A Biologically-Inspired Low-Bit-Rate Universal Audio Coder

Ramin Pichevar¹, Hossein Najaf-Zadeh¹, Louis Thibault¹

¹Advanced Audio Systems, Communications Research Centre, Ottawa, Canada

Correspondence should be addressed to Ramin Pichevar (ramin.pishehvar@crc.ca)

ABSTRACT

We propose a new biologically-inspired paradigm for universal audio coding based on neural spikes. Our proposed approach 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. A masking model is applied to the spikegrams to remove inaudible spikes and to increase the coding efficiency. The paradigm proposed in this paper is a first step towards the implementation of a high-quality audio encoder by further processing acoustical events generated in the spikegrams. Upon necessary optimization and fine-tuning our coding system, operating at 1 bit/sample for sound sampled at 44.1 kHz, is expected to deliver high quality audio for broadcast applications and other applications such as archiving and audio recording.

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 (e.g., 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/analysis are block-based, i.e. the signal is processed piece-wise 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
The information redundancy from the raw signal should be reduced. A desirable code/representation should capture the underlying 2D-time-frequency structures, so that they are more directly observable and well represented at low bit rates [11].

The aim of this article is to propose a shift-invariant representation that extracts acoustic events without smearing them, while providing coding efficiency. We will then show how this improved representation can be efficiently applied to audio coding by using adequate information coding and masking strategies. Comparison with similar techniques will be given afterwards.

In the remainder of this section we will give a brief survey of different coding schemes to justify our choices for our proposed approach.

1.1. Block-Based Coding
Most of the signal representations used in speech and audio coding are block based (i.e., DCT, MDCT, FFT). In the block-based coding scheme, the signal is processed piecewise in a series of discrete blocks, causing temporally smeared transients and non-stationary periodicities. On the other hand, 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.

1.2. Filterbank-Based Shift-Invariant Coding
In the filterbank design paradigm, the signal is continuously applied to the filters of the filterbank and its convolution with the impulse responses are computed. Therefore, the outputs of these filters are shift invariant. This representation does not have the drawbacks of block-based coding mentioned above, such as time variance. However, filterbank analysis is not sufficient for designing a general sound processing algorithm. Another important aspect not taken into account in this paradigm is coding efficiency or, equivalently the ability of the representation to capture underlying structures in the signal. A desirable code/representation should reduce the information redundancy from the raw signal so that the underlying structures are more directly observable. However, convolutional representations (i.e., filterbank design) increases the dimensionality of the input signal.

1.3. Overcomplete Shift-Invariant Representations
In an overcomplete basis, 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. The 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. Overcomplete representations have been advocated because they have greater robustness in the presence of noise. They are also a way to maximize information transfer, when different regions/objects of the underlying signal have strong correlations [4]. In other terms, the peakiness of values can be exploited efficiently in entropy coding. In order to find the “best matching kernels”, matching pursuit is used.

1.3.1. Generating Overcomplete Representations with Matching Pursuit (MP)
In mathematical notations, the signal \( x(t) \) can be decomposed into the overcomplete kernels as follow:

\[
x(t) = \sum_{m=1}^{M} \sum_{i=1}^{n_m} a^m_i g^m(t - \tau^m_i) + \epsilon(t)
\]  

where \( \tau^m_i \) and \( a^m_i \) 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 \( \tau^m_i \), \( a^m_i \), 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
\[ x(t) = < x(t), g_m > g_m + R_x(t) \]  \hspace{1cm} (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 [3] that the computational load of the matching pursuit can be reduced, if one can save values of all correlations in memory or can find an analytical formulation for the correlation given specific kernels.

