FIRE TEXTURE SOUND RE-SYNTHESIS USING SPARSE DECOMPOSITION AND NOISE MODELLING

Stefan Kersten[1] *

[1]Music Technology Group
Universitat Pompeu Fabra
Barcelona, Spain
stefan.kersten@upf.edu

Hendrik Purwins[1,2,3] †

[2]Neurotechnology Group
Berlin Institute of Technology, Berlin, Germany
[3]PMC Technologies Münster, Germany
hpurwins@gmail.com

ABSTRACT

In this paper we introduce a framework that represents environmental texture sounds as a linear superposition of independent foreground and background layers that roughly correspond to entities in the physical production of the sound. Sound samples are decomposed into a sparse representation with the matching pursuit algorithm and a dictionary of Daubechies wavelet atoms. An agglomerative clustering procedure groups atoms into short transient molecules. A foreground layer is generated by sampling these sound molecules from a distribution, whose parameters are estimated from the input sample. The residual signal is modelled by an LPC-based source-filter model, synthesizing the background sound layer. The capability of the system is demonstrated with a set of fire sounds.

1. INTRODUCTION

Many sounds in our daily surroundings have textural properties—yet *sound texture* is a term difficult to define, because these sounds are often perceived subconsciously and in a context-dependent way. Sound textures exhibit some of the statistical properties that are normally attributed to noise, but they arguably do convey information; not so much in an information theoretic sense, but rather as a carrier of emotional and situational percepts [14]. Indeed, sound texture—often denoted *atmosphere*—forms an important part of the sound scene in real life, in movies, games and virtual environments.

Current sound texture synthesis models don't normally take the physical and perceptual characteristics of a specific source sound type into account. Concatenative or granular methods, such as the ones in [7, 13, 16] capture the characteristics of the source material by segmenting the original sound into small segments and reassembling them according to a statistics either estimated from the source or based on heuristics. Other models borrow from related fields in signal processing, for example by extending an LPC-based source-filter model with a model of the residual's temporal variations [1, 18] or by learning coefficient sequences in wavelet domain representations [5]. Parametric statistical models promise insight into possible dependencies between a sound's wavelet coefficients, but can suffer from large parameter spaces and overfitting when applied to synthesis [8].

In [6], the author presents a physically inspired synthesis model for fire sounds that incorporates the following three main components: *lapping*, "combustion of gases in the air", *crackling*, "small scale explosions caused by stresses in the fuel" and *hissing*, "regular outgassing, release of trapped vapor" ([6], p. 412). In our work we intend to capture these characteristic elements of fire sounds by modelling them individually. In particular, we model the *crackling* component as a foreground layer represented by atoms in a sparse decomposition and the *lapping* and *hissing* components as a background layer, represented by the linear prediction coded residual of the sparse decomposition. The motivation behind this segmented model is the hope to lay the foundation for independent manipulation of the different layers during synthesis, thereby obtaining meaningful parameterisation for the synthesis model.

In previous work [9] we have modelled water stream sounds by first decomposing them in an overcomplete sparse representation by using the matching pursuit algorithm [12] and a dictionary of Gammatone atoms [11]. The atomic representation is intended to represent the *bubble* component of the water stream sounds and is statistically modelled by estimating a smoothed histogram of atom inter-onset intervals. The residual, representing mostly uncorrelated water noise is modelled by estimating the statistics of filter coefficients in the cascade time-frequency linear prediction (CTFLP) framework [1].

In this paper we extend this framework to fire sounds; in order to take into account the cross-atom correlations during the sharp transients usually found in fire sounds ([6], p. 412), we employ a Daubechies wavelet dictionary for obtaining the sparse decomposition matrix, which in a second step is subject to an agglomerative clustering procedure that groups atoms close in time or frequency into *molecules*. These groups of atoms are then treated as individual *crackling* events in the statistical modelling component. The sparse decomposition residual is assumed to contain mostly coloured noise from the *lapping* and and *hissing* components and is modelled with the CTFLP method mentioned above.

The rest of this paper is organised in the following way: In section 2 we introduce the signal representation and estimation methods that constitute our modelling framework; in 3 we present some example sounds, the parameters used for modelling their characteristics and synthesis results; in 4 we conclude our findings and provide an outlook on future work.

* The first author (S. K.) was supported by the FI-DGR 2010 scholarship of the Generalitat de Catalunya and a DTIC grant from the Universitat Pompeu Fabra.

† The second author (H. P.) was supported in part by the German Bundesministerium für Forschung und Technologie (BMBF), Grant No. Fkz 01GQ0850.

