \cdot P$, the number of $2L$-length DFT that must be memory-stored is $N \cdot P$.

However, we can save memory by taking into account $w_{\{s,p,k\}}[n] = w_{\{s,p\}}[n - 2^p k]$, which involves storing only the $2L$-length DFT of $w_{\{s,p\}}[n]$. The number of $2L$-length DFT to be stored is now reduced to $2^{P+1} - 2$. The remaining correlations can be computed using the time-delay properties of the DFT:

$$\begin{aligned}\langle \mathrm{e}_{i(m)}[n], w_{\{s,p\}}[n - 2^p k]\rangle &= \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} \mathrm{e}^{\mathrm{j}(2\pi i(m)/2L)n} w_{\{s,p\}}[n - 2^p k] \\ &= \frac{1}{\sqrt{N}} \mathrm{e}^{\mathrm{j}(2\pi i(m)/2L)2^p k} \\ &\quad \times \sum_{l=0}^{N-1-2^p k} \mathrm{e}^{\mathrm{j}(2\pi i(m)/2L)l} w_{\{s,p\}}[l],\end{aligned} \tag{16}$$

where $l = n - 2^p k$, and $w_{\{s,p\}}[l] = 0$, $\forall l < 0$, is considered. The last result can be related with the $2L$-length DFT of $w_{\{s,p\}}[n]$ as follows:

$$\begin{aligned}\langle \mathrm{e}_{i(m)}[n], w_{\{s,p\}}[n - 2^p k]\rangle &= \frac{1}{\sqrt{N}} \mathrm{e}^{\mathrm{j}(2\pi i(m)/2L)2^p k} \Big(W^{*}_{\{s,p\}}[i(m)] \\ &\quad - \sum_{l=N-2^p k}^{N-1} \mathrm{e}^{\mathrm{j}(2\pi i(m)/2L)l} w_{\{s,p\}}[l] \Big),\end{aligned} \tag{17}$$

where $W_{\{s,p\}}[i(m)]$ is the value of the $2L$-length DFT of $w_{\{s,p\}}[n]$ at the normalized frequency $i(m)/2L$. The algebraic sum in (17) must be computed for all k values, and it can be calculated by a complex digital filter. If this term is computed for consecutive values of k, additional complexity reduction can be obtained, resulting in $N - 2^p$ complex multiplications for all k values. Additionally, the first exponential in the same expression represents a complex multiplication each 2^p samples. Summarizing, the number of multiplications to compute (17) is $N - 2^p + N/2^p$ for each wavelet function $w_{\{s,p\}}[n]$. However, note that functions $w_{\{s,p\}}[n]$ are time-localized in such a way that they have many zero values, which implies a computational cost reduction.

As a conclusion, when a complex exponential is chosen the updating procedure needs: (1) to store the $2L$-length DFT of each wavelet function $w_{\{s,p\}}[n]$, and (2) to compute the effect of the $2^p k$ delay for each wavelet function ($w_{\{s,p,k\}}[n] = w_{\{s,p\}}[n - 2^p k]$).

3.3.2. Correlation between a complex exponential and a wavelet function when the wavelet function is chosen

The optimum atom is now $\mathbf{w}_{\{s,p,k\}(m)}$ and we intend to compute the cross-correlation between this function and all atoms $\mathbf{e}_i \in \mathscr{D}_{\mathrm{e}}$:

$$\langle w_{\{s,p,k\}(m)}[n], \mathrm{e}_i[n]\rangle = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} w_{\{s,p,k\}(m)}[n] \mathrm{e}^{-\mathrm{j}(2\pi i/2L)n} = \frac{1}{\sqrt{N}} W_{\{s,p,k\}(m)}[i], \tag{18}$$

where $W_{\{s,p,k\}(m)}[i]$ is the value of the $2L$-length DFT of $w_{\{s,p,k\}}[n]$ at the normalized frequency $i/2L$. In this case, the $2L$-length DFT of each wavelet function $w_{\{s,p,k\}}[n]$ has to be memory-stored for implementing the correlation updating

procedure. As in the case of subsection 3.3.1, the memory requirements can be reduced by applying $w_{\{s,p,k\}}[n] = w_{\{s,p\}}[n - 2^p k]$. In this way, the correlation in (18) can be simplified as

