ENTROPY-CONSTRAINED SPIKE MODULUS QUANTIZATION IN A BIO-INSPIRED UNIVERSAL AUDIO CODER

Ramin Pichevar, Hossein Najaf-Zadeh, Hassan Lahdili, and Louis Thibault

Advanced Audio Systems
Communications Research Centre, Ottawa, Canada
Ramin.Pishehvar@crc.ca, Hossein.NajafZadeh@crc.ca, Hassan.Lahdili@crc.ca, Louis.Thibault@crc.ca

ABSTRACT

We propose an optimal quantizer for the moduli (spike amplitudes) of the matching pursuit on gammatone/gammachirp kernels that have been shown previously to be very efficient for audio coding. The quantizer optimization is performed by Genetic Algorithm (GA). We show that the optimal quantization values found by GA can be approximated by a ”piecewise uniform” quantizer. The latter approximation is faster and has less overhead (i.e., there is no need for codebook transmission). Based on perceptual evaluation of audio quality, transparent quality is obtained for both the optimal and piecewise uniform quantization approaches. We also studied the performance of in-loop quantization vs. out-of-loop quantization when computational cost can be afforded.

1. INTRODUCTION

Non-stationary and time-relative structures such as transients, timing relations among acoustic events, and harmonic periodicities provide important cues for different types of audio processing (i.e., audio coding). Obtaining these cues is a difficult task. The most important reason why it is so difficult is that most approaches to signal representation are block-based, i.e. the signal is processed piecewise in a series of discrete blocks. Transients and non-stationary periodicities in the signal can be temporally smeared across blocks. Large changes in the representation of an acoustic event can occur depending on the arbitrary alignment of the processing blocks with events in the signal. Signal analysis techniques such as windowing or the choice of the transform can reduce these effects, but it would be preferable if the representation was insensitive to signal shifts. Shift-invariance alone, however, is not a sufficient constraint on designing a general sound processing algorithm. A desirable code should reduce the information rate from the raw signal so that the underlying structures are more directly observable (see [16] among others).

We outlined a framework in [4] that meets the above-mentioned requirements. We proposed a bio-inspired universal audio coder based on projecting signals onto a set of overcomplete atoms consisting of gammatone/gammachirp kernels. The projections on the kernels are called spikes, since they can be considered as the spikes generated by hair cells in the auditory pathway (see Fig. 1). The best atom at each iteration is found by matching pursuit. Our proposed method in [4] was an adaptive version of [16] and uses gammachirp kernels instead of the original gammatones used in [16]. In our approach, at each matching pursuit iteration, six different parameters (i.e., amplitude, delay, frequency, chirp factor, attack, and decay) are extracted in the adaptive case, while three parameters (i.e., amplitude, delay, frequency) are extracted in the non-adaptive case. Note that our approach is different from other works in the literature (i.e., [6]) in which gammatones are used as a filterbank and not as kernels for the generation of sparse representations based on matching pursuit. We also showed in [4] that, when used for audio coding, our approach outperforms the work in [16] in terms of bitrate and number of atoms for the same perceptual quality on different types of signals. The better performance is due to the fact that we proposed in [4] an adaptive scheme that fine tunes the kernels chosen by matching pursuit (section 1.1).

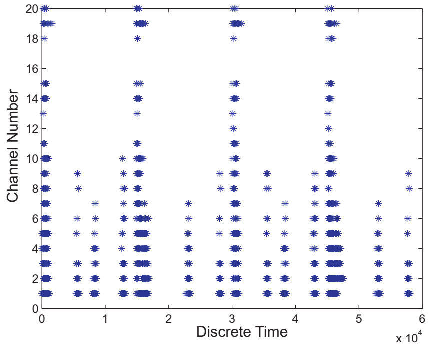

Figure 1: Spikegram of the percussion signal using the gammatone matching pursuit algorithm (spike amplitudes are not represented). Each dot represents the time and the channel where the spike fired (extracted by MP).