2. A NEW PARADIGM FOR AUDIO CODING

2.1. Generation of the spike-based representation

We propose an auditory sparse and overcomplete representation suitable for audio compression. In this paradigm the signal is decomposed into its constituent parts (kernels) by a matching pursuit algorithm. We use gammatone/gammachirp filterbanks as projection basis as proposed by [11] [10]. The advantage of using asymmetric kernels such as gammatone/gammachirp atoms is that they do not create pre-echos at onsets [3]. However, very asymmetric kernels such as damped sinusoids [3] are not able to model suitably harmonic signals. 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 above, the approach is an iterative one. We will compare two variants of the technique. The first variant, which is non-adaptive, is roughly similar to the general approach used in [10], which we applied to the specific task of audio coding. However, the second adaptive variant is a novel one, which take advantage of the additional parameters of the gammachirp kernels and the inherent nonlinearity of the auditory pathway [6][7]. Some detail on each variant are given below.

2.1.1. Non-Adaptive Paradigm

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

\[ g(f_c, t) = (t)^3 e^{-2\pi bt} \cos(2\pi f_c t) \hspace{1cm} t > 0, \]  \hspace{1cm} (3)

where \(f_c\) is the center frequency of the filter, distributed on ERB (Equal Rectangular Bandwidth) 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. 2). The residual signal \(R_x(t)\) decreases at each step.

2.1.2. Adaptive Paradigm

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

\[ g(f_c, t, b, l, c) = t^{l-1} e^{-2\pi bt} \cos(2\pi f_c t + clnt) \hspace{1cm} t > 0. \]  \hspace{1cm} (4)

It has been shown that the gammachirp filters minimize the scale/time uncertainty [6]. 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 \(b\) 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 [5]. In our suboptimal technique, we use the same gammatone filters as the ones used in the non-adaptive paradigm with values of \(l\) and \(b\) given in [6]. 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.

\[ G_{max1} = \arg\max_{f, t_0} \{ |< r, g(f, t_0, b, l, c) > | \} \hspace{1cm} g \in G \]  \hspace{1cm} (5)

\[ G_{max2} = \arg\max_{f, t_0} \{ |< r, g(f, t_0, b, l, c) > | \} \hspace{1cm} g \in G - G_{max1}, \]  \hspace{1cm} (6)

where \(G\) is the set of all kernels, and \(G - G_{max1}\) excludes \(G_{max1}\) from the search space. For the sake of simplicity, we use \(f\) instead of \(f_c\) in Eqs. (5) to (9). 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 $G_{\text{max1}}$ and $G_{\text{max2}}$.

\[
G_{\text{maxc}} = \arg\max_{c} \{ |r, g(f, t_{0}, b, l, c) >| \} \\
g \in G_{\text{max1}} \cup G_{\text{max2}}.
\] (7)

We then use the information found in the second step to find the best $b$.

\[
G_{\text{maxb}} = \arg\max_{b} \{ |r, g(f, t_{0}, b, l, c) >| \} \\
g \in G_{\text{maxc}}
\] (8)

We finally find the best $l$ among $G_{\text{maxb}}$ found in the previous step.

\[
G_{\text{maxl}} = \arg\max_{l} \{ |r, g(f, t_{0}, b, l, c) >| \} \\
g \in G_{\text{maxb}}
\] (9)

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 helps us obtain better coding gains. The reason for that 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.

2.2. Masking

We use gammachirp functions to decompose audio signals. In order to reduce the number of spikes that in return will increase the coding efficiency, we have developed a tentative masking model to remove inaudible spikes. Since there is not much difference in the spectrum of the gammachirp and gammatone functions, we have used gammatone functions to develop the masking model. For on-frequency temporal masking, that is the temporal masking effects in each critical band (channel), we calculate the temporal forward and backward masking as follows. First we calculate the absolute threshold of hearing in each critical band. Since the basis functions are short, the absolute threshold of hearing has been elevated by 10 dB/decade when the duration of basis function is less than 200 msec [14].