$$\begin{aligned}\langle w_{\{s,p,k\}(m)}[n], \mathrm{e}_i[n]\rangle &= \frac{1}{\sqrt{N}} \mathrm{e}^{-\mathrm{j}(2\pi i/2L)2^p k(m)} \times \sum_{l=0}^{N-1-2^p k(m)} w_{\{s,p\}(m)}[l]\mathrm{e}^{-\mathrm{j}(2\pi i/2L)l} \\ &= \frac{1}{\sqrt{N}}\mathrm{e}^{-\mathrm{j}(2\pi i/2L)2^p k(m)}\Big(W_{\{s,p\}(m)}[i] - \sum_{l=N-2^p k(m)}^{N-1} w_{\{s,p\}(m)}[l]\mathrm{e}^{-\mathrm{j}(2\pi i/2L)l}\Big), \end{aligned} \tag{19}$$

where $W_{\{s,p\}(m)}[i]$ is the value of the $2L$-length DFT of $w_{\{s,p\}(m)}[n]$ at the normalized frequency $i/2L$, $l = n - 2^p k$ and $w_{\{s,p\}}[l] = 0,\ \forall l < 0$, is considered. The problem of computing the last sum in (19) is now different. We must calculate this term for $i = 0, \ldots, L-1$ and for the value $k(m)$ corresponding to the chosen wavelet atom. Depending on the value of $k(m)$, this computation can be optimally obtained by FFT or complex digital filter-based methods. Furthermore, note that most of the functions $w_{\{s,p\}}[n]$ have zero values from $N - 2^p k(m)$ to $N-1$.

Therefore, when a wavelet function is chosen, the correlation updating procedure needs: (1) storing the $2L$-length DFT of each wavelet function $w_{\{s,p\}}[n]$, and (2) computing the effect of the $2^p k(m)$ delay for the chosen wavelet function ($w_{\{s,p,k\}(m)}[n] = w_{\{s,p\}(m)}[n - 2^p k(m)]$).

Finally, when a mixed dictionary composed of complex exponentials and wavelets ($\mathscr{D} = \mathscr{D}_\mathrm{e} \cup \mathscr{D}_\mathrm{w}$) is used, the memory requirements for the correlation updating procedure are:

(1) $2L$-length DFTs of the complex exponentials $\mathbf{e}_i$.
(2) Impulsive responses of the synthesis WP tree branches $\mathbf{w}_{\{s,p\}}$.
(3) $2L$-length DFTs of the wavelet functions $\mathbf{w}_{\{s,p\}}$.

The $2L$-length DFT and the P-depth WP transform of the signal $x[n]$ are computed to initialize the correlations between the input signal and the dictionary elements. The required number of multiplications per iteration is due to the correlation updating procedure of expression (5). One multiplication per atom is needed to multiply the weight $\alpha_{i(m)}$ corresponding to the optimum atom by the cross-correlations. Besides, the proposed implementation requires additional computation to obtain the cross-correlations, which is detailed as follows:

(1) When a complex exponential $\mathbf{e}_i(m)$ is the optimum atom, the cross-correlation with each wavelet function $\mathbf{w}_{\{s,p,k\}}$ is computed, according to Eq. (17), by the DFT of the function $\mathbf{w}_{\{s,p\}}$. Therefore, the number of multiplications is $N - 2^p + N/2^p$ for each wavelet function $\mathbf{w}_{\{s,p\}}$.
(2) When a wavelet function $\mathbf{w}_{\{s(m),p(m),k(m)\}}$ is the optimum atom, the cross-correlation with each complex exponential $\mathbf{e}_i$ is computed according to Eq. (19). Now, the number of multiplications depends on the delay $k(m)$ of the optimum atom. At the worst case, corresponding to $k(m) = N - 2^p(m)$, the complexity associated to expression (19) is the same than that of the $2L$-length FFT of the optimum atom.

4. Experimental results

We first intend to illustrate the advantages of the proposed mixed dictionary against two different dictionaries that operate in cascade for matching pursuit-based parametric audio coding. For comparison purposes, matching pursuit is performed under three different approaches: (1) using a single dictionary composed of complex exponentials and wavelets, (2) cascading a dictionary of complex exponentials followed by another of wavelets, (3) cascading a dictionary of wavelets followed by another of complex exponentials.