The solution proposed in [4] fits in the paradigm of sparse overcomplete representations, which has close ties with the compressive sensing framework [2]. Sparse representations are good at extracting non-stationary and time-relative structures such as transients, timing relations among acoustic events, and harmonic periodicities. They are also time shift-invariant [16].

In [4], we only studied the analysis/synthesis of a given signal using our proposed method when no quantization was used. In this article, we propose how to quantize the spike amplitudes and how to reach a trade-off between the quality of reconstruction and bitrate. To do so, we first defined a cost function consisting of a trade-off between distortion and entropy. We then optimize the cost function and finally find an approximation to the optimal solution. This approach fits in the paradigm of Rate-Distortion (R-D) based optimization techniques [13] [1]. It can also be compared to [3] [17] [15], in which a theoretical approach is used for the quantization of amplitude, phase, and frequency in the sinusoidal coding paradigm. However, our method does not rely on the high resolution assumption and can be applied to any given distortion measure. In addition, the extension of the theoretical derivation in [3] and [17] to the case of gammatone and gammachirp kernels with 6 coding parameters (amplitude, delay, frequency, chirp factor, attack, and decay) is not obvious.

1.1 The Bio-Inspired Audio Coder: Analysis/Synthesis

The analysis/synthesis part of our universal audio codec presented in [4] is based on the generation of sparse 2-D representations of audio signals, dubbed as spikegrams. The spikegrams are generated by projecting the signal onto a set of overcomplete adaptive gammachirp (gammatones with additional tuning parameters) kernels. The adaptiveness is a key feature we introduced in MP to increase the efficiency of the proposed method (see [4]). A masking model is applied to the spikegrams to remove inaudible spikes [11]. In addition a differential encoder of spike parameters based on graph theory is proposed in [5]. The block diagram of all the building blocks of the receiver and transmitter of our proposed universal audio coder is depicted in Fig. 2, of which the quantization block is discussed in this paper. Our paradigm uses matching pursuit (MP) with an adaptive fine tuning to extract signal features as described in the next section.

1.1.1 Generating Overcomplete Representations with MP

In mathematical notations, the signal $x(t)$ can be decomposed into the overcomplete kernels as follows:

$$x(t) = \sum_{m=1}^{M} \sum_{i=1}^{n_m} a_i^m g_m(t - \tau_i^m) + r_x(t), \qquad (1)$$

where τ_i^m and a_i^m are the temporal position and amplitude of the ith instance of the kernel g_m, respectively. The notation n_m indicates the number of instances of g_m, which need not be the same across kernels and M is the number of kernels. In addition, the kernels are not restricted in form or length.

In order to find adequate τ_i^m, a_i^m, and g_m, MP can be used. In this technique the signal $x(t)$ is decomposed over a set of kernels so as to capture the structure of the signal. The approach consists of iteratively approximating the input signal with successive orthogonal projections onto some bases. The signal can be decomposed into

$$x(t) = \sum_m < x(t), g_m > g_m + r_x(t), \qquad (2)$$

where $< x(t), g_m >$ is the inner product between the signal and the kernel and is equivalent to a^m in Eq. 1. Throughout this article the terms "spike amplitude" and "modulus" are used to designate a^m, and $r_x(t)$ is the residual signal.

1.1.2 Gammachirp-Based Representations

In [4] we used adaptive gammachirp kernels as described below. The impulse response of a gammachirp filter with the corresponding tuning parameters (b,l,c) is given below

$$g(f_c, t, b, l, c) = t^{l-1} e^{-2\pi bt} \cos(2\pi f_c t + c\ln(t)) \quad t > 0. \qquad (3)$$

Gammachirp filters minimize the scale/time uncertainty [8]. In this approach the chirp factor c, l, and b are found adaptively at each step. The chirp factor c allows us to slightly modify the instantaneous frequency of the kernels. The parameters l and b controls the attack and the decay of kernels. However, searching the three parameters in the parameter space is a very computationally complex task. Therefore, we use a suboptimal search that uses the same gammatone filters as the ones used in the non-adaptive paradigm with values of l and b given in [8]. The first step gives us the center frequency and start time (t_0) of the best gammatone matching filter. We also keep the second best frequency (gammatone kernel) and start time.

