3

New Trends in Biologically-Inspired Audio Coding

Ramin Pichevar, Hossein Najaf-Zadeh, Louis Thibault and Hassan Lahdili
Advanced Audio Systems, Communications Research Centre
Ottawa, Canada

1. Abstract

This book chapter deals with the generation of auditory-inspired spectro-temporal features aimed at audio coding. To do so, we first generate sparse audio representations we call spikegrams, using projections on gammatone or gammachirp kernels that generate neural spikes. Unlike Fourier-based representations, these representations are powerful at identifying auditory events, such as onsets, offsets, transients and harmonic structures. We show that the introduction of adaptiveness in the selection of gammachirp kernels enhances the compression rate compared to the case where the kernels are non-adaptive. We also integrate a masking model that helps reduce bitrate without loss of perceptible audio quality. We then quantize coding values using the genetic algorithm that is more optimal than uniform quantization for this framework. We finally propose a method to extract frequent auditory objects (patterns) in the aforementioned sparse representations. The extracted frequency-domain patterns (auditory objects) help us address spikes (auditory events) collectively rather than individually. When audio compression is needed, the different patterns are stored in a small codebook that can be used to efficiently encode audio materials in a lossless way. The approach is applied to different audio signals and results are discussed and compared. This work is a first step towards the design of a high-quality auditory-inspired “object-based" audio coder.

2. 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 techniques including audio coding, speech recognition, audio localization, and auditory scene analysis. Obtaining these cues is a difficult task. The most important reason why it is so difficult is that most approaches to signal representation/analysis are block-based, i.e. the signal is processed piecewise in a series of discrete blocks. Therefore, transients and non-stationary periodicities in the signal can be temporally smeared across blocks. Moreover, 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 representation should capture the underlying 2D-time-frequency structures, so that they are more directly observable and well represented at low bit rates (Smith & Lewicki, 2005). These structures must be easily extractable as auditory objects for further processing in coding, recognition, etc.

The aim of this chapter is to first introduce sparse biologically-inspired coding and then propose an auditory-inspired coding scheme, which includes many characteristics of the auditory pathway such as sparse coding, masking, auditory object extraction, and recognition (see Fig. 6). In the next section we will see how sparse codes are generated and why they are efficient.

3. Sparse Coding

Research on sparse coding is generally conducted almost independently by two group of researchers: signal processing engineers and biophysicists. In this chapter, we will try to make a link between these two realms. In a mathematical sense, sparse coding generally refers to a representation where a small number of components are active. In the biological realm, a sparse code generally refers to a representation where a small number of neurons are active with the majority of neurons being inactive or showing low activity (Graham & Field, 2006). Over the last decade, mathematical explorations into the statistics of natural auditory and visual scenes have led to the observation that these scenes, as complex and varied as they appear, have an underlying structure that is sparse. Therefore, one can learn a possibly overcomplete basis[1] set such that only a small fraction of the basis functions is necessary to describe a given audio or video signal. In section 5.1, we will see how these codes can be generated by projecting a given signal onto a set of overcomplete kernels. When the cell's amplitude is different from zero, we say that the neuron or cell is active and has emitted a spike. To show the analogy between sparse 2-D representations and the underlying neural activity in the auditory or visual pathway, we call the 2-D sparse representation spikegram (in contrast with spectrograms) and the components of a sparse representation cells or neurons throughout this chapter.
In a sparse code, the dimensionality of the analyzed signal is maintained (or even increased). However, the number of cells responding to any particular instance of the input signal is minimized. Over the population of likely inputs, every cell has the same probability of producing a response but the probability is low for any given cell (Field, 1994). In other words, we have a high probability of no response and a high probability of high response, but a reduction in the probability of a mid-level response for a given cell. We can thus increase the peakiness (kurtosis) of the histogram of cell activity and be able to reduce the total number of bits (entropy) required to code a given signal in sparse codes by using any known arithmetic coding approach. The sparse coding paradigm is in contrast with approaches based on Principal Component Analysis (PCA) (or Karhunen-Loeve transform), where the aim is to reduce the number of signifcant signal components. Fig. 1 shows the conceptual differences between the two approaches as described above.
Normally, sparseness occurs in space (population sparseness) or in time (lifetime sparseness). Population sparseness means that our 2-D sparse representation (spikegram) has very few active cells at each instance of time, while lifetime sparseness means that each cell in the representation is acitve only for a small fraction of the time span of the audio/video signal.

3.1 *Sparse Coding and ICA*

Sparse coding as described in this chapter can also be related to Independent Component Analysis (ICA) (Hyvarinen et al., 2009). In fact for some signals (i.e., an ensemble of natural images), the maximization of sparseness for a linear sparse code is basically the same as

[1] A set of bases in which the number of kernels/atoms is higher than the dimension of the audio/video signal

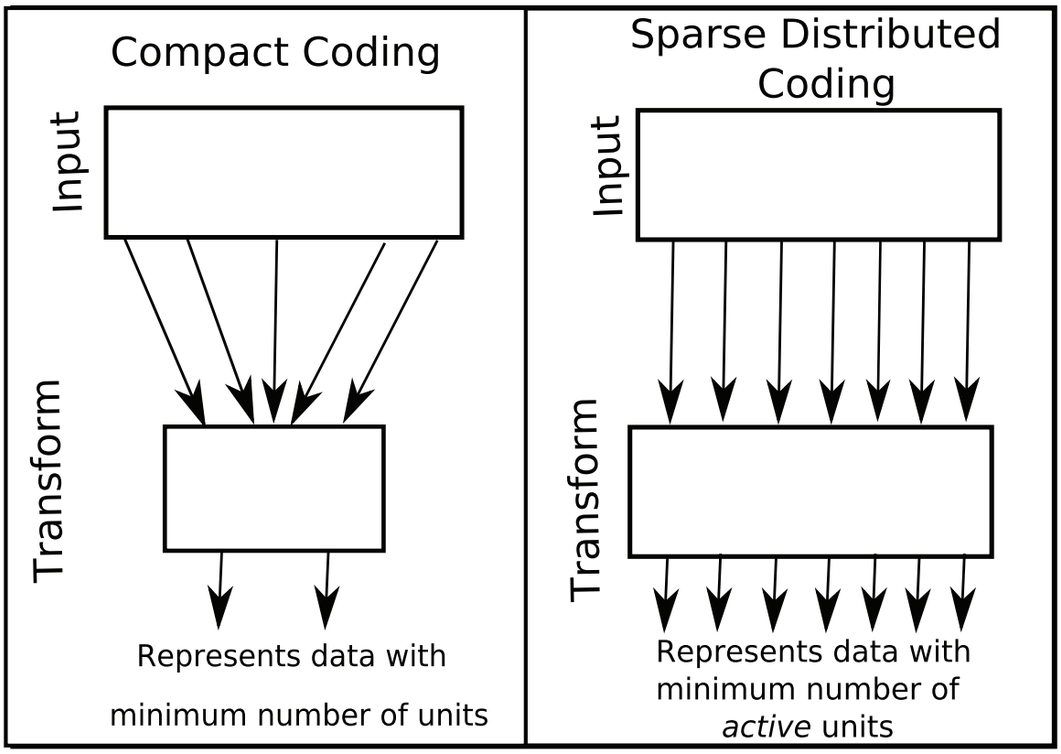


Fig. 1. Conceptual differences between sparse representations and PCA/Karhunen-Loeve transform (reproduced from (Field, 1994)). Note that in some cases the dimensionality of the sparse code is even higher than the input signal.

the maximization of non-gaussianity in the context of overcomplete ICA (Hyvarinen et al., 2009). Karklin and Lewicki also discussed the limits of applicability of the aforementioned equivalence in (Karklin & Lewicki, 2005) (Karklin & Lewicki, 2009). However, in the general case where components (cell activities) are not statistically independent (i.e., small patches of natural images) and noise is present in the system, maximizing sparseness is not equivalent to maximizing non-gaussianity and as a consequence ICA is not equivalent to sparse coding anymore.

4. Advantages of Sparse Coding

In this section we give some reasons (among others) on why sparse coding is such a powerful tool in the processing of audio and video materials.

Signal-to-Noise Ratio

A sparse coding scheme can increase the signal-to-noise ratio (Field, 1994). In a sparse code, a small subset of cells represents all the variance present in the signal (remember that most of the cells are inactive in a sparse code). Therefore, that small active subset must have a high response relative to the cells that are inactive (or have outputs equal to zero). Hencce, the probability of detecting the correct signal in the presence of noise is increased in the sparse coding paradigm compared to the case of a transform (e.g., Fourier Transform) where the

variance of the signal is spread more uniformly over all coefficients. It can also be shown that sparse/overcomplete coding is optimal when a transmission channel is affected by quantization noise and is of limited capacity (see (Doi et al., 2007) and (Doi & Lewicki, 2005)).

