OBJECT CODING OF HARMONIC SOUNDS USING SPARSE AND STRUCTURED REPRESENTATIONS

Grégory Cornuz[1], *Emmanuel Ravelli*[1,2], *Pierre Leveau*[1,2], *Laurent Daudet*[1]

[1]Institut Jean Le Rond d'Alembert, LAM team
Université Pierre et Marie Curie - Paris 6
11, rue de Lourmel
75 015 Paris - France
`lastname@lam.jussieu.fr`

[2]TSI department
GET-ENST (Télécom Paris)
37-39, rue Dareau
75 014 Paris - France
`firstname.lastname@enst.fr`

ABSTRACT

Object coding allows audio compression at extremely low bit-rates, provided that the objects are correctly modelled and identified. In this study, a codec has been implemented on the basis of a sparse decomposition of the signal with a dictionary of Instrument-Specific Harmonic atoms. The decomposition algorithm extracts "molecules" i.e. linear combinations of such atoms, considered as note-like objects. Thus, they can be coded efficiently using note-specific strategies. For signals containing only harmonic sounds, the obtained bitrates are very low, typically around 2 kbs, and informal listening tests against a standard sinusoidal coder show promising performances.

1. INTRODUCTION

Audio coding has traditionally evolved in two directions, depending on the target bitrate. At high rates, state-of-the-art audio coding is transform-based (e.g. MPEG4-AAC and MPEG4-TwinVQ [1]). At lower rates, parametric coders perform slightly better. MPEG4-SSC [2], based on a sinusoids+transients+noise model, outperforms MPEG4-AAC at 24kits; but is not designed for lower bitrates. MPEG4-HILN [3], based on a harmonics+sinusoids+noise model, works at lower bitrates but its performance appears to be very signal dependant and on average it is comparable to MPEG4-AAC at 16kbits and MPEG4-TwinVQ at 6kbits ; the benefit compared to the transform-based coders is that HILN allows additional functionality such as speed and pitch modifications at the synthesis.

For more flexibility on the type of possible transform-domain sound modifications (for instance the modification of timbre parameters of a single instrument in a polyphonic mixture), it is necessary to go one step further in the understanding of the contents of the audio file; this is the goal of so-called object-based audio coding, which fits well in the general context of MPEG4. Instead of coding transform coefficients or parameters of a low-level model, object audio coders consider higher-level "sound objects", consisting ideally of individual notes or chords. In [4], pitched sound objects consisting of the sum of harmonic sinusoidal partials are effciently estimated using a statistical approach; the resulting coder appears to perform better than transform and parametric coders on solo or duo of harmonic instruments at 8kbit/s and 2 kbit/s. However, this approach requires extensive computational resources which makes them unpractical for most applications.

In this paper, we present a novel object-based coding, which allows the computation of objects in a reasonable computational time. First, the sound is decomposed on a dictionary of instrument-dependent atoms, or groups of atoms ("molecules"), with a modified version of the matching pursuit algorithm. Then, the atoms parameters are encoded with variable precision. The main benefit of this approach is that very low bit-rates can be achieved at full bandwidth, while keeping an acceptable sound quality for most sound examples. The price to pay, besides computational complexity, is the necessity to store the full database of atoms at both encoder and decoder, a requirement that is more and more acceptable given the increase in storage capacities. This paper is organized as follows : in section 2, we describe the decomposition process into sound objects. In section 3, we detail how we encode the extracted sound objects. Finally, preliminary results are given in section 4.