$$G_{max1} = \underset{f,t_0}{\operatorname{argmax}} \{|< r, g(f, t_0, b, l, c) >|\} \qquad (4)$$

$$G_{max2} = \underset{f,t_0}{\operatorname{argmax}} \{|< r, g(f, t_0, b, l, c) >|\} \qquad (5)$$

where the search space in Eq. 4 is G, the set of all kernels. The search space in Eq. 5 is $G - G_{max1}$, the set of kernels that excludes G_{max1}. For the sake of simplicity, we use f instead of f_c in Eqs. (4) to (8). We then use the information found in the first step to find c. In other words, we keep only the set of the best two kernels in step one, and try to find the best chirp factor given $g \in G_{max1} \cup G_{max2}$.

$$G_{maxc} = \underset{c}{\operatorname{argmax}} \{|< r, g(f, t_0, b, l, c) >|\}. \qquad (6)$$

We then use the information found in the second step to find the best b for $g \in G_{maxc}$ in Eq. 7, and finally find the best l among $g \in G_{maxb}$ in Eq. 8.

$$G_{maxb} = \underset{b}{\operatorname{argmax}} \{|< r, g(f, t_0, b, l, c) >|\} \qquad (7)$$

$$G_{maxl} = \underset{l}{\operatorname{argmax}} \{|< r, g(f, t_0, b, l, c) >|\}. \qquad (8)$$

Therefore, six parameters are extracted in the adaptive technique : center frequencies, chirp factors c, time delays, spike amplitudes, b, and l.

In the following section, a new solution to the optimal quantization of the spike amplitudes (moduli) and a fast approximation to the optimal solution are proposed.

2. SPIKE AMPLITUDE QUANTIZATION

2.1 Cost Function and Optimization

The cost function we use is the result of a trade-off between the quality of reconstruction and the number of bits required to code each modulus. More precisely, given the vector of quantization levels (codebook) $\mathbf{q}$, the cost function to optimize is given by (R is the bitrate and D is the distortion):

$$\widehat{E}(\mathbf{q}) = D + \lambda R = \frac{\|\sum_i \hat{\alpha}_i g_i - \sum_i \alpha_i g_i\|^2}{\|\sum_i \alpha_i g_i + \eta\|^\gamma} + \lambda H(\hat{\alpha}), \qquad (9)$$

where $\eta = 10^{-5}$, $\gamma = 0.001$ are set empirically using informal listening tests. The entropy, $\widehat{E}(\mathbf{q})$, is computed using the absolute value of spike amplitudes. $\hat{\alpha}$ is the vector of quantized amplitudes and is computed as follows:

$$\hat{\alpha}_i = q_i \quad \text{if} \quad q_{i-1} < \alpha_i < q_i \qquad (10)$$

$H(\hat{\alpha})$ is the per spike entropy in bits needed to encode the information content of each element of $\hat{\alpha}$ defined as:

$$H(\hat{\alpha}) = -\sum_i p_i(\hat{\alpha}_i) \log_2 p_i(\hat{\alpha}_i), \qquad (11)$$