Two examples are taken for illustrating such advantages. Fig. 1 shows an audio fragment where a signal onset appears taken from the castanet excerpt. As can be seen, the best discrimination between tonal and transient features is accomplished by the first approach

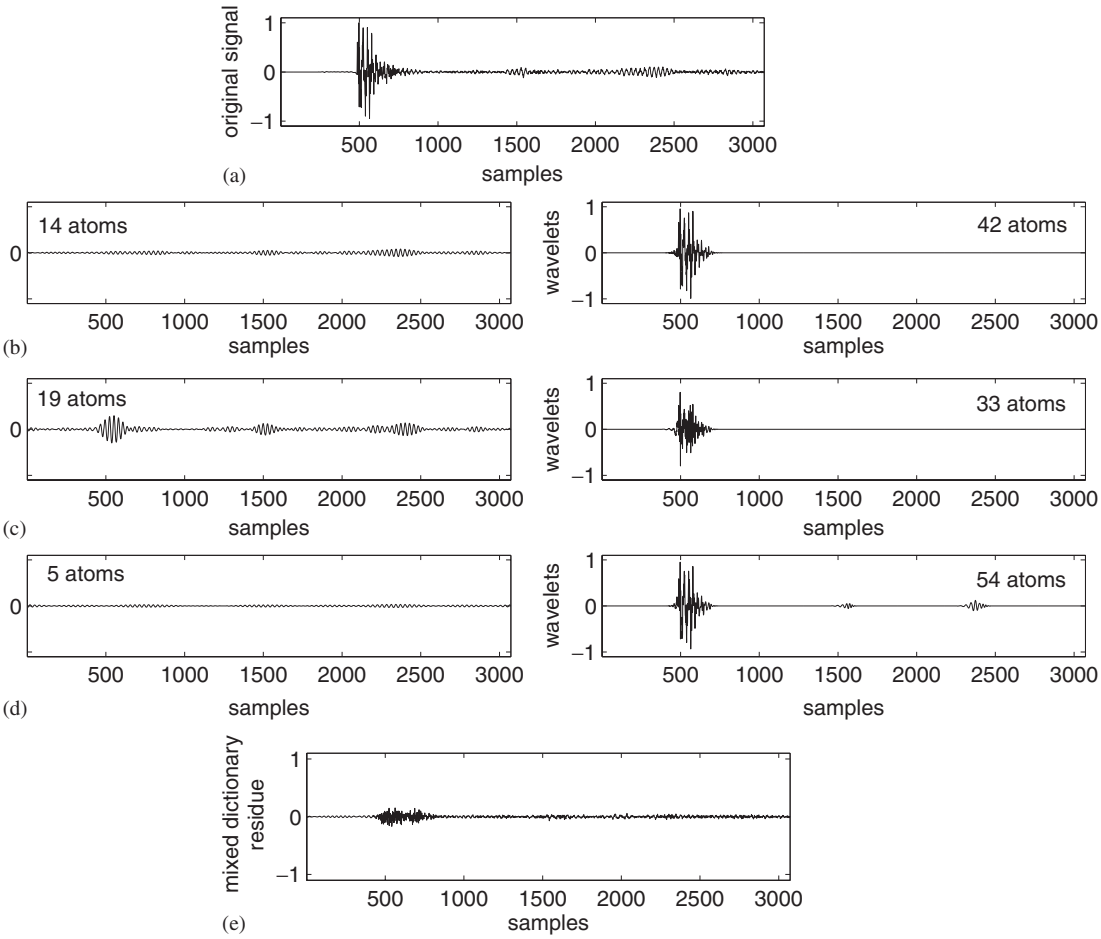

Fig. 1. (a) Audio fragment containing a signal onset taken from the castanet excerpt. (b) Sinusoids and wavelet atoms extracted from the audio frame by the first approach (mixed dictionary). (c) Sinusoids and wavelet atoms extracted by the second approach (sinusoids followed of wavelets). (d) Sinusoids and wavelet atoms extracted by the third approach (wavelets followed of sinusoids). (e) Residue obtained by the mixed dictionary.

(matching pursuit with the mixed dictionary). Cascading complex exponentials followed by wavelets (second approach) gives rise to both pre-echo and transient smoothing, while cascading wavelets followed by complex exponentials (third approach) involves extracting too many transient components.

The stopping criterion for all cases is the following: matching pursuit is halted when an atom extracts from the residue less than 2% of the total energy of the residue. This value is chosen to obtain a residue which has stochastic properties, as can be seen in Fig. 1, because tonal (or transient) components have to be removed from the residue in order to avoid artifacts in the synthesized noise [17].