Correspondence and Feature Detection

In an ideal sparse code, the activity of any particular basis function has a low probability. Since the response of each cell is relatively rare, tasks that require matching of features should be more successful, since the search space is only limited to those active cells (Field, 1994).
It has also be shown that the inclusion of a non-negativeness constraint into the extraction of sparse codes can generate representations that are part-based (Pichevar & Rouat, 2008) (Lee & Seung, 1999) (Hoyer, 2004). It is presumably easier to find simple parts (primitives) in an object than identifying complex shapes. In addition, complex shapes can be charachterized by the relationship between parts. Therefore, it seems that non-negative sparse coding can be potentially considered as a powerful tool in pattern recognition.

Storage and Retrieval with Associative Memory

It has been shown in the literature that when the inputs to an associative memory[2] network are sparse, the network can store more patterns and provide more effective retrieval with partial information (Field, 1994) (Furber et al., 2007).
As a simple argument of why sparse codes are efficient for storage and retrieval, Graham and Field (Graham & Field, 2006) gave the follwoing example. Consider a collection of 5x5 pixel images that each contain one block letter of the alphabet. If we looked at the histogram of any given pixel, we might discover that the pixel was on roughly half the time. However, if we were to represent these letters with templates that respond uniquely to each letter, each template would respond just 1/26th of the time. This letter code is more sparse-and more efficient-relative to a pixel code. Although no information is lost, the letter code would produce the lowest information rate. Moreover, a representation that was letter-based (and sparse) would provide a more efficient means of learning about the association between letters. If the associations were between individiual pixels, a relatively complex set of statistical relationships would be required to describe the co-occurences of letters (e.g., between the Q and U). Sparseness can assit in learning since each unit is providing a relatively complete representation of the local structure.

Shift Invariance

In transform-based (block-based) coding (e.g., Fourier Transforms), representations are sensitive to the arbitrary alignment of the blocks (analysis window) (see Fig. 2). Even wavelets are shift variant with respect to dilations of the input signal, and in two dimensons, rotations of the input signal (Simoncelli et al., 1992). However, with sparse coding techniques as defined in this manuscript this sensitivity problem is completely solved, since the kernels are positioned arbitrarily and independently (Smith & Lewicki, 2005).

4.1 Physiological evidence for sparse coding

Much of the discussion in recent years regarding sparse coding has come from the the theoretical and computational communities but there is substantial physiological evidence for sparse

[2] An associative memory is a dynamical system that saves memory attributes in its state space via attactors. The idea of associative memory is that when a memory clue is presented, the actual memory that is most like the clue will be recapitulated (see (Haykin, 2008) for details).

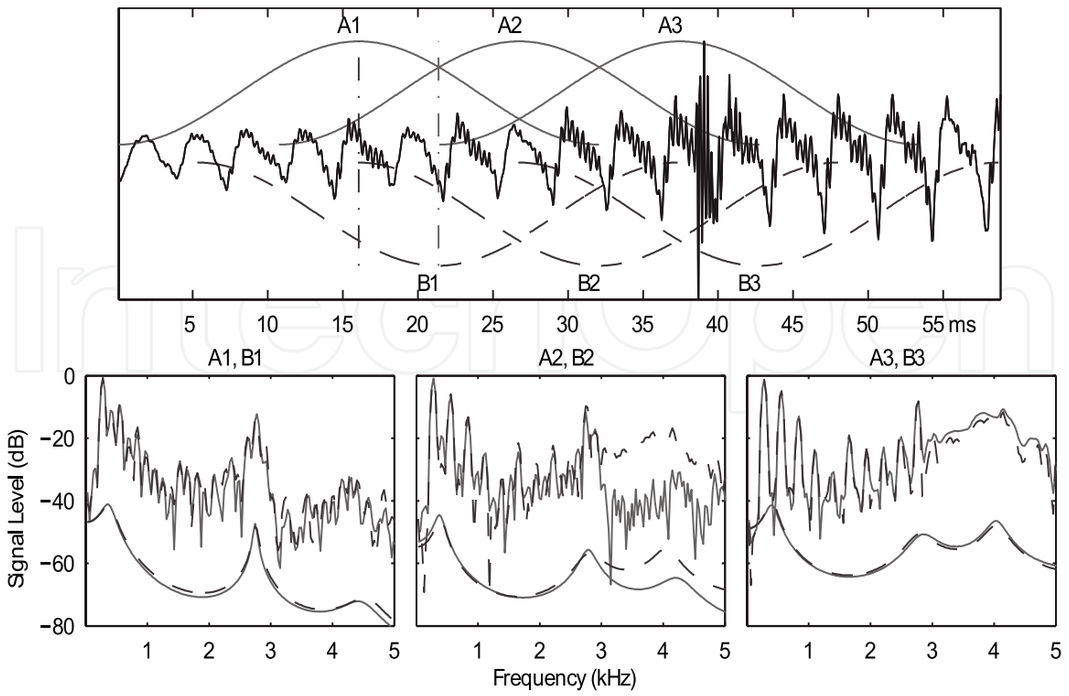


Fig. 2. Block-based representations are sensitive to temporal shifts. The top panel shows a speech waveform with two sets of overlaid Hamming windows, A1-3 (continuous lines above waveform) and B1-3(dashed lines below waveform). In the three lower panels, the power spectrum (jagged) and Linear Prediction Coding (LPC) spectrum of hamming windows offset by <5ms are overlaid (A, continuous; B, dahsed). In either of these, small shifts (e.g., from A2 to B2) can lead to large changes in the representation (reproduced from (Smith & Lewicki, 2005)).

coding in most biological systems. One neurophysiological theory that predicts the presence of sparse codes in the neural system is the efficient coding theory (Barlow, 1961) (Simoncelli & Olshausen, 2001). Efficient coding theory states that a sensory system should preserve information about its input while reducing the redundancy of the employed code (Karklin, 2007). As stated earlier, an efficient way of reducing redundancy is to make cell activity as sparse as possible (both in time and space). On the experimental side, Lennie (Lennie, 2003) estimated that given the limited resources of a neuron (i.e., limited energy consumption), the maximum number of active neurons is only 1/50th of any population of cortical neurons at any given time (see also (Baddeley, 1996) for a discussion on the energy efficiency of sparse codes). DeWeese and colleagues (DeWeese et al., 2003), recording from auditory neurons in the rat, have demonstrated that neurons in A1 (a specific cortical area) can reliably produce a signle spike in response to a sound. Also, evidence from olfactory systems in insects, somatosensory neurons in rat, and recording from rat hippocampus all demonstrate highly sparse responses (Graham & Field, 2006).

Sparse coding in its extreme forms a representation called “grandmother cell" code. In such a code, each object in the world (e.g., a grandmother) is represented by a single cell. Some evidence from neurophysiology may be linked to the presence of this very hierarchical repre-

sentation of information (Afraz et al., 2006). However, this coding scheme does not seem to be the prevelant mode of coding in sensory systems.
Sparse coding prevents accidental conjunction of attributes, which is related to the so-called binding problem (Barlow, 1961) (von der Malsburg, 1999) (Wang, 2005) (Pichevar et al., 2006). Accidental conjunction is the process in which different features from different stimuli are associated together, giving birth to illusions or even hallucinations. Although, sparsely coded features are not mutually exclusive, they nonetheless occur infrequently. Therefore, the accidental conjunction occurs rarely and not more frequently than in real life where “illusory conjunction" (the illusion to associate two different features from different stimuli together) occurs rarely.

5. The Mathematics of Sparse Coding

In most cases, in order to generate a sparse representation we need to extract an overcomplete representation. In an overcomplete representation, the number of basis vectors (kernels) is greater than the real dimensionality (number of non-zero eigenvalues in the covariance matrix of the signal) of the input. In order to generate such overcomplete representations, the common approach consists of matching the best kernels to different acoustic cues using different convergence criteria such as the residual energy. However, the minimization of the energy of the residual (error) signal is not sufficient to get an overcomplete representation of an input signal. Other constraints such as sparseness must be considered in order to have a unique solution. Thus, sparse codes are generated using matching pursuit by matching the most optimal kernels to the signal.