2. METHOD

2.1. Sparse Decomposition via Matching Pursuit

Following [15], a sparse decomposition of a sampled sound signal $x[t]$ is a linear combination of N amplitudes s_n and sound atoms $\phi_n[t]$:

$$x[t] = \hat{x}[t] + \epsilon[t] = \sum_{n=1}^{N} s_n \phi_n[t] + \epsilon[t], \quad (1)$$

where $\epsilon[t]$ represents the residual. Each atom $\phi_n[t]$ is a temporally shifted (by τ_n) version of one of the J atomic prototypes $\psi_j[t]$:

$$\phi_n[t] = \psi_{j_n}(t - \tau_n), \quad (2)$$

Matching Pursuit (MP) [12] is an iterative greedy method that can be used to obtain the decomposition in Equations 1 and 2. In each iteration, the atomic functions of the dictionary are correlated with the signal and the atomic function with the highest correlation is subtracted, yielding a residual signal. This process is repeated with the residual until a stopping criterion, in our case a predefined signal to residual ratio, is reached.

2.1.1. Daubechies Dictionary

The discrete wavelet transform decomposes a signal $x(t)$ into atoms of shifted and dilated bandpass wavelet functions $\psi(t)$ and shifted versions of a lowpass scaling function $\psi_0(t)$, i.e. the signal is represented on multiple time scales K and frequency scales J.

In our work we don't employ the time dilation structure of the wavelet transform as in [4], but we use an overcomplete dictionary of J wavelet bandpass functions $\psi_j(t)$, that were generated by the inverse discrete wavelet transform of a wavelet tree with a single impulse on each of the scales respectively:

$$\psi_j(t) \equiv 2^{-j/2} \psi(2^{-j} t) \quad (3)$$

The filters approximately span the whole audible frequency range and have a coupling between the length of their support in the time and frequency domains, effectively providing low frequency narrow band filters with long support and high frequency broad band filters with short support along an octave frequency scale.

Following our work in [8], where a Daubechies wavelet base provided a good representation for fire sounds, we chose an overcomplete dictionary of Daubechies wavelets functions with ten vanishing moments [3], evaluated at $J = 9$ scales with dilations corresponding to all possible shifts of the function (see Section 2.1).

2.2. Sound Molecules via Agglomerative Clustering

Let $x[t]$ be a sound decomposed according to Equation 1. Following [17], for a distance threshold θ we build the upper-triangular adjacency matrix $\mathbf{A}$, defined by:

$$a_{nm} = \begin{cases} 1, & d(\phi_n, \phi_m) \leq \theta, \\ 0 & \text{else.} \end{cases} \quad (4)$$

where $1 \leq j, n \leq N, n \leq m \leq N$ and $d(\phi_n, \phi_m)$, the distance measure based on temporal/spectral centroid/spread of the two atoms (7).

$$d(\phi_n, \phi_m) = \sqrt{w_t d_t(\phi_n, \phi_m)^2 + w_s d_s(\phi_n, \phi_m)^2} \quad (5)$$

$$d_t(\phi_n, \phi_m) = \frac{t_c(\phi_n) - t_c(\phi_m)}{t_s(\phi_n) + t_s(\phi_m)} \quad (6)$$

$$d_s(\phi_n, \phi_m) = \frac{s_c(\phi_n) - s_c(\phi_m)}{s_s(\phi_n) + s_s(\phi_m)} \quad (7)$$

Here $t_c(\phi)$, $t_s(\phi)$ are the temporal centroid and spread of atom ϕ and $s_c(\phi)$, $s_s(\phi)$ are the spectral centroid and spread. By using the distance measure d instead of correlation as used in [17] we can have non-zero distances for two close but short non-overlapping atoms, even though the dictionary is not divided into transient and tonal atoms. The coefficients w_t and w_s weigh the relative contributions of the temporal and spectral components, respectively, to the distance measure. They are model parameters that have to be tuned for each sound in order to obtain the desired shape and number of molecules.

All atoms that are pairwise sufficiently close with respect to their temporal and spectral centroid are collected to form a sound molecule. For this objective, the molecule atoms are weighted by their coefficients s_n. Starting from the first row in A, for all non-zero a_{nm} the corresponding coefficient/atom pair is added. Then, the algorithm looks for the next row sharing non-zero entries with the first row and adding the atom/coefficient pairs that are not yet part of the sum. This is iteratively continued until no more new atoms can be found.

2.3. Cascade Time Frequency Linear Prediction

Cascade time frequency linear prediction (CTFLP) is a combination of linear predictive coding (LPC) and frequency domain linear prediction (FDLP) that has been used for coding textural sounds by [1] and [18]. The intention is to capture both the spectral envelope characteristics of the source signal by conventional LPC and the envelope of the temporal fine structure by applying linear predictive coding to the LPC residual in the frequency domain.