\[
QT_{k} = AT_{k} + 10(\log_{10}(200) - \log_{10}(d_{k}))
\] (10)

where $AT_{k}$ is the absolute threshold of hearing for critical band $k$, $QT_{k}$ is the elevated threshold in quiet for the same critical band, and $d_{k}$ is the effective duration of the $k$th basis function defined as the time interval between the points on the temporal envelope of the the gammatone function where the amplitude drops by 90%. The masking sensation level is given by

\[
SL_{k}(i) = 10\log\left(\frac{a_{k}^{2}A_{k}^{2}}{QT_{k}}\right)
\] (11)

where $SL_{k}(i)$ is the sensation level of the $i$th spike in critical band $k$, $a_{k}(i)$ is the amplitude of the $i$th spike in critical band $k$, and $A_{k}$ is the peak value of the Fourier transform of the normalized gammatone function in critical band $k$. We set the initial level for the masking pattern in critical band $k$ to $QT_{k}$ and consider three situations for the masking pattern caused by a spike. When a maskee starts within the effective duration of the masker, the masking threshold is given by

\[
M_{k}(n_{i} : n_{i} + L_{k}) = \max(M_{k}(n_{i} : n_{i} + L_{k}), SL_{k}(i) - 20)
\] (12)

where $M_{k}$ is the masking pattern (in dB) in critical band $k$, $n_{i}$ is the start time index of the $i$th spike, and $L_{k}$ is the effective length of the gammatone function in critical band $k$ (defined as the effective duration $d_{k}$ of the gammatone function in critical band $k$ multiplied by the sampling frequency). Since gammatone functions are tonal-like signals, we assume that the masking level caused by a spike is 20 dB less than its sensation level. In order to avoid overmasking the spikes, we take the maximum of the masking threshold due to a spike and the threshold caused by other spikes in the same critical band at any time instance. We have also investigated adding up the masking thresholds caused by all spikes in the same critical band (in the linear domain) at any time instance. That approach would overmask the spikes and results in audible distortion.

Other situations are when a maskee starts after the effective duration of the masker (i.e., forward masking), and when a maskee starts before a masker (i.e., backward masking). For forward and backward masking, we assume a linear relation between the masking threshold (in dB) and the logarithm of the time delay between the masker and the maskee in msec [8]. Since the effective duration of forward
masking depends on the masker duration [13], we define an effective duration for forward masking in critical band $k$ as follows

$$F_{dk} = 100 \arctan(d_k)$$

(13)

The forward masking threshold is given by

$$FM_i(n) = (SL(i) - 20) \frac{\log_{10}(n_{i+L_k+1}^{n_{i+L_k+1}})}{\log_{10}(n_i^{n_i})}$$

(14)

where $n_i + L_k + 1 \leq n \leq n_i + L_k + FL_k$ and

$$FL_k = \text{round}(F_{dk}f_s),$$

(15)

$f_s$ denotes the sampling frequency. The index $i$ denotes the index of the spike and $k$ is the channel number. This forward masking contributes to the global masking pattern in critical band $k$ as follows

$$M_k = \max(n_i + L_k + 1 : n_i + L_k + FL_k)$$

(16)

For the backward masking, we assume 5 msec as the effective duration of masking for all critical bands regardless of the effective duration of gammatone functions. Hence, the backward masking is given by

$$BM_i(n) = (SL(i) - 20) \frac{\log_{10}(n_{i-0.005f_s}^{n_i})}{\log_{10}(n_{i-0.005f_s}^{n_i})}$$

(17)

Similar to the forward masking effect, the backward masking affects the global masking pattern in critical band $k$ as follows

$$M_k = \max(M_k(n_i - 0.005f_s : n_i - 1), BM_i)$$

(18)

For off-frequency masking effects, we have considered the masking effects caused by any spike in two adjacent critical bands. According to [12] a single masker produces an asymmetric linear masking pattern in the Bark domain, with a slope of -27 dB/Bark for the lower frequency side and a level-dependent slope for the upper frequency side. The slope for the upper frequency side is given by

$$s_u = -24 - \frac{230}{f} + 0.2L$$

(19)

where $f = f_c$ is the masker frequency (gammatone center frequency in our work) in Hertz and $L$ is the masker level in dB. We have used this approach to calculate the masking effects caused by each spike in the two immediate neighboring critical bands. The result of this approach was insignificant, which indicates a need for an effective masking model for off-frequency masking in spike coding.