2. DECOMPOSITION ALGORITHM

2.1. Signal Model

2.1.1. Instrument Specific Harmonic Atoms

The signal is modelled as a linear combination of N harmonic atoms $h_{s_n,u_n,f_{0_n},A_n,\Phi_n}$ parameterized in terms of scale s_n (atom duration), time localisation u_n, fundamental frequency f_{0_n}, fundamental chirp rate c_{0_n}, partials amplitudes $A_n = \{a_{m,n}\}_{m=1:M}$ and partials phases $\Phi_n = \{\phi_{m,n}\}_{m=1:M}$:

$$x(t) = \sum_{n=1}^{N} \alpha_n \, h_{s_n,u_n,c_{0_n},f_{0_n},A_n,\Phi_n}(t). \qquad (1)$$

Each harmonic atom can be written as

$$h_{s,u,f_0,c_0,A,\Phi}(t) = \sum_{m=1}^{M} a_m \, e^{j\phi_m} g_{s,u,m.f_0,m.c_0}(t). \qquad (2)$$

The amplitudes of the M partials are constrained to $\sum_{m=1}^{M} a_m^2 = 1$ and the signal corresponding to each partial is given by a *Gabor* atom normalised to unit energy

$$g_{s,u,f} = w \quad \frac{t-u}{s} \quad e^{2j\pi ft} \qquad (3)$$

with w as a weighting window.

When partials amplitudes are learned from a database (see 2.3.1), these atoms are called Instrument Specific Harmonic (ISH) atoms. Each amplitude vector A is then associated with a class (in our case an instrument) and a discrete pitch value. Generally, several vectors are used for each class and each pitch value.

2.1.2. *Instrument Specific Harmonic Molecules*

The long-term structures, such as music notes, cannot be efficiently modelled with a single ISH atom. However, building sets of ISH atoms (named *molecules*) can overcome this issue. The constraints for atoms to belong to a single molecule are the following:

- the atoms span a range of time locations u, with exactly one atom per location,
- all atoms come from the same instrument,
- the log-variation of fundamental frequency between any two consecutive atoms is bounded by a threshold D:

$$|\Delta \log f_0| \leq D \tag{4}$$

2.2. Decomposition algorithms

2.2.1. *The Matching Pursuit Algorithm*

Given a ISH dictionary, the problem becomes that of decomposing the signal as a collection of molecules of ISH atoms from this dictionary. A popular and efficient method to achieve atomic decompositions is the Matching Pursuit algorithm [5]. It can be modified for molecular decompositions [6, 7]. The Matching Pursuit algorithm proceeds as follows:

1. The correlations between the signal and all the atoms h of the dictionary are computed using inner products $\langle x, h\rangle = \sum_{t=1}^{T} x(t)\, h(t)$.
2. The atom h that has the largest absolute correlation $|\langle x, h\rangle|$ with the signal is selected, then subtracted from the signal with a weighting coefficient $\alpha = \langle x, h\rangle$.
3. Correlations are updated on the residual signal, and the algorithm is iterated to step 2 until the stopping condition is satisfied. This condition can be a target Signal-to-Residual energy Ratio (SRR), or a fixed number of iterations.

2.2.2. *Molecular Algorithm*

The algorithm that is here briefly introduced is fully described in [8]. Its flowchart is presented on Figure 1. Its main feature is to iteratively extract molecules of ISH atoms using a Matching Pursuit algorithm that has been modified as follows :

- **Best atom path selection:** an atom path is selected in instrument-specific time-pitch planes using dynamic programming. The search zone of this path is delimited around a *seed* atom: the atom that is the most correlated with signal. A threshold on the atom weights α_n is set to avoid the selection of low-amplitude atoms, and as a consequence to reduce the amount of data to encode.
- **Atom parameters tuning on the path**: the fundamental chirp rate c_0 is estimated jointly with the refinement of the fundamental frequency f_0 using a maximization of the inner product $|\langle x, h\rangle|$ with regard to f_0 and c_0. The partial phases ϕ_m of each atom of the molecule are computed using the following formula:

$$e^{j\phi_m} = \frac{\langle x, g_{s,u,m.f_0,m.c_0}\rangle}{|\langle x, g_{s,u,m.f_0,m.c_0}\rangle|}. \tag{5}$$

- **Atom weights optimization**: the respective weights of each atom are re-estimated using an orthogonal projection of the signal on the subspace corresponding to the atoms of the molecule.