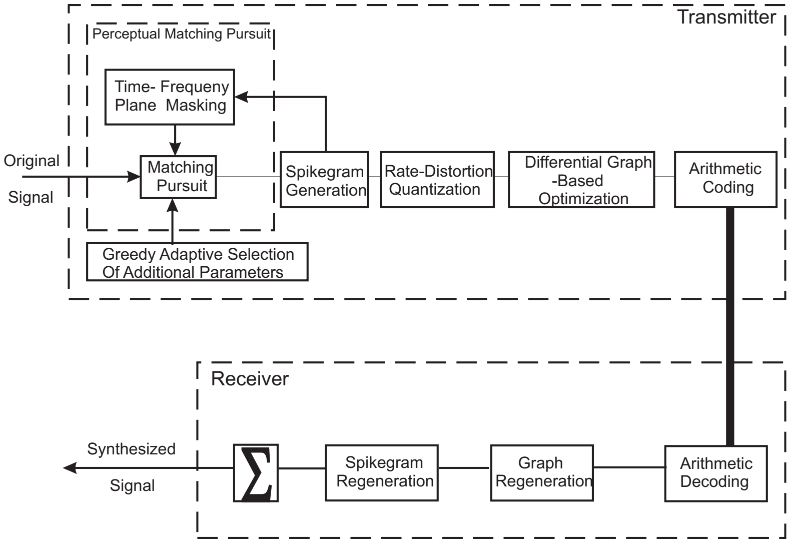
Transmitter
Perceptual Matching Pursuit
Time- Frequeny Plane Masking
Original Signal
Matching Pursuit
Spikegram Generation
Rate-Distortion Quantization
Differential Graph -Based Optimization
Arithmetic Coding
Greedy Adaptive Selection Of Additional Parameters
Receiver
Synthesized Signal
∑
Spikegram Regeneration
Graph Regeneration
Arithmetic Decoding

Figure 2: Block diagram of the Universal Bio-Inspired Audio Coder.

where $p_i(\hat{\alpha}_i)$ is the probability density function of $\hat{\alpha}_i$. The way the quantizer is defined in Eq. 10 reduces the dead zone problem (defined in [12]). To proceed with the optimization at a given number of quantization levels, we randomly set the initial values (initial population) for the q_i and perform Genetic Algorithm to find optimal solutions. The goal of the weighting in the denominator of D (Eq. 9) is to give a better reconstruction of low-energy parts of the signal.

Note that in Eq. 9 many different $\hat{\alpha}_i$ can contribute to the reconstruction of the original signal at a given instance of time t, which is not the case when quantization is applied on time samples (Lloyd algorithm). Therefore, the optimal $\hat{\alpha}_k$ are not statistically independent. In addition, in contrast with transform-based coder quantizations (done for instance with Lloyd algorithm), g_k are a few atoms selected from a large set of different atoms (tens of thousand) in the dictionary and there is an entropy maximization term in our cost function. It is therefore impossible to derive a closed-form theoretical solution for the optimal $\hat{\alpha}_i$ in the case of sparse representations. Hence, we should use adaptive optimization techniques. In order to avoid local minima, we use genetic algorithm to optimize the cost function given in the following section. Another advantage of the genetic algorithm is its flexibility in using different cost functions and distortion measures (mean square error, CRC-SEAQ, etc.).

2.2 Genetic Algorithm

A genetic algorithm (or GA) is a search technique used in computing to find true or approximate solutions to optimization and search problems [10]. GA is categorized as a global search heuristic. GA is a particular class of evolutionary algorithms that use techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover.

GA is implemented as a computer simulation in which a population of chromosomes of candidate solutions (called individuals) to an optimization problem evolves toward better solutions [1]. The evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from the current population (based on their fitness), and modified to form a new population at each iteration. Throughout this paper, the number of individuals per subpopulation is 700, maximum number of generations is equal to 50, the generation gap is 0.9, the number of variables is 2^5, and the binary precision is 15 bits for GA.

In Fig. 3, we plotted the mimum point of the cost function (as defined in Eq. 9) obtained by GA versus different number of quantization levels for both the entropy-constrained and non-constrained cases for speech. As the reader may see, the entropy-constrained approach gives always better results than the non-constrained paradigm. In addition according to these curves the optimal number of quantization levels is somewhere between 32 and 64.