Note that masking models used in most audio coding systems do not perform well on spike coding systems. The reason may be that in this coding paradigm, spikes are well localized in both time and frequency. Removing of any audible spike would produce musical noise that cannot be tolerated in high quality audio coding.

### 2.3. Coding

We pointed out earlier that sparse codes generate peaky histograms suitable for entropy coding. Therefore, we used arithmetic coding to allocate bits to these quantities.

Time-differential coding is used to further reduce the bit rate. More robust and efficient differential coding schemes such as the Minimum Spanning Tree (MST) are under investigation. Preliminary results give a gain of 50% on bit rate over the simple time-differential coding used in this article.

### 3. GENERATION OF SPIKEGRAMS FOR DIFFERENT SOUND CLASSES

We tested the algorithm for four different sounds: percussion, speech, castanet, white noise. In the next few sections, we will give results obtained on these different kinds of sound.

#### 3.1. Coding of Percussion

In this experiment, we code the percussion signal shown in Fig. 1 using the two variants described in the previous section: adaptive and non-adaptive approaches.

#### 3.1.1. Non-Adaptive Scheme

Non-adaptive scheme generates 30000 spikes. The matching pursuit is done for 30000 iterations to generate 30000 spikes.

Fig. 2 shows the spikegram generated by the non-adaptive method. As we can see, the onsets and
offsets of the percussion are detected clearly by the algorithm. There are 30000 spikes in the code (for 80000 samples of the original sound file) before temporal masking is applied.

![Fig. 1: Samples of a percussion sound.](image1)

We then used the masking technique detailed in section 2.2. The number of spikes after temporal masking is 29370. Note that spike coding gain in this case is $0.37N$ ($N$ is the number of samples in the original signal).

Two parameters are important for each spike: its position or spiking time and its amplitude. For now, we are using a lossless compression to encode these two parameters. We first extracted the histogram of the values (which seems to be very peaked). Therefore arithmetic coding is used for the compression of these values. For spike timing a differential paradigm is used. Times are first sorted in increasing order: Only the time elapsed since the last sorted spike is stored. This trick reduces the dynamic range of spike timings and makes it possible to perform arithmetic coding on timing information as well. We also used arithmetic coding for the compression of center frequencies. Using arithmetic coding, we used 135330 bits to code the spiking amplitudes and 51930 bits to code the timing information. For center frequencies, we used 45440 bits. These considerations give us a total bit rate of 2.90 bits/sample.

3.1.2. Adaptive Scheme
In this scheme the gammatone filters are used as described in the previous section. Fig. 3 shows the decrease of residual error through the number of iterations for the adaptive and non-adaptive approaches.

Table 1 gives results and a comparison of the two different schemes. The number of spikes for the non-adaptive scheme before masking for the same residual energy is 44% percent more than the number of spikes for the adaptive scheme. Note that the spike gain is $0.12N$.

3.2. Coding of Speech
The same two techniques are applied to speech coding. The speech signal used is the utterance “I’ll willingly marry Marylin”.

3.2.1. Non-Adaptive Scheme
The spikegram contains 56000 spikes before temporal masking. The number of spikes was reduced to 35208 after masking. Therefore, the spike coding gain is $0.44N$ ($N$ is signal length).

We used arithmetic coding to compress spike amplitudes and differential timing (time elapsed between consecutive spikes). Results are given in Table 2. The overall coding rate is 3.07 bits/sample.