The signal is first divided into overlapping frames with frame size N and hopsize H. Each frame is multiplied by a smoothing window and encoded by the LPC to obtain coefficients for an IIR filter that approximates the spectral envelope of the signal within the frame. After whitening the signal by applying the inverse envelope and the inverse window, the residual is transformed to the frequency domain with the discrete cosine transform. The frequency domain coefficients are subject to another LPC step that yields filter coefficients for a filter that approximates the square of the Hilbert envelope of the frame's temporal structure. We also estimate the residual energy after applying CTFLP and store it as a single coefficient.

The inverse procedure starts by generating a sample of white noise for each frame, transforming to the frequency domain with the DCT, imposing the temporal envelope filter, transforming back to the time domain with the inverse DCT, windowing and imposing the spectral envelope filter.

2.4. Foreground event density estimation

For modelling the foreground layer we assume that the constituent events are produced by a Poisson process, i.e. that the distribution of inter-event intervals –or conversely, the *event density* per time interval– is independent of all other events and, in our case, stationary. We estimate the inter-event distribution by applying kernel density estimation to the time intervals measured from the

	SNR	t	w_t	w_s
fire1	18	40	1	1
fire2	6	600	1	1
fire3	18	200	1	1
fire4	15	200	1	2
fire5	18	250	1	4
fire6	6	200	1	1

Table 1: Sparse decomposition and agglomerative clustering parameters for each of the six fire sounds: Signal-to-noise ratio SNR in dB, distance function threshold t, temporal distance weight w_t and spectral distance weight w_s.

events built in the molecule agglomeration step from Section 2.2. Note that for estimating inter-event intervals we measure the distance between two events' temporal centroids, not their onsets.

3. EXAMPLES

For our experiments we chose six samples of fire sounds. We extracted the first 10 seconds of the left channel of each sound in order to limit processing time. All sounds were sampled at their original sampling frequency of 44.1kHz.

Each sound was then decomposed into an atomic representation using the Matching Pursuit Toolkit [10], an efficient implementation of the Matching Pursuit algorithm described in Section 2.1. We used a dictionary of $J = c_n = 9$ Daubechies wavelet bandpass filters with ten vanishing moments that were generated as described in Section 2.1.1. We performed the sparse decomposition for each sound individually until a 18 dB ratio between the atomic part and the residual was reached. We also recorded, during the decomposition process, the signal to noise ratio (SNR) associated with each matching pursuit iteration, in order to be able to adapt the actual SNR used when estimating the synthesis model parameters to each sound separately.

The atoms obtained in the decomposition step were then grouped into higher level molecules using the agglomerative clustering algorithm described in Section 2.2. The goal here is to transform the atomic sparse representation into a "molecular" sparse representation, were molecules correspond to individual foreground events in the source sound. Table 1 lists the parameters used for decomposition and clustering, i.e. the signal to noise ratio of the decomposition, the distance function threshold that determines whether two atoms are considered "close" according to their distance and the weights for the temporal and spectral components respectively of the distance function.

For resynthesis, the resulting molecules were treated as individual events and a smoothed histogram of the inter-event intervals was estimated by kernel density estimation [2]. The generation algorithm then simply draws a molecule uniformly from the set of all molecules, reconstructs it at the current point in time, draws a delta time δt from the inter-event interval distribution, updates the current time by δt and proceeds until a maximum output sound duration has been reached[1].

[1]Note that although in our experiments all processing was performed offline, the resynthesis process is causal and readily suitable for realtime implementation.

N	H	N_s	N_t	$w(n)$
1024	256	20	20	$0.5\left(1-\cos\frac{2\pi n}{N-1}\right)$

Table 2: CTFLP parameters used for encoding the residual: Frame size N, hop size H, number of spectral envelope coefficients N_s, number of temporal envelope coefficients N_t a and the window function $w(n)$ (Hann).

The residual, in the ideal case containing only coloured background noise, is first coded by the CTFLP process described in section 2.3, yielding a total of 41 filter coefficients per frame (20 for describing the spectral envelope, 20 for the temporal envelope and one for the residual noise energy). Table 2 lists the parameters used in the encoding process.

During resynthesis new CTFLP frames were drawn independently from the set of all analysed frames, concatenated and converted to the time domain by the inverse CTFLP. Both synthesized signals, foreground events and background noise, were then superimposed to obtain the final synthesized result. All of the sounds are available online[2] in four versions: The original sound, the synthesized foreground (*_fg.wav*) and background (*_bg.wav*) individually and the synthesized mixture of foreground and background (*_fg+bg.wav*).