Fig. 2 shows an audio fragment containing a micro-transient taken from the glockenspiel excerpt. The structure of Fig. 2 is similar to that of Fig. 1. The mixed dictionary obtains again the best decomposition. The micro-transient synthesized by the second approach is less sharp than the one extracted by the first approach. Furthermore, the third approach does not successfully represent the

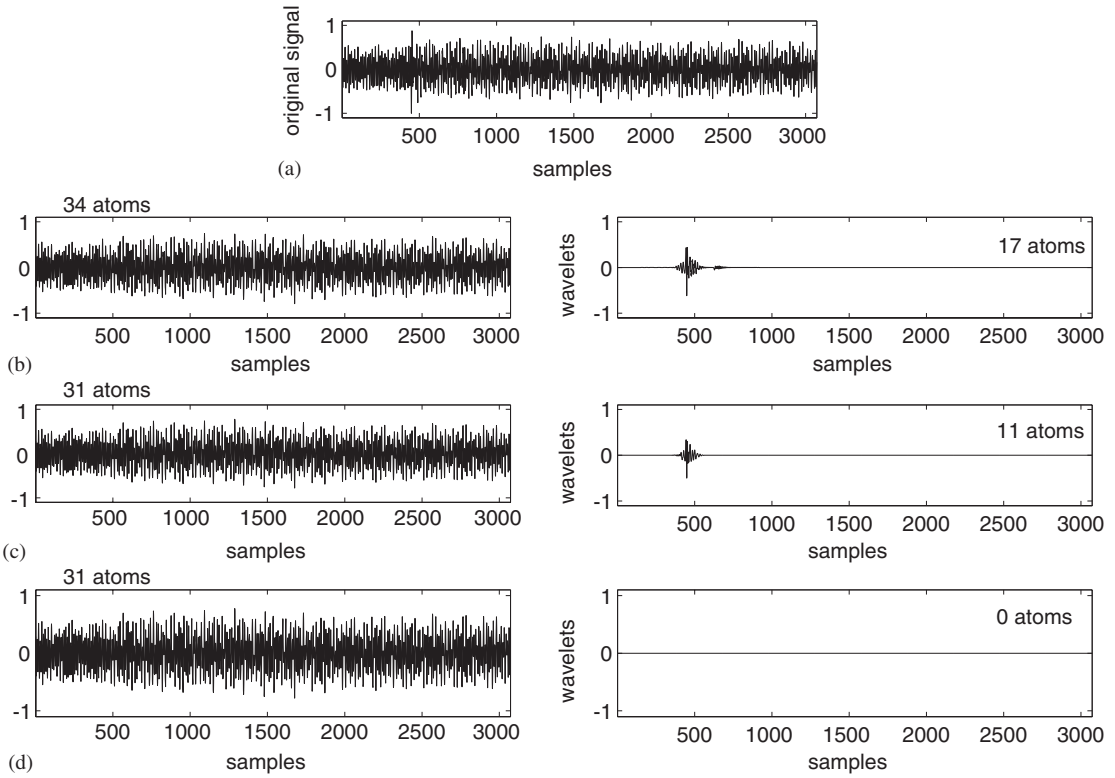

Fig. 2. (a) Audio fragment containing a micro-transient. (b) Sinusoids and wavelet atoms extracted from the audio frame by the first approach (mixed dictionary). (c) Sinusoids and wavelet atoms extracted by the second approach (sinusoids followed of wavelets). (d) Sinusoids and wavelet atoms extracted by the third approach (wavelets followed of sinusoids).

micro-transient because no wavelet function is extracted from the signal.

Figs. 1 and 2 show two signals containing a transient and a micro-transient, respectively, and in these cases the first approach gives the best results.

To assess the perceptual behavior of matching pursuit with the mixed dictionary, comparison with the third approach is proposed, since most of the STN-based parametric audio coder performs transient modeling followed by sinusoidal modeling. CD-quality one-channel audio and speech signals taken from the set of excerpts used in the MPEG standardization activities [18] are chosen for testing. The analysis/synthesis was done on a frame-by-frame basis using a 50% overlap 23-ms Hanning window ($N = 1024$). The number of complex exponentials in $\mathscr{D}_e$ is $L = 4096$, the decomposition depth of the WP tree is $P = 4$, and 32-coefficients filters that generate orthonormal Daubechies wavelets with maximum number of vanishing moments are used. The results do not change significantly by increasing the decomposition depth or changing the family of the wavelet filters [15].