2The amplitude can be seen as the synaptic strength with which each neuron is connected to another neuron.
<table>
<thead>
<tr>
<th></th>
<th>Adaptive (24 Channels)</th>
<th>Non-Adaptive (24 Channels)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spikes before masking</td>
<td>10000</td>
<td>30000</td>
</tr>
<tr>
<td>Spikes after masking</td>
<td>9430</td>
<td>29370</td>
</tr>
<tr>
<td>Spike gain</td>
<td>0.12N</td>
<td>0.37N</td>
</tr>
<tr>
<td>Bits for channel coding</td>
<td>30620</td>
<td>45440</td>
</tr>
<tr>
<td>Bits for amplitude coding</td>
<td>37430</td>
<td>135350</td>
</tr>
<tr>
<td>Bits for time coding</td>
<td>30250</td>
<td>51930</td>
</tr>
<tr>
<td>Bits for chirp factor coding</td>
<td>9940</td>
<td>0</td>
</tr>
<tr>
<td>Bits for coding b</td>
<td>21350</td>
<td>0</td>
</tr>
<tr>
<td>Bits for coding l</td>
<td>25500</td>
<td>0</td>
</tr>
<tr>
<td>Total bits</td>
<td>155090</td>
<td>232720</td>
</tr>
<tr>
<td>Bit rate (bit/sample)</td>
<td>1.93</td>
<td>2.90</td>
</tr>
</tbody>
</table>

Table 1: Comparative results for the coding of percussion (80000 samples) at high quality (scores above 4 on the ITU-R 5-grade impairment scale in informal listening tests) for the adaptive and non-adaptive schemes.

<table>
<thead>
<tr>
<th></th>
<th>Adaptive (24 Channels)</th>
<th>Non-Adaptive (24 Channels)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spikes before masking</td>
<td>12000</td>
<td>56000</td>
</tr>
<tr>
<td>Spikes after masking</td>
<td>10492</td>
<td>35208</td>
</tr>
<tr>
<td>Spike gain</td>
<td>0.13N</td>
<td>0.44N</td>
</tr>
<tr>
<td>Bits for channel coding</td>
<td>40960</td>
<td>118536</td>
</tr>
<tr>
<td>Bits for amplitude coding</td>
<td>35432</td>
<td>67048</td>
</tr>
<tr>
<td>Bits for time coding</td>
<td>40190</td>
<td>60376</td>
</tr>
<tr>
<td>Bits for chirp factor coding</td>
<td>9836</td>
<td>0</td>
</tr>
<tr>
<td>Bits for coding b</td>
<td>15260</td>
<td>0</td>
</tr>
<tr>
<td>Bits for coding l</td>
<td>16000</td>
<td>0</td>
</tr>
<tr>
<td>Total bits</td>
<td>157676</td>
<td>245960</td>
</tr>
<tr>
<td>Bit rate (bits/sample)</td>
<td>1.98</td>
<td>3.07</td>
</tr>
</tbody>
</table>

Table 2: Comparative results for the coding of speech (80000 samples) at high quality (scores above 4 on the ITU-R 5-grade impairment scale in informal listening tests) for the adaptive and non-adaptive schemes.

<table>
<thead>
<tr>
<th></th>
<th>Adaptive (24 Channels)</th>
<th>Non-Adaptive (24 Channels)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spikes before masking</td>
<td>7000</td>
<td>30000</td>
</tr>
<tr>
<td>Spikes after masking</td>
<td>6510</td>
<td>24580</td>
</tr>
<tr>
<td>Spike gain</td>
<td>0.08N</td>
<td>0.30N</td>
</tr>
<tr>
<td>Bits for channel coding</td>
<td>22930</td>
<td>85000</td>
</tr>
<tr>
<td>Bits for amplitude coding</td>
<td>39450</td>
<td>83810</td>
</tr>
<tr>
<td>Bits for time coding</td>
<td>33000</td>
<td>73540</td>
</tr>
<tr>
<td>Bits for chirp factor</td>
<td>7780</td>
<td>0</td>
</tr>
<tr>
<td>Bits for coding b</td>
<td>13900</td>
<td>0</td>
</tr>
<tr>
<td>Bits for coding l</td>
<td>6510</td>
<td>0</td>
</tr>
<tr>
<td>Total bits</td>
<td>123570</td>
<td>242350</td>
</tr>
<tr>
<td>Bit rate (bit/sample)</td>
<td>1.54</td>
<td>3.03</td>
</tr>
</tbody>
</table>

Table 3: Comparative results for the coding of castanet (80000 samples) at high quality (scores above 4 on the ITU-R 5-grade impairment scale in informal listening tests) for the adaptive and non-adaptive schemes.
3.2.2. Adaptive Scheme

Figs. 4 and 5 show that, in the case of speech, using the adaptive scheme can reduce both the number of spikes and the number of cochlear channels (filterbank channels) drastically.