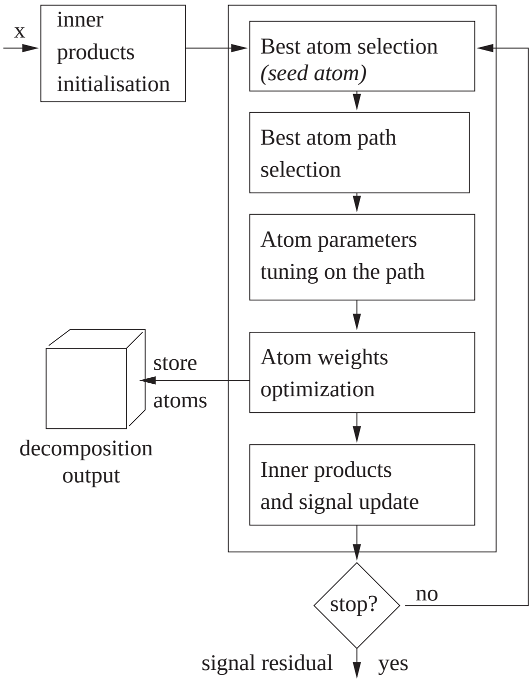

x
inner products initialisation
Best atom selection
(seed atom)
Best atom path selection
Atom parameters tuning on the path
Atom weights optimization
store atoms
decomposition output
Inner products and signal update
stop?
no
signal residual
yes

Figure 1: Flow chart of the algorithm for decomposing a signal into molecules of ISH atoms

2.3. Sampling the dictionary

In practical applications, the search step can only be performed on a finite number of atoms. Thus, one has to sample the dictionary by making the atom parameters s, u and f_0 discrete:

- The scale s often spans a small set of powers of 2.
- The time localisation u is typically set to equally spaced time bins, with a time shift Δu set to a fraction of the atom scale.
- The fundamental frequency f_0 is sampled logarithmically. This is a noticeable difference with the Harmonic MP algorithm [6], where fundamental frequencies are sampled linearly.

The amplitude vectors A are already a discrete set of vectors and the phase vectors are estimated using Equation 5.

2.3.1. *Learning the model*

For the following experiments, the vectors of partials amplitudes $\{A_{i,p,k}\}_{k=1\ldots K}$ are learned for each instrument/pitch class $\mathcal{C}_{i,p}$ on isolated notes from three databases: the RWC Musical Instrument Sound Database [9], IRCAM Studio On Line [10] and the University of Iowa Musical Instrument Samples [11]. We select five instruments producing harmonic notes: oboe (Ob), clarinet (Cl), cello (Co), violin (Vl) and flute (Fl).

For each isolated note signal, the time frame with maximal energy is computed and all the subsequent time frames whose energy lies within a certain threshold of this maximum are selected. This relative threshold is set to a ratio of 0.05 in the following. The partials amplitudes are computed on each of these training frames by

$$a_m = \frac{|\langle x, g_{s,u,m\times f_0,m\times c_0}\rangle|}{\sum_{m'=1}^{M} |\langle x, g_{s,u,m'\times f_0,m'\times c_0}\rangle|^2}^{1/2} \qquad (6)$$

where f_0 and c_0 are tuned in order to maximize the SRR on this frame, using the same optimisation method as in the parameter tuning step. The vector of amplitudes is then associated to the pitch class p that is the closest to f_0. The resulting number of vectors per instrument and per pitch class are indicated in Table 1.

Inst.	N_{train}	N_{train} per pitch
Ob	5912	169
Cl	9048	193
Co	13868	285
Vl	37749	700
Fl	13216	330

Table 1: Total number of training time frames per instrument and average number per pitch class.

The size of the dictionary varies linearly as a function of the number of amplitude vectors. Since the number of vectors is too large to ensure computationally tractable decompositions, we chose to reduce the number of vectors by vector quantization: K amplitude vectors are kept for each class $\mathcal{C}_{i,p}$ using the k-means algorithm with the Euclidean distance.