We performed a subjective listening test using the double blind triple stimulus methodology, in which signal triplets OAB were presented to ten experienced listeners. Here, O is the original signal; A and B are the modeled signals using the first and third approaches, respectively. For each test signal, A or B were randomly presented three times to each listener, together with the original. The listener was asked to indicate which signal

Table 1
Preference for mixed dictionary-vs-cascaded dictionaries (%)

Excerpt	Preference (%)
Suzanne Vega	55
German male speech	60
English female speech	70
Harpsichord	100
Castanets	100
Pitch pipe	52
Bagpipes	46
Glockenspiel	100
Plucked strings	70
Trumpet solo	56
Orchestra piece	60
Contemporary pop	100

(A or B) is closer to the original. The results averaged of all the listeners are shown in Table 1.

Mixed dictionary-based matching pursuit is usually preferred by listeners for audio signals with high transient content. In such sense, artifacts like "clicks" are avoided for audio frames containing sharp attacks, and pre-echo distortion is avoided for frames containing micro-transients due to residue spreading at the time interval where the micro-transient is located. For nearly steady audio signals, there is almost no perceptual difference between the two approaches. The results in Table 1 come to confirm the comments above.

Next, we intend to reveal the ability of matching pursuits with a mixed dictionary composed of complex exponentials and wavelets for audio compression by integrating this transient + sinusoidal joint modeling tool into a parametric audio coder [8]. The coded information that is sent to the decoder can be organized as:

- Amplitude, phase and frequency for each sinusoid.
- Amplitude, decomposition depth, subband at a given decomposition depth, and delay for each wavelet atom.
- Temporal and spectral envelopes of the residue modeled as in [8].

Note that all the information, except amplitudes and phase, is already quantized when each atom is selected from the mixed dictionary. Besides, psychoacoustic principles are taken into account to quantize the amplitudes [16].

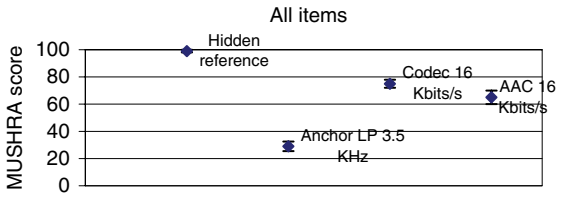

Fig. 3. MUSHRA listening test results showing mean grade and 95% confidence interval.

Fig. 3 shows subjective results comparing MPEG2/4-AAC [19] at 16 Kbits/s with the parametric audio coder proposed in [8]. Listening tests employed MUSHRA [20] methodology, including a hidden reference and a low-pass filtered anchor with 3.5 KHz bandwidth. Ten experienced listeners conducted the test with the audio material listed in Table 1, using headphones.

It was found that the parametric audio coder obtains, on average, better subjective results than MPEG2/4-AAC at 16 Kbits/s. Furthermore, the parametric audio coder outperformed MPEG2/4-AAC for all the excerpts, except for the orchestra piece. Good audio quality is assured in all cases and is slightly higher in musical signals than in speech signals.

Finally, Table 2 shows the bit rates obtained by the parametric audio coder proposed in [8] when matching pursuit makes use of the proposed mixed dictionary. The same excerpts listed in Table 1 are here considered. Table 2 not only shows the global bit rates, but also the bit rates corresponding to sinusoids, transients and noise. The bit rate corresponding to the header is about 0.1 Kbits/s. The overhead information approximately represents 45% of the global bit rate, while the remaining 55% corresponds to quantized values.

From the results in Fig. 3 and Table 2, we can say that very low bit rate good quality audio coding is achieved when matching pursuit with the proposed mixed dictionary is implemented in a parametric audio coder [8]. Bit rates close to 16 Kbits/s are obtained for all the test signals.

Table 2
Bit rates obtained when using the proposed mixed dictionary

Excerpt	Bit rates (Kbits/s)	Sinusoids	Transients	Noise
Suzanne Vega	16.5	12.1	1.0	3.3
German male speech	16.7	12.5	0.8	3.3
English female speech	18.0	13.9	1.0	3.0
Harpsichord	14.6	11.7	0.2	2.6
Castanets	18.6	11.8	4.3	2.4
Pitch pipe	11.9	8.2	0.1	3.5
Bagpipes	13.2	9.2	0.2	3.7
Glockenspiel	6.9	3.8	0.7	2.3
Plucked strings	16.9	13.9	0.1	2.8
Trumpet solo	16.4	13.0	0.4	2.9
Orchestra piece	15.3	12.8	0.2	2.2
Contemporary pop	18.7	15.6	0.2	2.8

5. Conclusions

The paper deals with the application of matching pursuit with a mixed dictionary composed of complex exponentials and wavelets for transient + sinusoidal modeling in parametric audio coding, as an alternative to matching pursuit with two dictionaries operating in cascade. Using the mixed dictionary, better subjective quality of the decoded audio signals is achieved. The price to pay is an increase of complexity, which does not make real-time implementation with low–medium cost DSP platforms possible.