To achieve the same quality, we need 12000 spikes (compared to 56000 spikes for the non-adaptive case). The number of spikes after masking is 10492 spikes. The spike coding gain is 0.13*N (the gain is 0.44*N in the non-adaptive case). Results are given in Table 2. The overall required bit rate is 1.98 bits/sample in this case (roughly 35 percent lower than in the non-adaptive case).

3.3. Coding of Castanet

We used the adaptive coding algorithm, and obtained an ITU-R impairment scale score of 4 in informal listening tests. The number of spikes before temporal masking is 7000. We applied the temporal masking and reduced the number of spikes to 6510.

Spike coding gain is 0.08*N in the adaptive case and 0.30*N in the non-adaptive case. The bit rate is 1.54 bits/sample in the adaptive case and 3.03 bits/sample. Results for both the adaptive and non-adaptive schemes are given in Table 3.

3.4. Coding of White Noise

In another set of experiments, we modelled white noise by the adaptive and non-adaptive approaches and compared the results. As we can see in Fig 6, as for other signal types, the adaptive paradigm outperforms the non-adaptive one. Note that our deterministic model has been able to model the stochastic white noise.

3.5. Discussion

Our technique generated spike gains ranging from 0.08*N to 0.12*N. This is much lower than 1.26*N obtained in [1], 3.2*N in [9], and 0.66*N obtained in [2] for signals sampled at 4 to 8 kHz. The latter-mentioned techniques are based on thresholding the outputs of a filterbank and generating spikes each time the threshold is crossed. This peak-picking approach generates redundant spikes and higher number of spikes for the same audio materials compared to the proposed technique.

4. FUTURE WORK

The matching pursuit algorithm is a relatively slow algorithm. We have derived a closed-form formula for the correlation between gammatone and gammachirp filters that can be used to speed up the process.

The dynamics (evolution through time) of spike amplitudes, channel frequencies, etc. through time can give us some good hints on how these values must be coded. Preliminary results on this issue have given very encouraging performance.
The introduction of perceptual criteria and weak/weighted matching pursuit is another potential performance booster to be investigated.

In this article, we used time differential coding to code spikes. A more efficient way would be to consider spikes as graph nodes and optimize coding cost through different paths. This approach is under investigation.

Some preliminary quantization sensitivity analysis is under investigation.

The representation we have proposed in this article generates independent acoustical events on a coarse time scale. However, on a finer time scale, each acoustical event consists of dependent and correlated elements (spikes). This dependency can be used to further reduce redundancy.

The masking paradigm used in this article works better for low-frequency contents than high-frequency information. A modified version of the approach should put more emphasis on higher frequencies.

5. CONCLUSION

We have proposed a new biologically-inspired paradigm for universal audio coding based on neural spikes. Our proposed approach is based on the generation of sparse 2-D representations of audio signals, dubbed as spikegrams by matching pursuit. A masking model is applied to the spikegrams to remove inaudible spikes and to increase the coding efficiency.

We have replaced the peak-picking approach used in [2] and [1] by the matching pursuit, which is much more efficient in terms of dimensionality reduction. We have also proposed the adaptive fitting of the chirp, decay, and attack factors in the gammachirp filterbank. This change has reduced both the computational load and the bit rate of the coding system. We further introduced additional masking to the output of the representation obtained by matching pursuit. Arithmetic coding is used for lossless compression of the spike parameters to further reduce the bit rate.

6. REFERENCES