5.1 Generating Overcomplete Representations with Matching Pursuit (MP)

Matching Pursuit (MP) is a greedy search algorithm (Tropp, 2004) that can be used to extract sparse representations over an overcomplete set of kernels. Here is a simple analogy showing how MP works. Imagine you want to buy a coffee that costs X units with a limited number of coins of higher and lower values. You first pick higher valued coins until you cannot use them anymore to cover the differnce between the sum of your picked up coins and X. You then switch to lower-valued coins to reach the amount X and continue with smaller and smaller coins till either there is no smaller coin left or you reach X units. MP is doing the exact same thing in the signal domain. It tries to reconstruct a given signal $x(t)$ by decreasing the energy of the atom used to shape the signal at each iteration. In mathematical notations, the signal $x(t)$ can be decomposed into the overcomplete kernels as follow

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

where τ_i^m and a_i^m are the temporal position and amplitude of the i-*th* 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. 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 matching pursuit 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 basis. The signal can be decomposed into

$$x(t) =< x(t), g_m > g_m + r_x(t), \tag{2}$$

where $< x(t), g_m >$ is the inner product between the signal and the kernel and is equivalent to a^m in Eq. 1. $r_x(t)$ is the residual signal.
It can be shown (Goodwin & Vetterli, 1999) that the computational load of the matching pursuit can be reduced, if one saves values of all correlations in memory or finds an analytical formulation for the correlation given specific kernels.

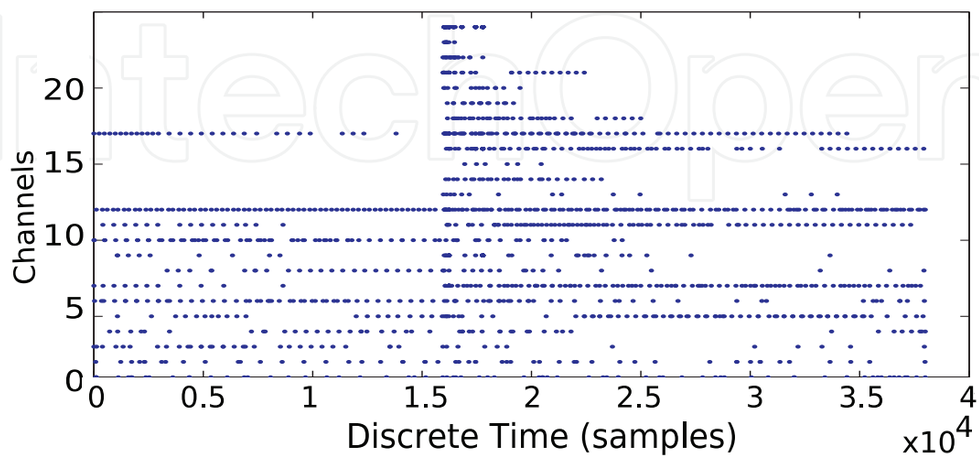


Fig. 3. Spikegram of the harpsichord using the gammatone matching pursuit algorithm (spike amplitudes are not represented). Each dot represents the time and the channel where a spike is fired.

5.2 Shape of Kernels

In the previous section we showed how a signal $x(t)$ can be projected onto a basis of kernels g_m. The question we address in this section is to find optimal bases for different types of signals (e.g., image, audio). As mentioned before, the efficient coding theory states that sensory systems might have evolved to highly efficient coding strategies to maximize the information conveyed to the brain while minimizing the required energy and neural ressources. This fact can be the starting point to finding "optimal waveforms "g_m for different sensory signals.

5.2.1 *Best Kernels for Audio*

Smith and Lewicki (Smith & Lewicki, 2006) found the optimal basis $g_m \in G$ for environmental sounds by maximizing the Maximum Likelihood (ML) $p(x|G)$ given that the prior probability of a spike, $p(s)$, is sparse. Note that the ML part of the optimization deals with the maximization of the information transfer to the brain and the sparseness prior minimizes the energy consumption. Therefore, the optimization here is totally inspired by the efficient coding theory. In mathematical notation, the kernel functions, g_m, are optimized by performing gradient ascent on the log data probability (including ML and sparseness terms),

$$E = \frac{\partial}{\partial g_m} \log p(x|G) = \frac{\partial}{\partial g_m} \left[\log p(x|G, \hat{s}) + \log(p(\hat{s})) \right] \tag{3}$$

If we assume that the noise present in the system is gaussian, Eq. 3 can be rewritten as:

$$E = \frac{1}{\sigma_e} \sum_i a_i^m \left[x - \hat{x} \right]_{\tau_i^m} \tag{4}$$

where $[x - \hat{x}]_{\tau_i^m}$ indicates the residual error over the extent of kernel g_m at position τ_i^m and $\hat{s}$ is the estimated s. At the start of the training, Smith and Lewicki initialized g_m as Gaussian noise and trained (found optimal g_m) by running the optimization on a database of natural sounds. The natural sounds ensemble used in training combined a collection of mammalian vocalizations with two classes of environmental sounds: ambient sounds (rustling brush, wind, flowing water) and transients (snapping twigs, crunching leaves, impacts of stone or woood). Results from optimization show only slight differences between the optimal kernels obtained by Eq. 3 and the gammatone/gammachirp (Irino & Patterson, 2006) family of filters that approximate cochlea in the inner ear (see Fig. 4). However, as pointed out by Smith and Lewicki, totally different kernels will be obtained, if we restrain our training set to only a subclass of environmental sound or if we change the type of signal used as the training set. In the remaining of this chapter, we use the safe assumption that the physiologically optimal kernels for audio are the gammatone/gammachirp filters.

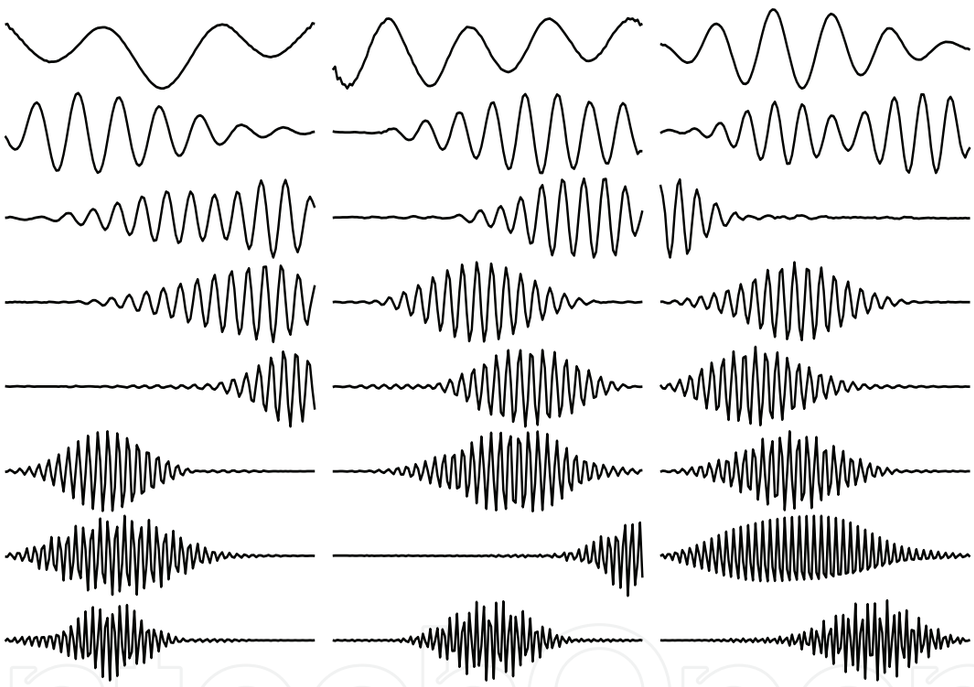


Fig. 4. Efficient coding of a combined sound ensemble consisting of environmental sounds and vocalization yields filters similar to the gammatone/gammachirp family. The impulse response of some of the optimal filters are shown here (reporduced from (Lewicki, 2002)).

5.2.2 *Best kernels for Image*

By using the same efficient coding theory, and by following the same steps as for extracting the optimal basis g_m for audio (i.e., optimizing an ML with sparseness prior and Eq. 3), Olshausen and Field found that the physiologically optimal kernels for image are Gabor wavelets (Olshausen & Field, 1996) (see Fig. 5). Since our focus in this chapter is on audio coding, we refer the reader to (Olshausen & Field, 1996) (among others) for furhter discussion on the extraction of optimal kernels for images.