3. PARAMETERS CODING

We use a simple scheme where a representation is first estimated from the signal (see previous section) and then the parameters of the representation are quantized and coded a posteriori. At the decoder, the quantized parameters are decoded and used to synthesize a new signal.

Two properties of the representation allows efficient coding at very low bit rate. Firstly, the molecular algorithm builds “objects”, composed by a succession of atoms. The parameters of the atoms which belongs to the same molecule are highly correlated and thus can be efficiently coded. Secondly, due to the greedy nature of the molecular algorithm some parameters are already quantized before the coding stage; these parameters are consequently coded without any loss using entropy coding.

In the following, we list all the parameters of the model and the method we have chosen to code them.

- The scale s_n is constant and thus is not coded.
- The time localisation u_n is on a grid with a step size $s_n/2$. Only the absolute position of the first atom of a molecule is coded, the positions of the following atoms are then the consecutive values on the grid. The only additional parameter required by the decoder is the number of atoms that belong to the molecule.
- The fundamental frequency f_{0_n} of every atom is coded in its crude version (before the atom parameters tuning stage, see previous section). For the atoms of a molecule except the first one, we compute differences between consecutive values of the fundamental frequency, and the resulting values are entropy coded.
- The weight α of the first atom of a molecule is coded using a standard uniform quantizer + entropy coding approach [12]. The weights of the next atoms are coded using differential coding and uniform quantization.
- The amplitude of the partials A_n are already vector quantized. We then simply transmit the index of the corresponding vector in the dictionary. The index is composed by: the pitch class (crude version of the fundamental frequency, already coded) + the instrument class (coded one time for each molecule) + the index in the table of the corresponding pitch/instrument class. The index is entropy coded.
- We do not code the fundamental chirp rate c_{0_n} as as we found that this parameter is not perceptually relevant enough given the necessary bit budget needed to code it.
- The phases are not coded. We use an alternative approach where the phases are interpolated at the decoder to ensure a continuity between the partials of the succesive atoms.

4. EXPERIMENTS

The coder is evaluated on 5 solos (clarinet, cello, flute, oboe, violin) and 4 duos (clarinet/flute, cello/flute, cello/violin, flute/flute), extracted from commercial CDs (hence having no relationship with the single notes database used for learning).

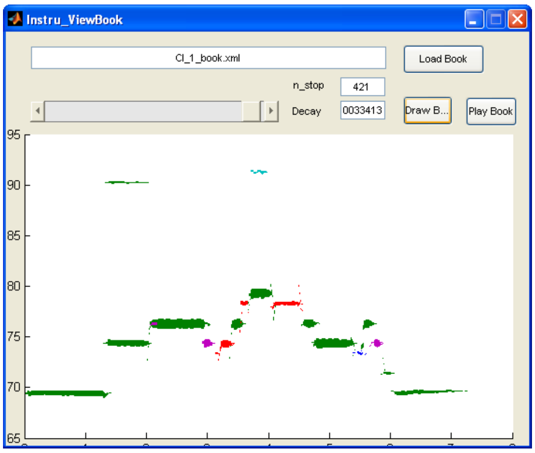
The two steps of the coding process, namely the signal decomposition and the parameters coding, have been performed with the following parameters:

- **Sampling parameters:** for our application, the choice of a single scale s corresponding to a duration of about 50 ms is sufficient. It is long enough to have a good frequency resolution. Concerning the localization period Δu, it is here set at half the scale, short enough to track the perceptually relevant amplitude and frequency modulations of the signal that correspond to expressive features such as vibrato or tremolo (between 4 and 10 Hz). The fundamental frequency is sampled with a step of $1/10$ tone.
- **Decomposition parameters:** The general threshold for the decomposition has been set to 15 dB or 250 atoms per second. For the atom path formation, the difference between consecutive fundamental frequencies is the corresponding sampling step of the f_0 sampling: $1/10$ ton.
- **Quantization parameters:** The weight of the first atom of a molecule is quantized on 6 bits, and the weights of the next atoms are quantized on 4 bits. The order of the DPCM quantizer is set to one. The entropy coder we use for all parameters is the adaptive arithmetic coder from Witten et al. [13].