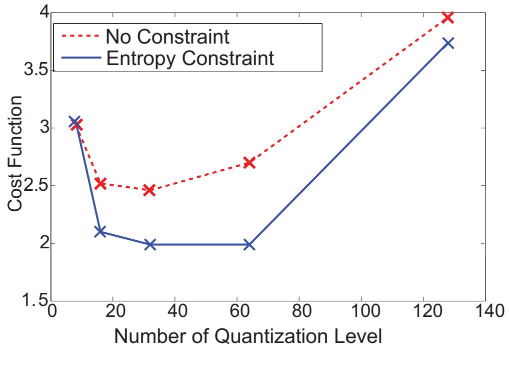

Figure 3: The cost function for the constrained and non-constrained quantizers at different numbers of quantization levels for speech.

2.3 Quantization Results

Four different signal types are used: percussion, harpsichord, castanet, and speech. For each signal type the R-D optimiza-

[1] Used toolbox: http://www.shef.ac.uk/acse/research/ecrg/getgat.html

	32 Levels		64 Levels	
	SEAQ	Bits/spike	SEAQ	Bits/spike
Percussion	-0.04	1.48	-0.10	2.74
Castanet	-0.70	2.27	-0.33	2.84
Harpsichord	-0.90	1.56	-0.09	2.34
Speech	-0.32	2.15	-0.14	2.73

Table 1: Perceptual quality evaluation of the optimal quantization for different signal types and different numbers of quantization levels with CRC-SEAQ. A score between 0 and -1 corresponds to transparent quality. The bitrate includes the sign bit.

tion based on GA as described above is run and the optimal codebook in each case is found. The analysis/synthesis gammachirp MP is applied and spikes are extracted. Each spike amplitude is then quantized according to the optimal codebook. A perceptual quality evaluation (CRC-SEAQ) [2] is used to assess the quality of the reconstructed signal after quantization compared to the reconstructed signal without quantization. Table 1 shows that based on the proposed approach we obtained transparent quality for all signal types and for both numbers of quantization levels. These results are similar to our informal listening test results. The bitrate is obtained by applying arithmetic coding to spike amplitudes. The arithmetic coding is applied to longer blocks (1 second) than what is used for the computation of entropy in the cost function in order to decrease bitrate. Note that the GA is applied on the absolute value of the spikes and the sign bit is sent separately (this strategy gives better results than when signed codebooks are used).

3. PIECEWISE UNIFORM QUANTIZATION

Running GA for each signal is a time consuming task. In addition, sending a new codebook for each signal type and/or frame is an overhead we may want to avoid. In this section we propose faster ways to find a suboptimal solution to the quantization results that keeps transparency in quality. The goal is achieved by performing a piecewise uniform approximation of the codebook by using the histogram of the moduli.

Fig. 4 shows the optimal quantization levels (q_i) for four different types of signals. The optimal signal is obtained using the GA algorithm explained in the previous section.

As we can see, the optimal levels can be approximated as piecewise linear segments (meaning that the quantizer is ”piecewise” linear). The optimal levels are updated by the following method for each one-second-long frame:

- Find the 40-bin histogram h of the spike amplitudes.
- Threshold the histogram by the sign function so that $h_t = sign(h)$ to find spike amplitude clusters (concentrations). Smooth out the curves by applying a moving average filter with the following impulse response: $m(n) = \sum_k 0.125\delta(n-k)$ for $k = 1, 2, ...8$.
- Set a crossing threshold of 0.4 on the smoothed curve. Each time the curve crosses the threshold, define a new uniform quantizer between the two last threshold crossings.

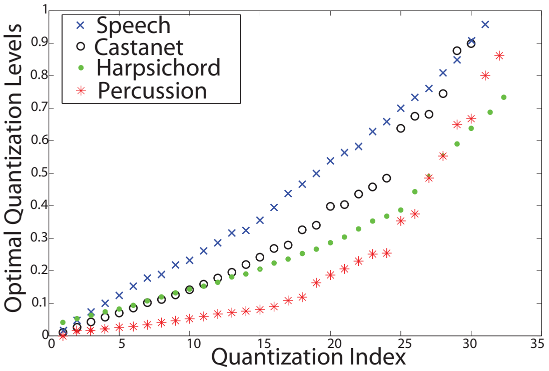

Figure 4: Optimal quantization levels for different sound categories. Spike amplitudes are normalized to one.