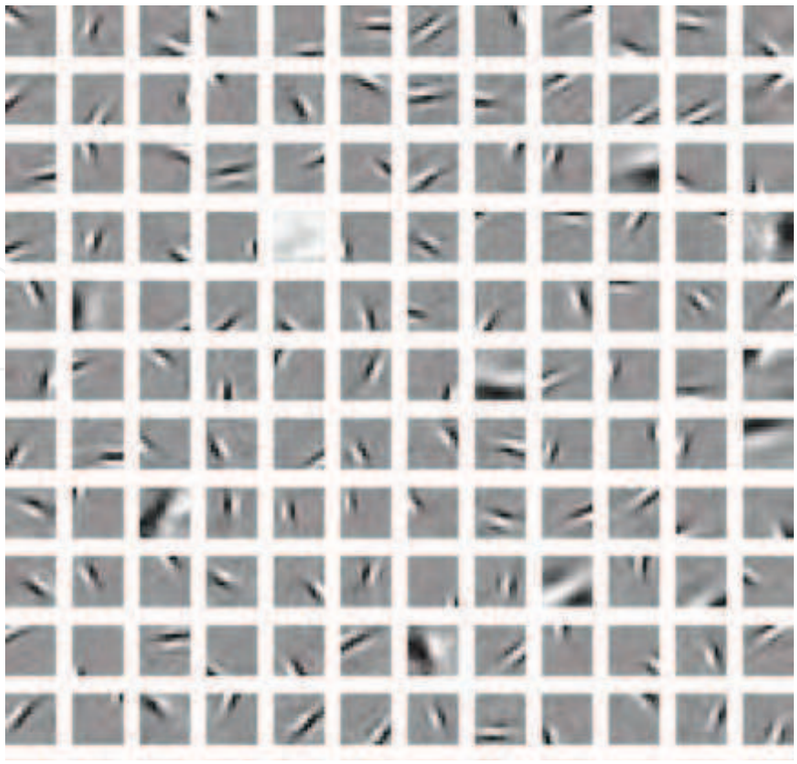


Fig. 5. Results of the search for optimal kernels using maximum likelihood with sparseness prior on 12x12 pixel images drawn from natural scenes. The kernels are Gabor-like. Reproduced from (Olshausen & Field, 1996).

6. A New Paradigm for Audio Coding

In the second half of this chapter, we will briefly describe the biologically-inspired audio coder we have developped based on the concepts already presented in the first half of this chapter (i.e., sparse coding).

6.1 The Bio-Inspired Audio Coder

The analysis/synthesis part of our universal audio codec is based on the generation of auditory-inspired 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 (see section 6.2.2). The adaptiveness is a key feature we introduced in Matching Pursuit (MP) to increase the efficiency of the proposed method (see section 6.2.2). An auditory masking model has been developed and integrated into the MP algorithm to extract audible spikes (see section 7). In addition a differential encoder of spike parameters based on graph theory is proposed in (Pichevar, Najaf-Zadeh, Lahdili & Thibault, 2008). The quantization of the spikes is given in section 8. We finally propose a frequent pattern discovery block in section 10. The block diagram of all the building blocks of the receiver and transmitter of our proposed universal audio coder is depicted in Fig. 6 of which the graph-based optimization of the differential encoder is explained in (Pichevar, Najaf-Zadeh, Lahdili & Thibault, 2008).

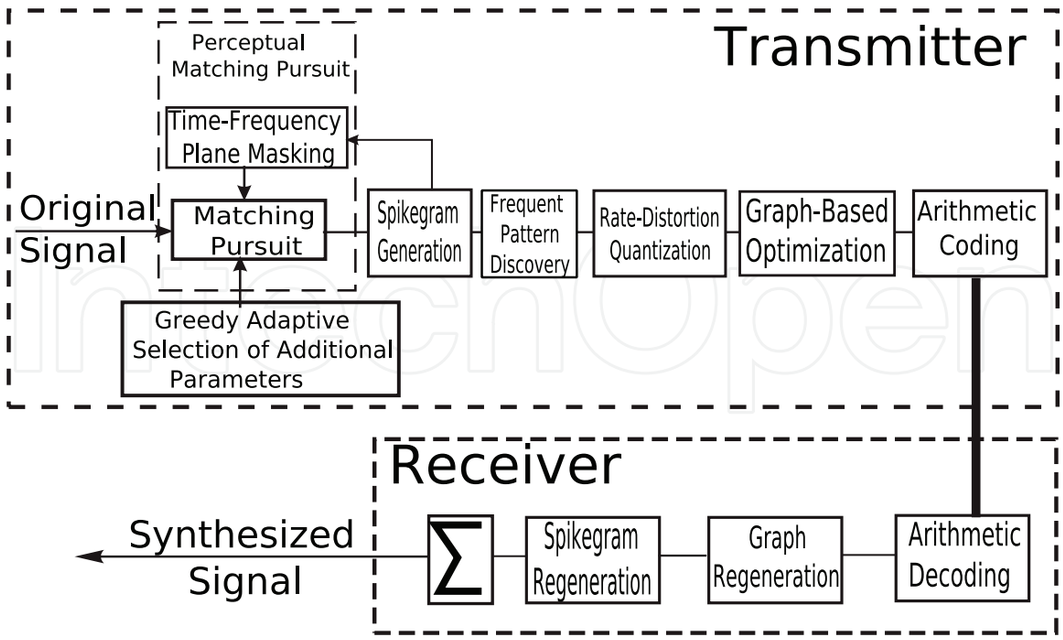


Fig. 6. Block diagram of our proposed Universal Bio-Inspired Audio Coder.

6.2 Generation of the spike-based representation

We use here the concept of generating sparse overcomplete representations as described in section 5 to design a biologically-inspired sparse audio coder. In section 5.2, we saw that the gammatone family of kernels is the optimal class of kernels according to the efficient coding theory. Therefore, they are used in our approach. In addition, using asymmetric kernels such as gammatone/gammachirp atoms is that they do not create pre-echos at onsets (Goodwin & Vetterli, 1999). However, very asymmetric kernels such as damped sinusoids (Goodwin & Vetterli, 1999) are not able to model harmonic signals suitably. On the other hand, gammatone/gammachirp kernels have additional parameters that control their attack and decay parts (degree of symmetry), which are modified suitably according to the nature of the signal in our proposed technique. As described in section 5, the approach used to find the projections is an iterative one. In this section, we will compare two variants of the projection technique. The first variant, which is non-adaptive, is roughly similar to the general approach used in (Smith & Lewicki, 2006), which we applied to the specific task of audio coding. However, we proposed the second adaptive variant in (Pichevar et al., 2007), which takes advantage of the additional parameters of the gammachirp kernels and the inherent nonlinearity of the auditory pathway (Irino & Patterson, 2001)(Irino & Patterson, 2006). Some details on each variant are given below.

6.2.1 Non-Adaptive Paradigm

In the non-adaptive paradigm, only gammatone filters are used. The impulse response of a gammatone filter is given by

$$g(f_c, t) = t^3 e^{-2\pi bt} \cos(2\pi f_c t) \quad t > 0, \tag{5}$$

where f_c is the center frequency of the filter, distributed on Equal Rectangular Bandwith (ERB) scales. At each step (iteration), the signal is projected onto the gammatone kernels (with different center frequencies and different time delays). The center frequency and time delay that give the maximum projection are chosen and a spike with the value of the projection is added to the "auditory representation" at the corresponding center frequency and time delay (see Fig. 3). The signal is decomposed into the projections on gammatone kernels plus a residual signal $r_x(t)$ (see Eqs. 1 and 2).

6.2.2 Adaptive Paradigm

In the adaptive paradigm, gammachirp filters are used. The impulse response of a gammachirp filter with the corresponding tuning parameters (b,l,c) is given below

$$g(f_c, t, b, l, c) = t^{l-1} e^{-2\pi bt} \cos(2\pi f_c t + c \quad \ln t) \quad t > 0. \tag{6}$$

It has been shown that the gammachirp filters minimize the scale/time uncertainty (Irino & Patterson, 2001). 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, l and b control the attack and decay of the kernels. However, searching the three parameters in the parameter space is a very computationally intensive task. Therefore, we use a suboptimal search (Gribonval, 2001) in which, we use the same gammatone filters as the ones used in the non-adaptive paradigm with values of l and b given in (Irino & Patterson, 2001). This 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.

$$G_{max1} = \underset{f,t_0}{\operatorname{argmax}} \{|< r, g(f, t_0, b, l, c) >|\}, \quad g \in G \tag{7}$$

$$G_{max2} = \underset{f,t_0}{\operatorname{argmax}} \{|< r, g(f, t_0, b, l, c) >|\}, \quad g \in G - G_{max1} \tag{8}$$

For the sake of simplicity, we use f instead of f_c in Eqs. 8 to 11. 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 $g \in G_{max1} \cup G_{max2}$.

$$G_{maxc} = \underset{c}{\operatorname{argmax}} \{|< r, g(f, t_0, b, l, c) >|\}. \tag{9}$$

We then use the information found in the second step to find the best b for $g \in G_{maxc}$ in Eq. 10, and finally find the best l among $g \in G_{maxb}$ in Eq. 11.

$$G_{maxb} = \underset{b}{\operatorname{argmax}} \{|< r, g(f, t_0, b, l, c) >|\} \tag{10}$$

$$G_{maxl} = \underset{l}{\operatorname{argmax}} \{|< r, g(f, t_0, b, l, c) >|\}. \tag{11}$$

Therefore, six parameters are extracted in the adaptive technique for the "auditory representation": center frequencies, chirp factors (c), time delays, spike amplitudes, b, and l. The last two parameters control the attack and the decay slopes of the kernels. Although, there are additional parameters in this second variant, as shown later, the adaptive technique contributes to better coding gains. The reason for this is that we need a much smaller number of filters (in the filterbank) and a smaller number of iterations to achieve the same SNR, which roughly reflects the audio quality.