With these parameters, we obtain computation times equivalent to 10x real-time on a 3Ghz computer and Matlab, largely dominated by the decomposition algorithm.

4.1. Full codec and reduced codec

During the analysis stage, at the end of the decomposition, the molecular algorithm tends to produces molecules that do not correspond to underlying music notes in the performance. These molecules of low energy are thus not perceptually nor physically relevant and are only extracted to reduce the overall SNR. As a consequence, the decomposition should be stopped before the apparition of such molecules. However, it is hard to find an analytical solution for a stopping criteria in the decomposition. Instead, in the framework of this study, we have prefered to manually decide the optimum number of iterations for each audio signal. A Matlab Graphical User Interface has thus been implemented (Fig. 2) where the user can listen to the synthesized signal in function of the number of iterations and thus choose the optimum number of iterations in the molecular algorithm. Such optima have been found, except for two files (Cello solo and Cello/Violin duo) where the original stopping criteria of the molecular algorithm gave the best results. We call the coder based on this manipulation the "reduced codec"; while the coder which encodes the complete set of molecules is the "full codec".



Cl_1_book.xml

n_stop 421

Decay 0033413

Figure 2: *Matlab Graphical User Interface allowing the user to visualize the representation and to select the optimal threshold for the decomposition. Different colors indicate different instruments labels.*

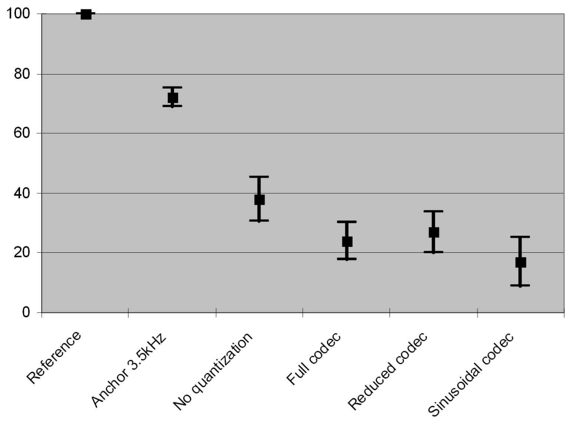


Figure 3: *MUSHRA overall mean scores*

	Full Codec (FC)	Reduced Codec (RC)
Clarinet	1.3	1.1
Cello	3.8	3.8 (*)
Flute	1.3	1.1
Oboe	2.6	1.0
Violin	4.4	2.3
Cl. / Fl.	2.4	1.9
Ce. / Fl.	1.6	1.5
Ce. / Vl.	3.9	3.9 (*)
Fl. / Fl.	4.6	2.6

Table 2: Bitrates (in kb per second) for each test file and the two variants of our codec. for the 2 files marked with an asterisk (*), the reduced codec was found equal to the full codec.

	HR	AN	NQ	FC	RC	SC
Clarinet	100	69	44	29	21	29
Cello	100	81	32	20	20	6
Flute	100	76	29	31	34	30
Oboe	100	70	40	12	20	18
Violin	100	66	62	33	33	14
Cl. / Fl.	100	74	41	36	42	36
Ce. / Fl.	100	74	25	15	21	8
Ce. / Vl.	100	70	43	13	13	6
Fl. / Fl.	100	68	30	23	21	3