	CRC-SEAQ	
	32-Levels	64 Levels
Percussion	-1.30	-0.25
Castanet	-0.50	-0.10
Harpsichord	-1.10	-0.15
Speech	-0.95	-0.44

Table 2: Comparison of 32-level and 64-level piecewise uniform quantizers for different audio signals. A CRC-SEAQ score between -1 and -2 is associated with near transparent quality. No codebook side information is necessary to send to the receiver in this case.

3.1 Results from Piecewise Uniform Quantization

For the different signal types we used in section 2.3, we proceeded with the fast piecewise uniform quantization described in the previous subsection. We noticed that the 32-level quantizer gives only near-transparent coding results with CRC-SEQA (see Table 2) for the piecewise uniform quantizer. However, the quality is transparent when 64 levels are used. These observations have been confirmed with informal listening tests. This behavior is due to the fact that the 64-level quantizer has more uniform quantization levels than the 32-level quantizer. Therefore, we recommend the 64-level quantizer when the piecewise uniform approximation is used.

The overall codec bitrate can be computed by combining the bitrate in Tables 1-3 of [4] for the unquantized case and values for amplitude quantization in Table 2 of this article.

4. IN-LOOP VS. OUT-OF-LOOP QUANTIZATION

Here, we investigate the difference between in-loop [12] and out-of-loop quantizations. In the out-of-loop quantization, MP is run on unquantized spike amplitudes and stored in a vector. These unquantized values are then quantized according to the optimal codebook found by the GA or the piecewise uniform quantizer described in the previous section (results for the out-of-loop quantization are given in Tables 1 and 2). We here propose a different strategy: the in-loop quantization, which needs 2 passes of MP as described in [12]. During the first pass MP is applied to the original signal and the optimal quantization values are found (by GA or piecewise uniform approximation). MP is then run a second time and the amplitudes are quantized at each iteration

[2] http://www.itu.int/rec/R-REC-BS.1387-1-200111-I/e

before computing the residual ($\hat{\alpha}_i$ are quantized moduli) by using the codebook found in pass 1:

$$x_i = \hat{\alpha}_i g_i + r_i. \qquad (12)$$

We performed the in-loop quantization for different signal types. Fig. 5 compares the performance of the in-loop and out-of-loop quantizers for castanet using our spikegram paradigm. Note that the in-loop quantization has far better performance at a greater computational cost. In addition, another drawback of the in-loop quantization is that the full MP decomposition must be computed for every bitrate.

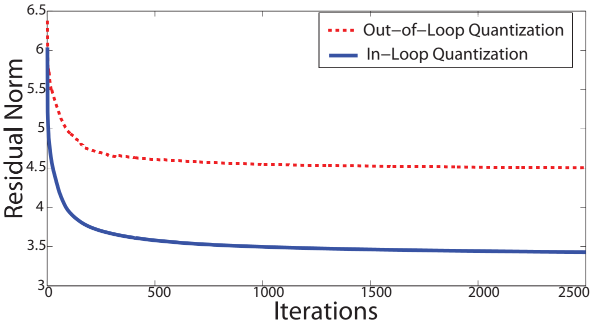

Figure 5: The residual norm ($\|r_i\|$) in Eqs. 2 and 12 of in- and out-of-loop quantizations at each MP iteration for castanet.

5. FUTURE WORK

In the current work, the mean square error has been used in the cost function. In a future work, the cost function of the GA can be replaced by the output of CRC-SEAQ. In addition, the cost function of the GA can also be written as a constrained optimization problem: minimizing the distortion error subject to the inequality constraint on bitrate. The optimal λ (Eq. 9) in this latter case can be found by bisection or Newton's method [14]. By doing so, one can plot the operational rate-distortion curves [13] for our proposed paradigm.