	Speech		Castanet		Percussion	
	Adapt.	Non-Adapt.	Adapt.	Non-Adapt.	Adapt.	Non-Adapt.
Number of spikes	10492	35208	6510	24580	9430	29370
Spike gain	0.13N	0.44N	0.08N	0.30N	0.12N	0.37N
Bitrate (bit/sample)	1.98	3.07	1.54	3.03	1.93	2.90

Table 1. Comparison of the adaptive and non-adaptive schemes for spike generation for three different audio signals. The average saving in bitrate over all materials is around 45%. N is the signal length (number of samples in the signal).

6.3 Comparison of Adaptive and Non-Adaptive Paradigms

In this section we compare the performance of the adaptive and non-adaptive schemes. Results and a comparison of the two different schemes in terms of bitrate and number of spikes extracted for high quality (scale 4 on ITU-R impairment scale) are given in Table 1. With the adaptive scheme, we observe an average drop of 45% in the bitrate compared to the non-adaptive approach. The spike gain (decrease in spikes for a given signal of N samples) decreases drastically when the adaptive paradigm is used as well. Fig. 7 compares the adaptive to the non-adaptive approach for different numbers of cochlear channels (the number of center frequencies used in the gammatone kernels).

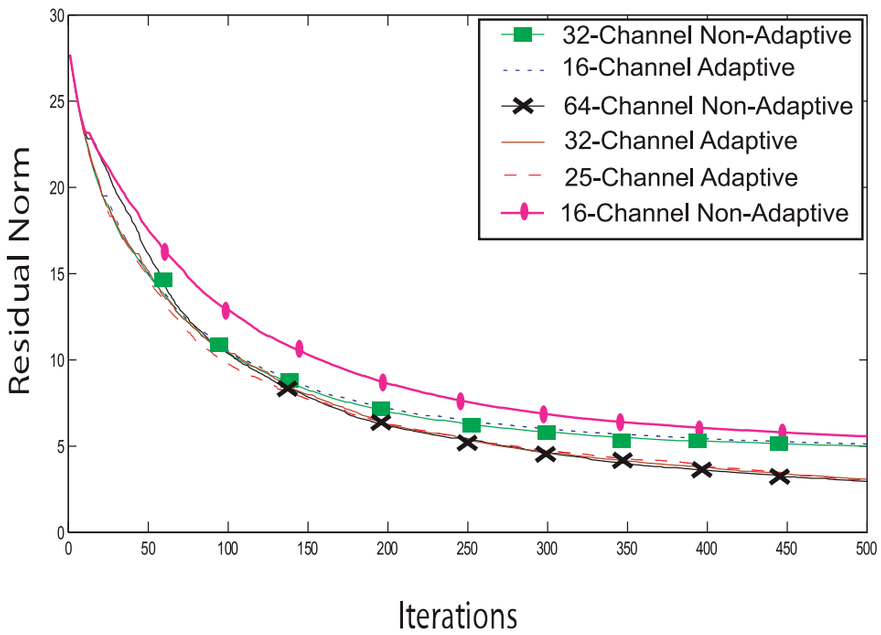


Fig. 7. Comparison of the adaptive and non-adaptive spike coding schemes of speech for different number of channels. In this figure, only the chirp factor is adapted. For the case where all three parameters are adapted see Fig. 3 in (Pichevar et al., 2007).

7. Masking of Spikes

In previous sections, we showed how sparse representations based on spikes can be generated using our proposed alogrithm based on MP. We also showed how we can increase the performance of our system by shaping adaptively the kernels used in MP to our signal. However, in the previous section we generated a sparse signal that is close to the original signal in the mean-squared error sense and we ignored the effects of perceptual masking on the generation of the signal. In fact, we will see later in this section that some of the spikes generated in section 6 are not perceptible by the human ear. To this end, we will first review the basics of masking in the auditory system.

7.1 Fundamentals of Masking

Auditory masking occurs when the perception of one sound (the maskee) is affected by the presence of another sound (the masker). This happens because the original neural activity caused by the first signal is reduced by the neural activity of the other sound in the brain. Masking can be classified in two distinct categories. In temporal (non-simultaneous masking) the masker and the maskee are not present at the same time. In the case of a spikegram, the temporal masking is present when two spikes are fired in the same channel (two dots in the same horizontal line on the spikegram) and are relatively close in time. On the other hand, simultaneous masking happens when two spikes fire at the same time in different channels. Fig. 8 outlines the mechanism of different types of masking. The masker (a spike on the spikegram) is presented. This masker can potentially mask another spike (or make the presence of another spike inaudible) if the latter falls within the pre- or post-synaptic curves with an amplitude below the curve or if it is applied simultaneously with a frequency close enough to the masker and with the appropriate amplitude.

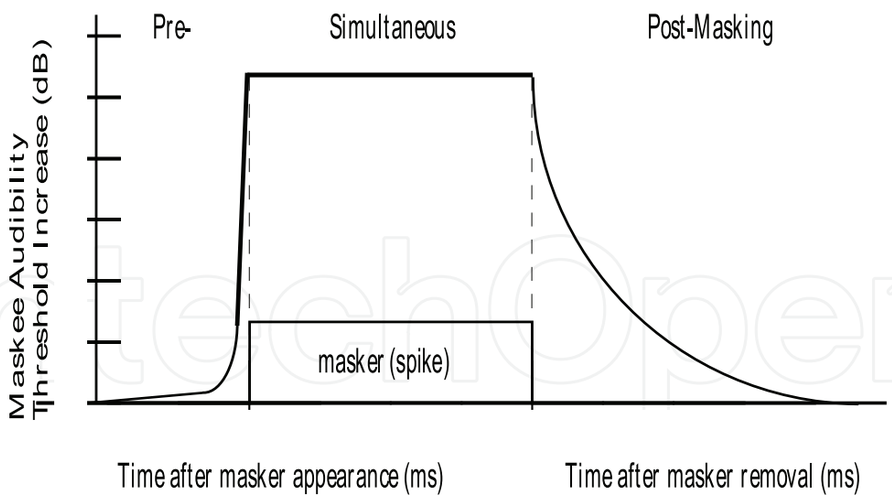


Fig. 8. Temporal masking of the human ear. Pre-masking occurs prior to masker onset and lasts only a few milliseconds; Post-masking may persist for more than 100 milliseconds after masker removal (after (Painter & Spanias, 2000))

Audio Material	Mean Subj. Score for MP	Mean Sub. Score for PMP	PEAQ Obj. Score for MP	PEAQ Obj. Score for PMP
Susan Vega	-0.7500	-0.9667	-0.593	-0.330
Trumpet	-0.9667	-0.4333	-1.809	-0.791
Orchestra	-0.7667	-0.4000	-1.239	-0.915
Harpsichord	-0.5000	-0.3667	-1.699	-0.867
Bagpipe	-0.4000	-0.2000	-0.765	-0.502
Glockenspiel	-0.2000	-0.333	-0.925	-1.266
Plucked Strings	-0.4667	-0.2667	-1.050	-1.050

Table 2. Mean subjective and objective scores for a few audio files processed with MP and PMP. Objective Difference Grade (ODG) are shown in the table for subjective tests.