Since one of the objectives in our work is to agglomerate atoms from a sparse representation into higher level molecules that correspond to perceptual foreground events. The signal dependent distance function threshold and the relative weights of the temporal and spectral distance function components are of crucial importance. While in these experiments the parameters have been tuned by trial and error, heuristically setting the threshold close or slightly above the mean distance of all the atoms in the representation led to acceptable results.

Figure 1 shows three molecules from three different sounds: The first, *fire3*, contains relatively isolated events that can also be identified in the molecules built. The second, *fire6*, contains mostly low-frequency rumble and the molecules span the maximum length allowed by the agglomeration process. In the third example the molecules span multiple foreground events, which indicates that the relative weights w_t and w_s have not been optimal for this sound.

All of the synthesized sounds (except the one for *fire1*) exhibit a certain smearing of the sharp transients which can be attributed to the shape of the Daubechies wavelet filters we employed, which don't seem to be able to capture the full transient content.

The resynthesized residuals exhibit mainly two artefacts: The smearing of transients and bursts of noise that are not synchronised with the foreground events and let the resynthesis appear "noisier" than the original. The first can be explained by the analysis window frame and hop size for the CTFLP coding, which determines the tradeoff between uncertainty in the temporal and the spectral envelope.

Both artefacts are related to the problem of determining the decomposition depth, i.e. the SNR threshold at which to stop sparse coding and start residual coding. Some of the sounds, in particular *fire3*, contain explosion tails that follow the transient impulses but are not captured by the sparse model. Consequently there is

[2]`http://tinyurl.com/cv4unp4`

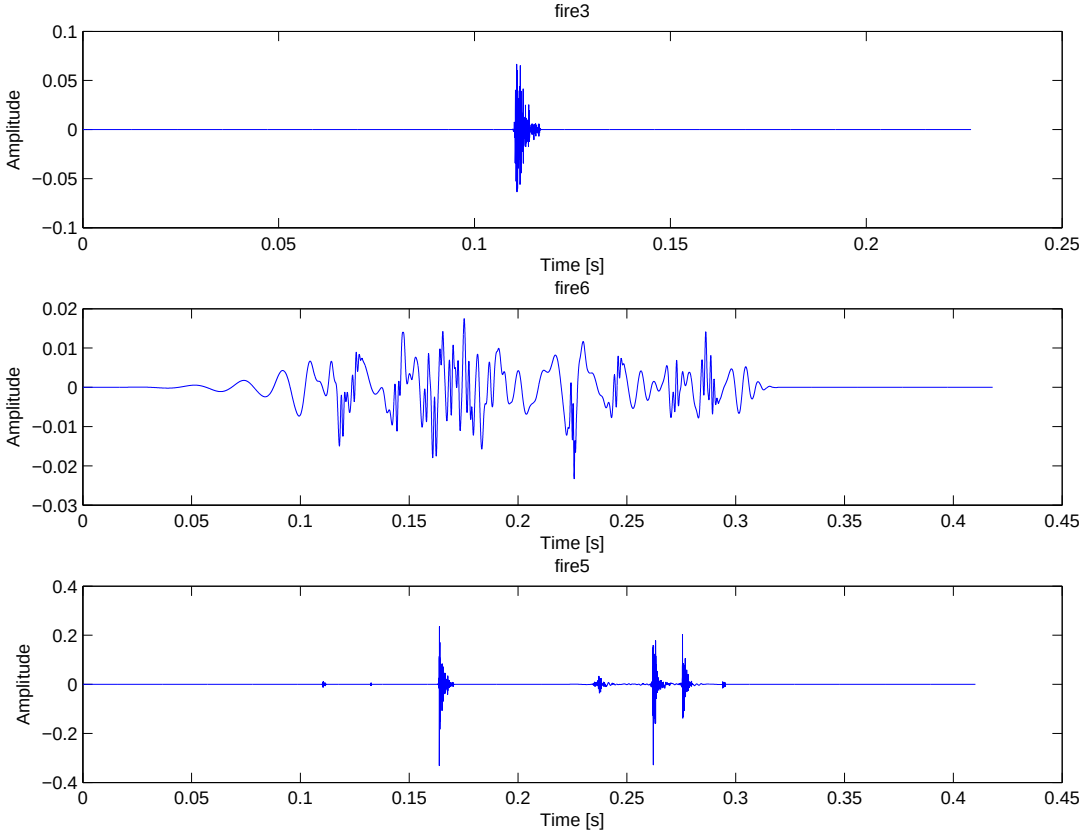

Figure 1: Molecules found by the agglomerative clustering process from three different sounds, *fire3*, *fire6* and *fire5*.

to be expected a high correlation between foreground events and background residual that would need to be taken into account for synthesis, e.g. by making the residual frame distribution dependent on the energy of the foreground events. Another aspect is that our current model assumes that each CTFLP is independent of its predecessors in time, i.e. temporal correlations between successive residual frames are not captured by the statistical model. In order to adapt the sparse decomposition threshold to the ability of the background model to encode the residual, the threshold should not be a fixed SNR, but rather depend on the properties of the residual noise after CTFLP coding, e.g. the flatness of its power spectrum.