The application of dependent MP that uses the notion of consistency to project values found by MP onto consistent values [18] should be considered in a later work. The size of the dictionary can also have a great impact on the effects of the quantization errors on reconstruction [7] [9]. This issue should be further investigated.

Formal listening test must be conducted and results must be compared with objective tests (i.e., CRC-SEAQ) and with standard codecs.

6. CONCLUSION

We propose an optimal quantization method for spike amplitudes in our bio-inspired universal audio coder based on genetic algorithm. We also propose a suboptimal fast quantization based on a piecewise uniform approximation. In this suboptimal scheme, only the start and end points of each segment are sent to the receiver. Therefore, no codebook information is needed to be sent to the receiver.

We obtained transparent quantization results for both the optimal and suboptimal quantization methods for different signal types. We also compared the in-loop and out-of-loop quantization paradigms and showed that in-loop quantization outperforms out-of-loop quantization at a greater computational cost.

REFERENCES

[1] T. Berger. *Rate-distortion theory: A mathematical basis for data compression*. Prentice-Hall, 1971.

[2] D. Donoho. Compressive sensing. *IEEE Trans. on Information Theory*, 52(4):1289–1306, 2006.

[3] P. Korten et al. High-resolution spherical quantization of sinusoidal parameters. *IEEE Trans. on audio, speech, and language proc.*, 15(3):966–981, 2007.

[4] R. Pichevar et al. A biologically-inspired low-bit-rate universal audio coder. In *Proceedings of Audio Engineering Society Convention*, 2007.

[5] R. Pichevar et al. Differential graph-based coding of spikes in a biologically-inspired universal audio coder. In *Audio Engineering Society Convetion, Amsterdam, The Netherlands*, 2008.

[6] C. Feldbauer, G. Kubin, and B. Kleijn. Anthropomorphic coding of speech and audio: A model inversion approach. *EURASIP-JASP*, (9):1334–1349, 2005.

[7] V. Goyal and M. Vetterli. Consistency in quantized matching pursuit. In *International Conference on Acoustics, Speech, and Signal Processing*, 1996.

[8] T. Irino and R.D. Patterson. A dynamic compressive gammachirp auditory filterbank. *IEEE Trans. on Audio and Speech Processing*, 14(6):2008–2022, 2006.

[9] Q. Liu, Q. Wang, and L. Wu. Size of the dictionary in matching pursuit algorithm. *IEEE Trans. on Signal Processing*, 52(5):3403–3408, 2004.

[10] M. Mitchell. *An Introduction to Genetic Algorithms (Complex Adaptive Systems)*. MIT Press, 1998.

[11] H. Najaf-Zadeh, R. Pichevar, L. Thibault, and H. Lahdili. Perceptual matching pursuit for audio coding. In *Audio Engineering Society Convetion, Amsterdam, The Netherlands*, 2008.

[12] R. Neff and A. Zakhor. Modulus quantization for matching pursuit video coding. *IEEE Trans. on Circuits and Systems for Video Technology*, 10(6):895–912, 2000.

[13] A. Ortega and K. Ramchandran. Rate-distortion methods for image and video compression. *IEEE Signal Processing Magazine*, 15(16):23–50, 1998.

[14] K. Ramchandran and M. Vetterli. Best wavelet packet bases in a rate-distortion sense. *IEEE Trans. on Image Processing*, 2(2):160–175, 1993.

[15] E. Ravelli and L. Daudet. Embedded polar quantization. *IEEE Signal Processing Letters*, 14(10):657–660, 2007.

[16] E. Smith and M. Lewicki. Efficient auditory coding. *Nature*, 7079:978–982, 2006.

[17] R. Vafin and B. Kleijn. Entropy-constrained polar quantization and its application to audio coding. *IEEE Trans. speech and audio proc.*, pages 220–232, 2005.

[18] V. Goyal M. Vetterli and N.T. Thao. Quantized overcomplete expansion in R^N: Analysis, synthesis, and algorithms. *IEEE Trans. on Information Theory*, 44(1):16–31, 1998.