7.2 Perceptual Matching Pursuit

Based on the masking mechanism explained above, we proposed in (Najaf-Zadeh et al., 2008) the Perceptual Matching Pursuit, which basically extend MP to the perceptual domain. By doing so, only an audible kernel is extracted at each iteration. Moreover, contrary to the matching pursuit algorithm, PMP will stop decomposing an audio signal once there is no audible part left in the residual. Details of how the pre- and post-masking curves as in Fig. 8 are extracted for spikegrams as well as the simultaneous masking are given in (Najaf-Zadeh et al., 2008). Here, in table 2 we give results on how PMP retains the same audio quality as MP. In order to verify the objective scores, we conducted a semi-formal listening test, based on the ITU.R BS. 1116 method, to evaluate the quality of the test signals. Six subjects took part in a Ştriple stimulus hidden referenceŤ test and listened to the audio materials (presented in Table 2) over the headphone in a quiet room. The CRC SEAQ software was used in the test which allowed the listener to seamlessly switch among the three stimuli. In each trial, the stimuli ŞAŤ was always the reference stimulus known by the subject. Two other stimuli, ŞBŤ or ŞCŤ, were either a hidden reference, identical to ŞAŤ, or a synthesized version of the same audio material. None of ŞBŤ or ŞCŤ was known to the subject. The listener had to identify the synthesized version (either ŞBŤ or ŞCŤ) and to grade its quality relative to that of the reference on ŞAŤ. The grading scale was continuous from 1 (very annoying) to 5 (no difference between the reference and the synthesized file). The average subjective scores for MP and PMP were 4.4214 and 4.5762, and the standard deviations of the scores were 0.2522 and 0.2612 respectively. Values in Table 2 are the mapping of the subjective test scores (between 1 and 5) to the Objective Difference Grade (ODG) that varies between -4 to +4, according to the ITU.R BS 1116 standard. Positive values in the ODG represents evaluation errors by subjects (basically errors in identifying the hidden reference), while negative values are the subjective scores, with 0 being the case where no difference between the reference and the coded material is detected by the subject[3] and -4 representing the biggest difference between the reference and the coded signal. Although the confidence intervals for the subjective scores are overlapping, the majority of the test materials received higher subjective scores for PMP, which is consistent with the objective evaluation. The reader may notice that PMP reduces the total number of spikes to be extracted for the same audio quality, thus requires lower bitrate for the same audio quality.

[3] http://www.itu.int/rec/R-REC-BS.1116-1-199710-I/e

8. Quantization of Spikes

The amplitude of spikes generated in spikegrams should be quantized before transmission. The cost function we use to find the optimal levels of quantization is 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):

$$\widehat{E}(\mathbf{q}) = D + \lambda R = \frac{\|\sum_i \hat{\alpha}_i g_i - \sum_i \alpha_i g_i\|^2}{\|\sum_i \alpha_i g_i + \eta\|^\gamma} + \lambda H(\hat{\mathbf{ff}}), \tag{12}$$

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. $\hat{\mathbf{ff}}$ is the vector of quantized amplitudes and is computed as follows:

$$\hat{\alpha}_i = q_i \quad \text{if} \quad q_{i-1} < \alpha_i < q_i \tag{13}$$

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

$$H(\hat{\mathbf{ff}}) = -\sum_i p_i(\hat{\alpha}_i) \log_2 p_i(\hat{\alpha}_i), \tag{14}$$

where $p_i(\hat{\alpha}_i)$ is the probability density function of $\hat{\alpha}_i$. The way the quantizer is defined in Eq. 13 reduces the dead zone problem (defined in (Neff & Zakhor, 2000)). 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. 12) is to give a better reconstruction of low-energy parts of the signal.

Note that in Eq. 12 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, in (Pichevar, Najaf-Zadeh, Thibault & Lahdili, 2008) we derived optimal quantization levels using Genetic Algorithm (GA) (Mitchell, 1998). Results obtained in (Pichevar, Najaf-Zadeh, Thibault & Lahdili, 2008) showed that the optimal quantizer is not linear in the case of spikegrams.

9. 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. 9 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.

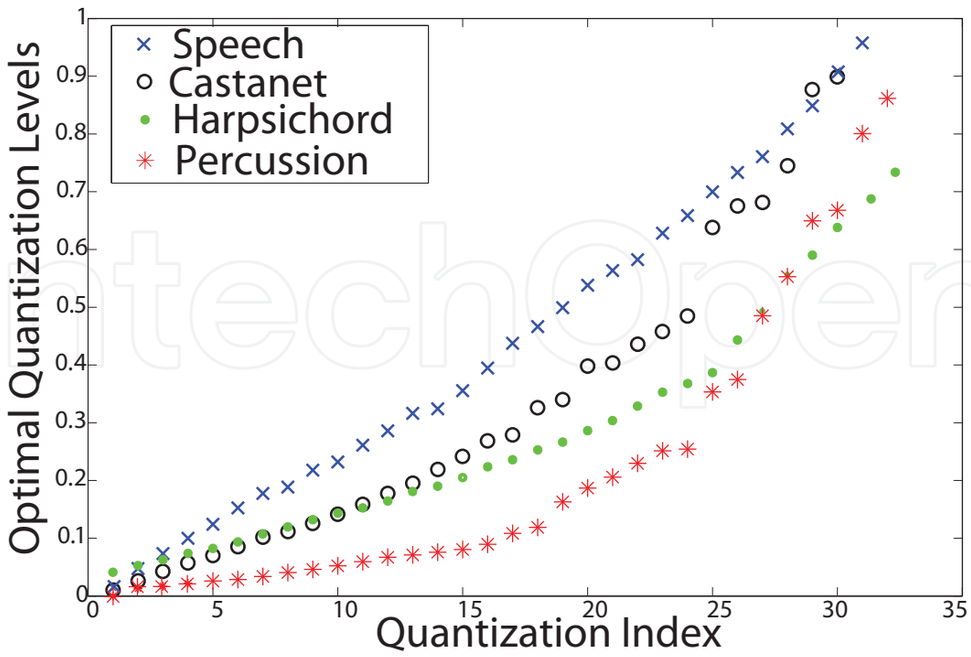


Fig. 9. Optimal quantization levels for different sound categories. Spike amplitudes are normalized to one.

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 $h_t = \text{sign}(h)$ 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.

9.1 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-SEAQ (see Table 3) 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 (Pichevar et al., 2007) for the unquantized case and values for amplitude quantization in Table 3 of this article.

	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 3. Comparison of 32-level and 64-level piecewise uniform quantizers for different audio signals. A CRC-SEAQ score between 0 and -1 is associated with transparent quality. No codebook side information is sent to the receiver in this case.

10. Extraction of Patterns in Spikegrams

As mentioned in previous sections, the spike activity of each channel can be associated to the activity of a neuron tuned to the center frequency of that channel. The ultimate goal in the pattern recognition paradigm is to find a generative neural architecture (such as a synfire chain (Abeles, 1991) or a polychronous network (Izhikevich, 2006)) that is able to generate a spikegram such as the one we extract by MP (see Fig. 3) for a given audio signal. However, for the time being we proposed a solution to a simplified version of the aforementioned problem in (Pichevar & Najaf-Zadeh, 2009). In fact, we propose to extract “channel-based or frequency-domain patterns" in our generated spikegrams using temporal data mining (Mannila et al., 1997) (Patnaik et al., 2008). Since these patterns are repeated frequently in the signal and are the building blocks of the audio signal, we may call them auditory objects (Bregman, 1994). In contrast with other approaches (i.e., Harmonic Matching Pursuit and Meta-Molecular Matching Pursuit) that are able to extract some predefined sound structures such as harmonics (Krstulovic et al., 2005) very precisely, our proposed approach is able to extract patterns without any a priori knowledge of the type of structure present in the sound. The reader may also refer to (Karklin, 2007) and (Karklin & Lewicki, 2009) for another approach to extract statistical dependencies in a sparse representation that uses latent (hidden) variables to exploit higher order statistics.