4. CONCLUSIONS

In the present work our motivation has been to decompose natural texture sounds into perceptually meaningful elements that can be manipulated separately during synthesis in order yield a variety of sounds from a single model. We have cast the objective into a framework that first decomposes a sound into atoms and residual and applies different resynthesis strategies to both parts.

While the foreground event extraction by agglomerative clustering works very well for sounds where foreground and background are clearly separated by the sparse decomposition, it fails to separate sharp transients from background noise when both are mixed in the sparse representation.

Our next steps in this line of research will be to formulate procedures that optimise some of the analysis parameters in a signal dependent way. The sparse decomposition threshold should be set in accordance with the ability of the background model to represent the residual, and the molecule clustering algorithm could be extended by placing a prior on the shape and density of the molecules that are to be expected in a given sound.

We also started to apply the framework to the water stream sounds from [9], where we hope to improve the synthesis quality for those sounds that are not modelled well by independent bandpass responses.

Finally, in future work we intend to develop meaningful transformations that employ the multi-level representation framework, for example by modifying the inter-event interval distribution in order to create fire textures of varying density.

5. ACKNOWLEDGMENTS

Many thanks to the anonymous reviewers and their invaluable comments.

6. REFERENCES

[1] M. Athineos and D. Ellis. Sound texture modelling with linear prediction in both time and frequency domains. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*, volume 5, pages V–648–51 vol.5, 2003.

[2] C. M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, New York, NY, USA, 2006.

[3] I. Daubechies. Orthonormal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 41(7):909–996, 1988.

[4] L. Daudet. Sparse and structured decompositions of signals with the molecular matching pursuit. *Audio, Speech, and Language Processing, IEEE Transactions on*, 14(5):1808–1816, 2006.

[5] S. Dubnov, Z. Bar-Joseph, R. El-Yaniv, D. Lischinski, and M. Werman. Synthesizing sound textures through wavelet tree learning. *Computer Graphics and Applications, IEEE*, 22(4):38–48, July 2002.

[6] A. Farnell. *Designing Sound*. Oct. 2010.

[7] R. Hoskinson. *Manipulation and Resynthesis of Environmental Sounds with Natural Wavelet Grains*. PhD thesis, The University of British Columbia, 2002.

[8] S. Kersten and H. Purwins. Hybrid sparse models of water stream texture sounds, Sept. DAFX-11, Tonophonie Workshop.

[9] S. Kersten and H. Purwins. Sound texture synthesis with hidden markov tree models in the wavelet domain. In *Sound and Music Computing Conference*, 2010.

[10] S. Krstulovic and R. Gribonval. Mptk: Matching pursuit made tractable. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 3, page III, 2006.

[11] R. Lyon, A. Katsiamis, and E. Drakakis. History and future of auditory filter models. In *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, pages 3809 –3812, June 2010.

[12] S. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *Signal Processing, IEEE Transactions on*, 41(12):3397–3415, 1993.

[13] J. Parker and B. Behm. Creating audio textures by example: tiling and stitching. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '04)*, volume 4, pages iv–317–iv–320, 2004.

[14] R. M. Schafer. *The Soundscape: Our Sonic Environment and the Tuning of the World*. Destiny Books, 1994.

[15] S. Scholler and H. Purwins. Sparse approximations for drum sound classification. *Selected Topics in Signal Processing, IEEE Journal of*, 5(5):933 –940, Sept. 2011.

[16] D. Schwarz and N. Schnell. Descriptor-Based sound texture sampling. In *Proceedings of SMC Conference 2010*, Barcelona, Spain, 2010.

[17] B. Sturm, J. Shynk, and S. Gauglitz. Agglomerative clustering in sparse atomic decompositions of audio signals. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pages 97–100, 2008.

[18] X. Zhu and L. Wyse. Sound texture modeling and timefrequency LPC. In *Proc. of the 7th Int. Conf. on Digital Audio Effects (DAFX-04)*, Naples, Italy, Oct. 2004.