Mixed dictionary-based matching pursuit has been successfully applied to transient + sinusoidal joint modeling of audio signals, showing that synthesized transients are precisely located at the part of the audio signal where the energy burst is. This fact is responsible for the better quality of signals with transients. Experimental results show that the proposed signal processing tool can be incorporated into a parametric audio coder with very good performance at 16 Kbits/s, even better than MPEG2/4-AAC at the same rate. Good audio quality is assured for all excerpts.

Audio demonstrations are made available online by anonymous ftp: ftp://himilce.ujaen.es/varios/muestras/

References

[1] M. Ali, Adaptive signal representation for audio coding, Ph.D. Thesis, Department of Electrical Engineering, University of Minnesota, 1995.
[2] H. Purnhagen, B. Edler, C. Ferekidis, Object-based analysis/synthesis audio coder for very low bit rates, Preprint 4747, 104th AES Convention, Amsterdam, The Netherlands, May 1998.
[3] S. Levine, J. Smith, A sines + transients + noise audio representation for data compression and time/pitch scale modifications, Preprint 4781, 105th AES Convention, San Francisco, USA, September 1998.
[4] T.S. Verma, A perceptually based audio signal model with application to scalable audio compression, Ph.D. Thesis, Standford University, 1999.
[5] H. Purnhagen, N. Meine, HILN—the MPEG-4 parametric audio coding tools, in: Proceedings of ISCAS, Geneva, Italy, vol. 3, 2000, pp. 201–204.
[6] B. den Brinker, E. Schuijers, W. Oomen, Parametric coding for high quality audio, Paper 5554, 112th AES Convention, Munich, Germany, May 2002.
[7] ISO/IEC MPEG, AVC test results validate superior technology, Technical Report N6085, 2003.
[8] P. Vera-Candeas, N. Ruiz-Reyes, M. Rosa-Zurera, J. Curpian-Alonso, P.J. Reche-Lopez, Signal-adaptive parametric modeling for high quality low bit rate audio coding, Preprint 6176, 116th AES Convention, Berlin, Germany, May 2004.
[9] T.S. Verma, T.H.Y. Meng, A 6 kbps to 85 kbps scalable audio coder, in: Proceedings of ICASSP, Istanbul, Turkey, 2000, pp. 877–880.
[10] M.M. Goodwin, Adaptative signal models: theory, algorithms and audio applications, Kluwer Academic Publishers, Dordrecht, 1998.
[11] S.S. Chen, Basis pursuit, Ph.D. Thesis, Department of Statistics, Standford University, 1995.
[12] H. Lee, D.P. Sullivan, T.H. Huang, Improvement of discrete band-limited signal extrapolation by iterative subspace modification, Proceedings of ICASSP'87, vol. 3, 1987, pp. 1569–1572.
[13] S. Mallat, Z. Zhang, Matching pursuit with time-frequency dictionaries, IEEE Trans. Signal Process. 41 (December 1993) 3397–3415.
[14] G. Davis, Adaptive nonlinear approximations, Ph.D. Thesis, Department of Mathematics, New York University, 1994.
[15] P. Vera-Candeas, N. Ruiz-Reyes, M. Rosa-Zurera, D. Martinez-Munoz, F. Lopez-Ferreras, Transient modelling by matching pursuits with a wavelet dictionary for parametric audio coding, IEEE Signal Process. Lett. 11 (3) (March 2004).
[16] P. Vera-Candeas, N. Ruiz-Reyes, M. Rosa-Zurera, F. Lopez-Ferreras, J. Curpian-Alonso, New matching pursuit based sinusoidal modelling method for audio coding, IEE Proceedings—Visual Image Signal Process. 151 (1) (February 2004).

[17] N.H. van Schijndel, M. Gomez, R. Heusdens, Towards a better balance in sinusoidal plus stochastic representation, IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, 2003, pp. 197–200.

[18] ISO/MPEG, Call for proposal for new tools for audio coding, ISO/IEC JTC1/SC29/WG11, MPEG2001/N3793, January 2001.

[19] ISO/IEC MPEG, Information technology—generic coding of moving pictures and associated audio information, Part 7: Advanced audio coding (AAC), International Standard, 13818-7, 1997.

[20] ITU-R, Method for the subjective assessment of intermediate quality level of coding systems (MUSHRA), ITU-R Recommendation, BS. 1534, 2001.