10.1 Frequent Episode Discovery

The frequent episode discovery framework was proposed by Mannila and colleagues (Mannila et al., 1997) and enhanced in (Laxman et al., 2007). Patnaik *et al.* (Patnaik et al., 2008) extended previous results to the processing of neurophysiological data. The frequent episode discovery fits in the general paradigm of temporal data mining. The method can be applied to either serial episodes (ordered set of events) or to parallel episodes (unordered set of events). A frequent episode is one whose frequency exceeds a user specified threshold. Given an episode occurrence, we call the largest time difference between any two events constituting the occurrence as the span of the occurrence and we use this span as a temporal constraint in the algorithm. The details of the algorithm can be found in (Pichevar & Najaf-Zadeh, 2009).
In (Pichevar & Najaf-Zadeh, 2009), we showed that the extraction of patterns in the spikegram is biased towards denser regions. Therefore we proposed a 3-pass extraction algorithm in which extracted patterns are subtracted from the original signal at each pass and explore sparser regions as well.
Fig. 10 shows the extracted patterns for each of the three distinct passes for percussion. Since unordered episodes are discovered, the order of appearance of spikes in different channels

Percussion	Pass 1	Pass 2	Pass 3	Overall
No. extracted spikes	1682	771	335	2788
No. codebook elements	47	36	11	94
Codebook size in bits	2200	1976	320	4496
Raw bit saving	9968	4403	1820	16191
Effective bit saving	7768	2427	1500	11695
Castanet	Pass 1	Pass 2	Pass 3	Overall
No. extracted Spikes	596	684	580	1860
No. codebook elements	8	20	37	65
Codebook size in bits	440	1436	2340	4216
Raw bit saving	2660	4095	3253	10008
Effective bit saving	2220	2659	913	5792
Speech	Pass 1	Pass 2	Pass 3	Overall
No. extracted Spikes	1262	689	395	2346
No. codebook elements	8	21	11	40
Codebook size in bits	338	1053	288	1679
Raw bit saving	3238	3859	2250	11026
Effective bit saving	2899	2806	1962	7667

Table 4. Results for a 3-Pass pattern extraction on 1-second frames. **Percussion:** The total number of bits to address channels when no pattern recognition is used equals 23704 and the saving in addressing channels due to our algorithm is 49%. **Castanet:** The total number of bits to address channels when no pattern recognition is used is 21982 and there is a saving of 26% with our proposed algorithm. **Speech:** The total number of bits to address channels when no pattern recognition is used is 19118 and there is a saving of 40%.

can change for a given pattern. However, the channels in which spike activity occurs are the same for all similar patterns. In other words the patterns are similar up to a permutation in the order of appearance of each spike. Fig. 10 also shows that our 3-pass algorithm is able to extract patterns in the high, low and mid-frequency ranges, while a 1-pass algorithm would have penalized some sparser spikegram regions.

In Table 4, the number of extracted spikes is shown for each pass and the raw bit saving and effective bit saving in addressing channels as described above are given for percussion, castanet, and speech. Our algorithm was able to extract between 1860 and 2788 spikes in different episodes out of the total 4000 spikes. The longest pattern found in percussion is 13-spike long and is repeated on average 17 times in the signal frame, while the longest pattern for castanet is 14-spike long and is repeated 33 times on average in frames. In the meantime, the longest pattern for speech is 100-spike element and is repeated 8 times on average in the frames. Results show that the bitrate coding gain obtained in addressing frequency channels ranges from 26 % to 49% depending on signal type. Note that since the pattern extraction coding is lossless, the informal subjective quality evaluations in (Pichevar et al., 2007) for the audio materials still hold when our new audio extraction paradigm is applied.

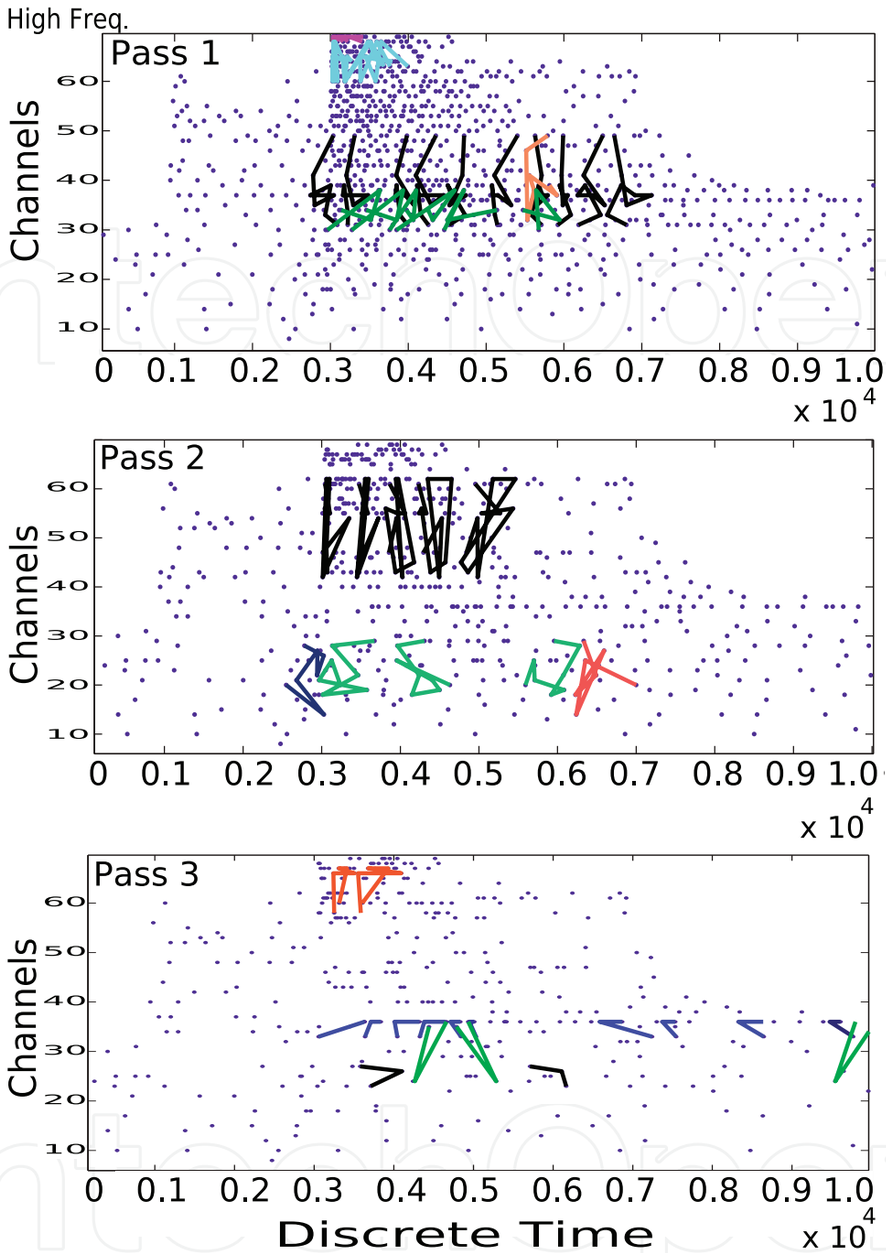


Fig. 10. Spikegrams (dots) and the most *relevant* extracted patterns (lines) at each of the 3 passes for percussion for a 250 ms frame. Different colors/grayscales represent different episodes. Only spikes not discovered during the previous pass are depicted at each pass. Note that since unordered episodes are discovered, patterns are similar up to a permutation in the temporal order.

11. Conclusion

We outlined in this chapter, a new type of emergent signal coding called sparse representation, which is able to preserve the exact timing of acoustical events and edges of images.

Sparse codes have also other interesting properties such as shift invariance. Furthermore, we discussed the biological plausibiltiy of sparse representations in the brain when it comes to the processing of audio and video. We then described our proposed audio coder, which is a merger of sparse coding with biologically-inspired coding. We showed how sparse code generation (spikegrams), masking, quantization and pattern extraction can be done in our proposed framework. Our proposed approach is a first step towards the development of an object-based universal audio coder. Object-based coders belong to a totally new generation of coders that can potentially reduce current achievable bitrates by an order of magnitude; as an analogy, consider the amount of information we can save by transmitting only the color, radius, and position of a given circle in a visual scene instead of sending all pixels of that circle one by one!

In a future work, we will extract the structural dependencies of spike amplitudes and/or other parameters in the spikegram such as the chirp factor, etc. We will also investigate the design of a generative neural model based on spikegrams. Formal subjective listening tests for the overall system will be conducted in the future. In order to speed up the spikegram extraction of audio signals, we have conducted preliminary tests on replacing the MP stage (see Fig. 6) by neural circuitry that can be implemented on embedded and parallel hardware. We will further explore this avenue in a future work. The application of different ideas outlined in this chapter (i.e., pattern recognition and masking model) are not limited to spikegrams and can be applied to other sparse representations found in the literature. In addition, the frequency episode discovery algorithm discussed in this article can be used in other applications such as speech recognition, sound source separation, and audio classification.

12. References

Abeles, M. (1991). *Corticonics: Neural circuits of the cerebral cortex*, Cambridge University Press.

Afraz, S., Kiani, R. & Esteky, H. (2006). Microstimulation of inferotemporal cortex influences face categorization, *Nature* **442**: 692–695.

Baddeley, R. (1996). An efficient code in V1?, *Nature* **381**: 560–561.