Table 3: Mean of the MUSHRA scores for each version of each test signal

4.2. Listening tests

To evaluate our codecs, we performed several listening tests based on the standard MUSHRA method [14]. 15 persons took part in the listening tests to compare 5 versions of each signal: a hidden reference (HR), an anchor signal (AN) (3,5 kHz low-pass), the synthesized signal (NQ) obtained at the end of the molecular algorithm (without any quantization), the full codec (FC), the reduced codec (RC), and a simple frame-based sinusoidal coder used as a reference parametric codec (SC). The average bitrate of the codecs is around 3 kb per second for the Full Codec, and around 2 kb per second for the reduced codec (see Table 2). For the Sinusoidal Codec (SC), the bitrate was fixed at 2 kb per second. The mean of the scores obtained for each version of each signal are in Table 3. The overall means are in Fig. 3. These results first show that the reduced codec has performances that are similar or better than the full codec except for two files (Clarinet and Fl. / FL.), a case where more bits actually decrease the quality. It also shows that the reduced codec performs similarly or better than the reference sinusoidal coder except for one file (Clarinet).

5. CONCLUSION

In this paper we have described preliminary experiments that demonstrate that object-based coding of complex polyphonic music is both technically feasible and computationally tractable, when some hypothesis on the sources are verified (in our case, this is mainly an hypothesis of harmonicity). The resulting representations can achieve coding at bitrates as low as 2 kbs for monophonic sounds, with a sound quality that is in general comparable to sinusoidal coding, and in some cases significantly better. Further improvements will be focused on three directions : first, we can improve on our decomposition model, for instance on estimating jointly the combinations of notes that optimally explains the signal for polyphonic music. Then, we can improve the quantization and coding techniques, in order to reduce the loss of sound quality due to quantization. Finally, we want to investigate criteria for stopping strategies in the Matching Pursuit decomposition process corresponding to the optimal codec (here called the reduced codec).

Another question, which is still open at the moment, is whether these techniques would still perform well with an increase of the number of instruments. It would as well be interesting to evaluate its performance on sounds that are not included in the training set but still verify the harmonicity assumption (for instance, voice), or on other classes of sounds such as percussive sounds or noisy sounds.

6. ACKNOWLEDGEMENTS

The authors wish to thank Emmanuel Vincent for providing the reference sinusoidal coder, and the members of the LAM for taking part in the listening tests.

7. REFERENCES

[1] International Organization for Standardization, "ISO/IEC 14496-3:2001, information technology - coding of audio-visual objects - part 3: Audio," 2001.

[2] A.C. den Brinker, E.G.P. Schuijers, and A.W.J. Oomen, "Parametric coding for high-quality audio," in *Proceedings of the 112th AES Convention*, Munich, Germany, May 2002.

[3] H. Purnhagen and N. Meine, "HILN-the MPEG-4 parametric audio coding tools," in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS 2000)*, May 2000, vol. 3, pp. 201–204.

[4] E Vincent and MD Plumbley, "Low bitrate object coding of musical audio using bayesian harmonic models," *To appear in IEEE Trans. on Audio, Speech and Language Processing*.

[5] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, December 1993.

[6] R. Gribonval and E. Bacry, "Harmonic decomposition of audio signals with matching pursuit," *IEEE Trans. on Signal Processing*, vol. 51, no. 1, pp. 101–111, January 2003.

[7] L. Daudet, "Sparse and structured decompositions of signals with the molecular matching pursuit," *IEEE Trans. on Speech, Audio and Language Processing*, vol. 14, no. 5, pp. 1808–1816, September 2006.

[8] P. Leveau, E. Vincent, G. Richard, and L. Daudet, "Instrument-specific harmonic atoms for mid-level music representation," *submitted to IEEE Trans. on Speech, Audio and Language Processing*.

[9] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC Musical Instrument Sound Database," Distributed online at http://staff.aist.go.jp/m.goto/RWC-MDB/.

[10] "University of iowa music instrument samples database, http://theremin.music.uiowa.edu/mis.html," .

[11] "Ircam studio online database, http://forumnet.ircam.fr/402.html?l=1," .

[12] Allen Gersho and Robert M. Gray, *Vector Quantization and Signal Compression*, Springer, 1991.

[13] Ian H. Witten, Radford M. Neal, and John G. Cleary, "Arithmetic coding for data compression," *Commun. ACM*, vol. 30, no. 6, pp. 520–540, 1987.

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