Barlow, H. (1961). Possible principles underlying the transformation of sensory messages, *Sensory Communication* pp. 217–234.

Bregman, A. (1994). *Auditory Scene Analysis: The Perceptual Organization of Sound*, MIT Press.

DeWeese, M., Wehr, M. & Zador, A. (2003). Binary spiking in auditory cortex, *J. Neuroscience* **23**: 7940–7949.

Doi, E., Balcan, D. & Lewicki, M. (2007). Robust coding over noisy overcomplete channels, *IEEE Transactions on Image Processing* **16**(2): 442–452.

Doi, E. & Lewicki, M. (2005). Sparse coding of natural images using an overcomplete set of limited capacity units, *Advances in Neural Information Processing Systems*, pp. 442–452.

Field, D. (1994). What is the goal of sensory coding?, *Neural Computation* **6**: 559–601.

Furber, S., Brown, G., Bose, J., Cumpstey, J., Marshall, P. & Shapiro, J. (2007). Sparse distributed memory using rank-order neural codes, *IEEE Transactions on Neural Networks* **18**(3): 648–659.

Goodwin, M. & Vetterli, M. (1999). Matching pursuit and atomic signal models based on recursive filter banks, *IEEE Transaction on signal processing* **47**(7): 1890–1902.

Graham, D. & Field, D. (2006). Sparse coding in the neocortex, *Evolution of Nervous Sys. ed. J. H. Kaas and L. A. Krubitzer* .

Gribonval, R. (2001). Fast matching pursuit with a multiscale dictionary of gaussian chirps, *IEEE Transaction on signal processing* **49**(5): 994–1001.

Haykin, S. (2008). *Neural networks: a comprehensive foundation*, Prentice Hall.
Hoyer, P. (2004). Non-negative matrix factorization with sparseness constraints, *Journal of Machine Learning Research* **5**: 1457–1469.
Hyvarinen, A., Hurri, J. & Hoyer, P. (2009). *Natural Image Statistics- A probalistic approach to early computational vision*, Springer-Verlag, Berlin.
Irino, T. & Patterson, R. (2001). A compressive gammachirp auditory filter for both physiological and psychophysical data, *JASA* **109**(5): 2008–2022.
Irino, T. & Patterson, R. (2006). A dynamic compressive gammachirp auditory filterbank, *IEEE Trans. on Audio and Speech Processing* **14**(6): 2008–2022.
Izhikevich, E. (2006). Polychronization: Computation with spikes, *Neural Computation* **18**: 245–282.
Karklin, Y. (2007). *Hierarchical statistical models of computation in the visual cortex*, PhD thesis, Carnegie Mellon University.
Karklin, Y. & Lewicki, M. (2005). A hierarchical bayesian model for learning non-linear statistical regularities in non-stationary natural signals, *Neural Computation* **17**(2): 397–423.
Karklin, Y. & Lewicki, M. (2009). Emergence of complex cell properties by learning to generalize in natural scenes, *Nature* **457**: 83–86.
Krstulovic, S., Gribonval, R., Leveau, P. & Daudet, L. (2005). A comparison of two extensions of the matching pursuit algorithm for the harmonic decomposition of sounds, *Workshop on the Applications of Signal Processing to Audio and Acoustics, New Platz, New York*.
Laxman, S., Sastry, P. & Unnikrishnan, K. (2007). Discovery of frequent generalized episodes when events persist for different durations, *IEEE Trans. on Knowledge and Data Eng.* **19**: 1188–1201.
Lee, D. & Seung, S. (1999). Learning the parts of objects by non-negative matrix factorization, *Nature* **401**: 788Ű791.
Lennie, P. (2003). The cost of cortical computation, *Curr. Biol.* **13**: 493–497.
Lewicki, M. (2002). Efficient coding of natural sounds, *Nature Neuroscience* **5**: 356–363.
Mannila, H., Toivonen, H. & Verkamo, A. (1997). Discovery of frequent episodes in event sequences, *Data Mining and Knowledge Discovery* **1**: 259–289.
Mitchell, M. (1998). *An Introduction to Genetic Algorithms (Complex Adaptive Systems)*, MIT Press.
Najaf-Zadeh, H., Pichevar, R., Thibault, L. & Lahdili, H. (2008). Perceptual matching pursuit for audio coding, *Audio Engineering Society Convetion, Amsterdam, The Netherlands*.
Neff, R. & Zakhor, A. (2000). Modulus quantization for matching pursuit video coding, *IEEE Trans. on Circuits and Systems for Video Technology* **10**(6): 895–912.
Olshausen, B. & Field, D. (1996). Emergence of simple-cell receptive field properties by learning a sparse code for natural images, *Nature* **381**: 607–609.
Painter, T. & Spanias, A. (2000). Perceptual coding of digital audio, *Proceedings of the IEEE* **88**: 451–515.
Patnaik, D., Sastry, P. & Unnikrishnan, K. (2008). Inferring neural network connectivity from spike data: A temporal mining approach, *Scientific Programming* **16**: 49–77.
Pichevar, R. & Najaf-Zadeh, H. (2009). Pattern extraction in sparse representations with application to audio coding, *European Signal Processing Conf., Glasgow, UK*.
Pichevar, R., Najaf-Zadeh, H., Lahdili, H. & Thibault, L. (2008). Differential graph-based coding of spikes in a biologically-inspired universal audio coder, *Audio Engineering Society Convetion, Amsterdam, The Netherlands*.

Pichevar, R., Najaf-Zadeh, H. & Thibault, L. (2007). A biologically-inspired low-bit-rate universal audio coder, *Proceedings of Audio Engineering Society Convention*.

Pichevar, R., Najaf-Zadeh, H., Thibault, L. & Lahdili, H. (2008). Entropy-constrained spike modulus quantization in a bio-inspired universal audio coder, *European Signal Processing Conf., Switzerland*.

Pichevar, R. & Rouat, J. (2008). An improved sparse non-negative part-based image coder via simulated annealing and matrix pseudo-inverse, *IEEE Conf. on Audio, Speech, and Signal Proc., Las Vegas, USA*.

Pichevar, R., Rouat, J. & Tai, L. (2006). The oscillatory dynamic link matcher for spiking-neuron-based pattern recognition, *Neurocomputing* **69**: 1837–1849.

Simoncelli, E., Freeman, W., Adelson, E. & Heeger, D. (1992). Shiftable multi-scale transforms, *IEEE Transactions on Information Theory* **38**(2): 587–607.

Simoncelli, E. & Olshausen, B. (2001). Natural image statistics and neural representation, *Annual Review of Neuroscience* **24**: 1193–1216.

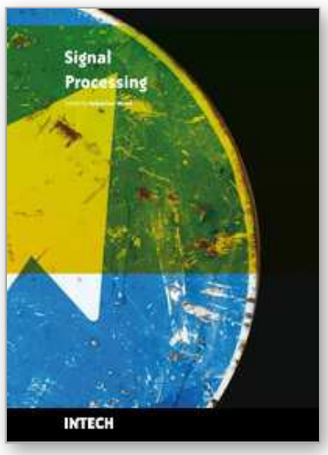
Smith, E. & Lewicki, M. (2005). Efficient coding of time-relative structure using spikes, *Neural Computation* **17**: 19–45.

Smith, E. & Lewicki, M. (2006). Efficient auditory coding, *Nature* **7079**: 978–982.

Tropp, J. (2004). Greed is good: Algorithmic results for sparse approximation, *IEEE Trans. on information theory* **50**(10): 2231–2242.

von der Malsburg, C. (1999). The what and why of binding: the modelers perspective, *Neuron* **24**: 692–695.

Wang, D. (2005). The time dimension for scene analysis, *IEEE Trans. On Neural Networks* **16**: 1401–1426.



Signal Processing

Edited by Sebastian Miron

ISBN 978-953-7619-91-6
Hard cover, 528 pages
Publisher InTech
Published online 01, March, 2010
Published in print edition March, 2010

This book intends to provide highlights of the current research in signal processing area and to offer a snapshot of the recent advances in this field. This work is mainly destined to researchers in the signal processing related areas but it is also accessible to anyone with a scientific background desiring to have an up-to-date overview of this domain. The twenty-five chapters present methodological advances and recent applications of signal processing algorithms in various domains as telecommunications, array processing, biology, cryptography, image and speech processing. The methodologies illustrated in this book, such as sparse signal recovery, are hot topics in the signal processing community at this moment. The editor would like to thank all the authors for their excellent contributions in different areas of signal processing and hopes that this book will be of valuable help to the readers.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen