SOUND TEXTURE SYNTHESIS WITH HIDDEN MARKOV TREE MODELS IN THE WAVELET DOMAIN

Stefan Kersten, Hendrik Purwins
Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain
{stefan.kersten,hendrik.purwins}@upf.edu

ABSTRACT

In this paper we describe a new parametric model for synthesizing environmental sound textures, such as running water, rain, and fire. Sound texture analysis is cast in the framework of wavelet decomposition and multiresolution statistical models, that have previously found application in image texture analysis and synthesis. We stochastically sample from a model that exploits sparsity of wavelet coefficients and their dependencies across scales. By reconstructing a time-domain signal from the sampled wavelet trees, we can synthesize distinct but perceptually similar versions of a sound. In informal listening comparisons our models are shown to capture key features of certain classes of texture sounds, while offering the flexibility of a parametric framework for sound texture synthesis.

1. INTRODUCTION

Many sounds in our 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 [1]. Indeed, sound texture—often denoted *atmosphere*—forms an important part of the sound scene in real life, in movies, games and virtual environments.

In this work our goal is to synthesize environmental sounds with textural properties, such as running water, waves, fire, crowd noises, etc. Eventually, we intend to provide a building block for an application that automatically generates soundscapes for virtual environments. Our work is in the context of stochastic sound synthesis: given a *textural* analysis or target sound with statistical characteristics sufficiently close to stationarity, we want to synthesize stochastic variations that are perceptually close in their characteristics to the original but are not mere reproductions. In a *data-driven* approach we build a model by statistical signal analysis. The distributions captured by the model are then used to synthesize perceptually similar sounds by stochastic sampling.

Previous research suggests that textural sounds are perceived by human listeners in terms of the statistical properties of constituent features, rather than by individual acoustic events [2, 3]. The ability to model texture in a statistical sense, without detailed knowledge or assumptions about the structure of the source material, leads to several desirable properties that a texture model should possess:

- **Compactness of representation**: The model should require significantly less parameters than the original coded audio.
- **Statistical properties**: The signal statistics should be discoverable using a limited amount of training data.

In general a texture model for synthesis can be split in an analysis part and the actual synthesis part. The goal of the analysis phase is to estimate the joint statistics of signal coefficients in some decomposition space and combine them in a parametric or non-parametric model by statistical analysis. For audio signals, we typically need to estimate not only the *vertical* coefficient relationships, i.e. their interdependencies across the frequency axis, but also their *horizontal* dependencies across time. During the synthesis phase, a new time series of decomposition coefficients is generated by stochastic sampling from the model. If our model sufficiently captured the structural coefficient dependencies, then after transforming the sampled coefficients to the time domain, we obtain a signal that perceptually resembles the original but is not exactly the same.

Multiresolution (MR) signal analysis methods, and in particular the discrete wavelet transform, have been shown to be well suited for modeling the dynamics of sound textures, where important perceptual details are present in various frequency bands and on different time scales [4, 5, 6]. Even though the wavelet transform can be considered almost sparse for many natural signals [8], the coefficients retain inter- and intra-scale dependencies that have to be taken into account in a statistical decomposition and synthesis model. It has been shown that for natural signals like 2D images, the wavelet coefficients themselves are non-Gaussian, but approximately Gaussian conditioned on their context, i.e. neighboring coefficients in scale and location [7]. The hidden Markov tree (HMT) model [8] is a parametric statistical model, that captures inter-scale dependencies and is particularly suited to be applied to tree structured data like wavelet transform coefficients.

While previous approaches to sound texture synthesis have mostly been based on non-parametric density estima-

tion techniques—see Section 2 for an overview—the HMT has been successfully applied to a wide range of image processing problems, leaving room for speculation that it will also be applicable to sound texture modeling. It thus forms the basis of our approach to sound texture synthesis. Our model is similar to previous work in that we also use the wavelet transform for multiresolution signal analysis and perform density estimation in wavelet decomposition trees. Our estimator, however, instead of being based on nonparametric density estimation, explicitly casts the wavelet coefficient statistics and their interdependencies in a graphical model within the maximum likelihood framework. As with parametric models in general, when the modeling assumptions match the signals being modeled fairly well, we can gain from a principled probabilistic approach, e.g. by introducing priors, dealing with missing data and performing inference.

The rest of the paper is structured as follows: In Section 2 we give an overview of current approaches to sound texture modeling and multiresolution statistical analysis. In Section 3 we introduce the basic building blocks of our texture model, the discrete wavelet transform and the hidden Markov tree model and how these fit together in a synthesis model. In Section 4 we present results of natural sound textures synthesized from our model and finally draw some conclusions and mention possible future work in Section 5.

2. RELATED WORK

While image texture modeling has been under active investigation for at least 35 years, sound texture modeling has begun to find a similarly thorough treatment only relatively recently; for an overview with a focus on synthesis see [9].

Many approaches to sound texture modeling have been heavily inspired by methods originally developed for modeling texture images. In [4] the authors describe a nonparametric sound texture model that learns conditional probabilities along paths in a wavelet decomposition tree. Path probability densities are estimated first for inter-scale coefficient dependencies and in a second step for intra-scale predecessor probabilities. In a similar fashion, [5] estimate the sound texture statistics on wavelet tree coefficients by kernel density estimation and histogram sampling, inspired by the approach taken by Efros and Leung for image texture synthesis [10]. The authors report improved results compared to the ones obtained by [4], but didn't conduct a conclusive quantitative evaluation.

A large body of research is devoted to the field of multiresolution statistical models, and in particular MR Markov models, for a comprehensive overview see [11]. The hidden Markov tree model has been applied to a wide range of problems in image and signal processing, such as denoising [8, 12, 13, 14] and texture classification and synthesis [15].

3. METHODS

3.1 The discrete wavelet transform

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

$$\begin{aligned} \psi_{J,K}(t) &\equiv 2^{-J/2}\psi(2^{-J}t - K) \\ \phi_{J_0,K}(t) &\equiv 2^{-J_0/2}\phi(2^{-J_0}t - K) \\ J, K &\in \mathbb{Z} \end{aligned} \tag{1}$$

When designed with certain constraints, the wavelet and scaling functions form an orthonormal basis with the following signal representation [16]:

$$\begin{aligned} z(t) &= \sum_K u_K \phi_{J_0,K}(t) + \sum_{J=0}^{J_0} \sum_K w_{J,K}\psi_{J,K}(t) \\ u_K &= \int z(t)\phi^*_{J_0,K}(t)dt \\ w_{J,K} &= \int z(t)\psi^*_{J,K}(t)dt \end{aligned} \tag{2}$$

where $*$ denotes complex conjugation. u_K and $w_{J,K}$ are called *scaling* and *detail coefficients*, respectively. In (1) and (2), J specifies the scale or resolution of analysis – the smaller J, the higher the resolution. J_0 is the lowest level of resolution, where the analysis yields both detail coefficients and scaling coefficients. In the case of audio signals, K denotes the temporal support of analysis, i.e. the amount of time a wavelet $\psi(t)$ is shifted from its support at time zero. The *detail coefficient* $w_{J,K}$ measures the signal content at time $2^J K$ and frequency $2^{-J} f_0$, where f_0 is the wavelet's center frequency. The *approximation coefficient* u_K measures the local mean at time $2_{J_0}K$. Following [8] and in order to reduce notational overhead, we will adopt a simplified indexing scheme for basis functions of the decomposition and the resulting coefficients: instead of indexing by scale J and shift K, we will use a one-dimensional mapping $J, K \mapsto \mathbb{Z}$, where the indices $i \in \mathbb{Z}$ have a fixed but unspecified ordering.

In practice, the DWT can be implemented with a pyramidal filterbank algorithm, where the signal is recursively split into lowpass and highpass filter responses, that together form a quadrature mirror filter pair. Both responses are downsampled by two; the highpass response forms the detail coefficients, while the lowpass response is used for further recursive analysis until a maximum depth is reached.

Due to the recursive structure of the DWT and the shift and dilation relations based on powers of two, the decomposition can be represented as a *forest* (list) of binary trees, where each coefficient in scale J has two children in the next finer resolution scale. At the coarsest level of detail the signal is represented as pairs of detail and approximation coefficients, at which a binary tree of detail coefficients is rooted. The decomposition of the time-frequency

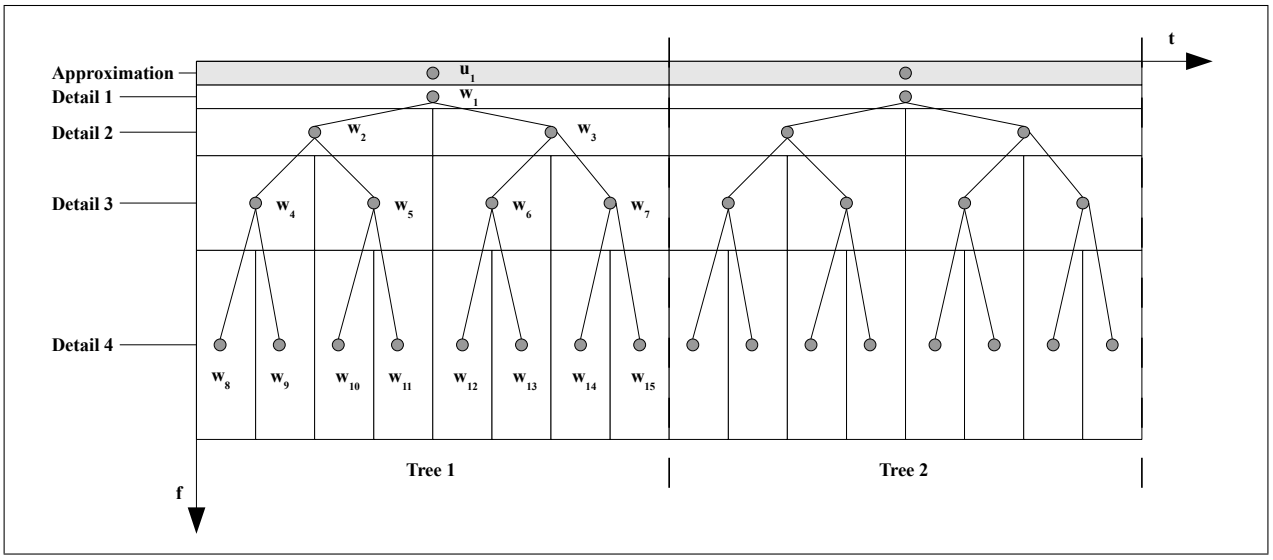


Figure 1: Four level wavelet time-frequency decomposition. Shown are two consecutive trees with frequency running downward and time to the right.

plane into multiple wavelet trees is shown graphically in Fig. 1.

The recursive shifting and dilation performed by the DWT is also the reason for some desirable properties for the analysis of natural sounds [8]: *Locality*, i.e. each bandpass atom ψ_i is localized in both time and frequency, which implies *multi-resolution*, i.e. a nested set of scales in time and frequency is analyzed and *compression*, i.e. the wavelet coefficient matrices of many real-world signals are nearly sparse. These properties are desirable for the goal of estimating the statistics of a wavelet decomposition, as will become evident in the next paragraph.

3.2 Hidden Markov Tree Models

In general, a hidden Markov model introduces hidden *state variables* that are linked in a graphical model with Markov dependencies between the states, as is the case for the widely used hidden Markov model (HMM). Often the hidden states can be viewed as encoding a hidden physical *cause* that is not directly observable in the signal itself or its transformation in feature space.

In our research we focus on MR Markov processes that are defined on pyramidally organized binary trees, in particular the hidden Markov tree model. In this model, each node in a wavelet decomposition tree is identified with a mixture model, i.e. a hidden, discrete valued state variable with M possible values and an equal number of parametric distributions (usually Gaussians) corresponding to the individual values of the hidden state (Fig. 2).

Instead of assuming a Markov dependency on the wavelet coefficients as in parametric estimation methods (see Section 2), the HMT model introduces a first order Markov dependency between a given hidden state and its children. In other words and for the example tree in Fig. 2, given their parent state variable s_1, the subtrees rooted at the children s_2 and s_3 are conditionally independent. Similarly, since the wavelet coefficients are modeled by a distribu-

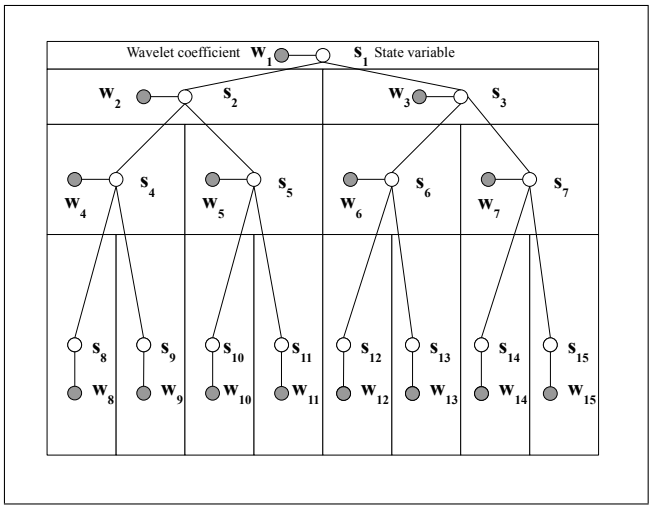


Figure 2: Hidden Markov tree model. Associated with each wavelet coefficient at a certain position in the tree structure (black node) is a hidden state variable (white node), that indexes into a family of parametric distributions.

tion that is only dependent on the node's state, w_2 and w_3 are also independent of the rest of the tree given their parent state s_1. Given enough data for parameter estimation and by increasing the number of states M, we can approximate the marginal distributions of wavelet coefficients to arbitrary precision. This allows us to model marginal distributions that are highly non-Gaussian, but are Gaussian conditioned on the parent state variable.

Even though the wavelet transform can be considered a de-correlator and the decomposition is sparse, the wavelet coefficients of real-world signals can not be considered independent, and there remain inter-coefficient dependencies that need to be taken into account in a statistical model. Fig. 3 shows histograms of wavelet coefficients for a particular scale in natural sound signals. These distributions

are sharply peaked around zero with long symmetric tails, corresponding to the sparsity property of the wavelet transform: there's a large number of small and only a few large coefficients.

For modeling the non-Gaussian wavelet coefficient marginal statistics, we use a two-state Gaussian mixture model, where one state encodes the peaked distribution of small coefficients and the other state encodes the tailed distribution of high-valued coefficients. For wavelet coefficients $\mathbf{w} = (w_1, \ldots, w_N)$ and hidden states $\mathbf{s} = (s_1, \ldots, s_N)$, the HMT is defined by the parameters

$$\begin{aligned} \theta = \{p_{s_1}(m), \epsilon^{mr}_{i,p(i)}, \mu_{i,m}, \sigma^2_{i,m}\} \\ m, r \in \{0,1\}, 1 \leq i \leq N \end{aligned} \tag{3}$$

with:

- $p_{s_1}(m) = P(s_1 = m)$, the probability of the root node s_1 being in state m,
- $\epsilon^{mr}_{i,p(i)} = P(s_i = m | s_{p(i)} = r)$, the conditional probability of child s_i being in state $m \in \{0,1\}$ given the parent $s_{p(i)}$ is in state $r \in \{0,1\}$,
- $\mu_{i,m}$, the mean of the wavelet coefficient w_i given s_i is in in state m $(1 \leq i \leq N)$ and
- $\sigma^2_{i,m}$, the variance of the wavelet coefficient w_i given s_i is in state m $(1 \leq i \leq N)$.

3.2.1 Training of the HMT

In order to find the best parameters fitting a given source sound, we update the model parameters θ given the training data $\mathbf{w} = \{w_i\}$ (a forest of binary wavelet trees, see Section 3.1) using a *maximum likelihood* (ML) approach. The *expectation maximization* (EM) framework provides a solid foundation for estimating the model parameters θ and the probabilities of the hidden states $\mathbf{s}$ and has been formulated for wavelet-based HMM's in [8]. The objective function to be optimized is the *log-likelihood function* $\ln f(\mathbf{w}|\theta)$ of the wavelet coefficients given the parameters. The EM algorithm iteratively updates the parameters until converging to a local maximum of the log likelihood.

In the following we provide a schematic description of the algorithm. More information on how we initialized our models and on the convergence criterion can be found in Section 3.3.

1. **Initialization**
   (a) Select an initial model estimate θ^0,
   (b) Set iteration counter $l = 0$.
2. **E step**: Calculate $P(\mathbf{s}|\mathbf{w}, \theta^l)$, the probability of the hidden state variables $\mathbf{S}$, yielding the expectation

$$Q(\theta|\theta^l) = \sum_{\mathbf{s} \in \{0,1\}^N} P(\mathbf{s}|\mathbf{w}, \theta^l) \ln f(\mathbf{w},\mathbf{s}|\theta). \tag{4}$$

3. **M step**: Update parameters θ, in order to maximize $Q(\theta|\theta^l)$:

$$\theta^{l+1} = \arg\max_{\theta} Q(\theta|\theta^l). \tag{5}$$

4. **Convergence**: Set $l = l+1$. If converged, then stop; else, return to E step.

3.2.2 E Step

The formulas for solving the hidden Markov tree E step presented in [8] are susceptible to underflow due to the multiplication of a large number of probabilities smaller than one. In [13] the authors develop an algorithm that is immune to underflow and computes the probabilities $p(s_i = m|\mathbf{w}, \theta)$ directly, instead of deriving them from $p(s_i = m, w = \mathbf{w})$. The probabilities $p(s_i = m, s_{\rho(i)} = n|\mathbf{w}, \theta)$, needed for computing the conditional state probabilities, can also be extracted directly from their algorithm. Similar to the original algorithm in [8], the above-mentioned probabilities are computed in separate *upward* and *downward* recursions, comparable to the computation of *forward* and *backward* variables in conventional hidden markov models. The algorithm has a slightly higher computational complexity than the one in [8], although it is still linear in the number of observation trees. For a more thorough treatment of the computations involved see [13].

3.2.3 M Step

After having calculated $P(\mathbf{s}|\mathbf{w}, \theta^l)$ in the E step, the M step consists in straight-forward closed-form updates of the conditional state probabilities and the parameters of the observation distributions.

First we calculate the probability of node i being in state m

$$p_{s_i}(m) = \frac{1}{K} \sum_{k=1}^{K} P(s_i^k = m|\mathbf{w}^k, \theta^l) \tag{6}$$

Then we update the model parameters by averaging over the quantities computed in the E-step for each of the K training examples:

$$\epsilon^{mr}_{i,p(i)} = \frac{\sum_{k=1}^{K} P(s_i^k = m, s_{p(i)}^k = r|\mathbf{w}^k, \theta^l)}{K p_{s_{p(i)}}(r)} \tag{7}$$

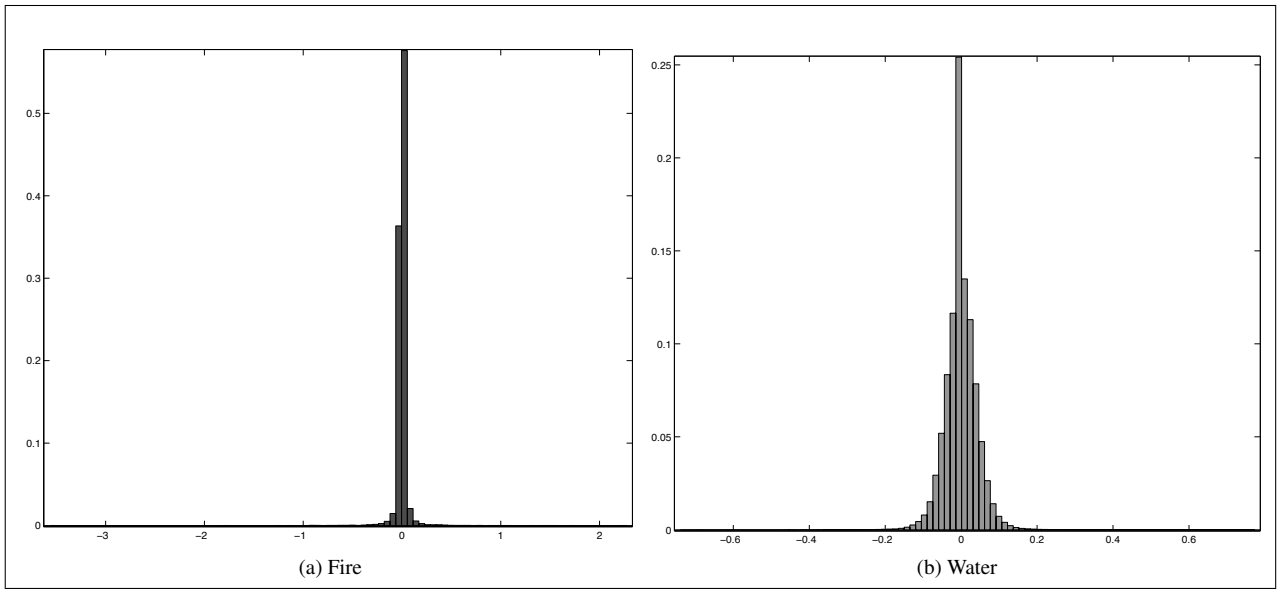
$$\mu_{i,m} = \frac{\sum_{k=1}^{K} w_i^k p(s_i^k = m|\mathbf{w}, \theta^l)}{K p_{s_i}(m)} \tag{8}$$

$$\sigma^2_{i,m} = \frac{\sum_{k=1}^{K} (w_i^k - \mu_{i,m})^2 P(s_i^k = m|\mathbf{w}^k, \theta^l)}{K p_{s_i}(m)} \tag{9}$$

3.3 Application to Sound Texture Synthesis

In this section we describe how the hidden Markov Tree model is adapted to a sound texture synthesis application. [1]

[1] All of the algorithms used in this work were implemented in the functional programming language Haskell and a link for downloading the package can be found at `http://mtg.upf.edu/people/skersten?p=Sound%20Texture%20Modeling`



(a) Fire (b) Water

Figure 3: Histograms of wavelet coefficients on the first (finest) scale of a five-level decomposition for two natural sounds, fire (left) and running water (right). The wavelet coefficient statistics for fire are clearly non-Gaussian, while for running water the statistics approach the Gaussian distribution.

3.3.1 Wavelet decomposition

The signal is first decomposed into wavelet coefficients on different scales, using Daubechies wavelet functions with five and ten vanishing moments [16].

The wavelet decomposition yields a forest of binary trees, each rooted at one of the corse scale wavelet coefficients. The length of the window corresponding to an individual tree depends on the depth of the wavelet decomposition. In our experiments, we chose a decomposition depth of 15 and 16, respectively, which corresponds to a context length of $2^{15} = 32768$ (or 0.74s at a sample rate of 44100) and $2^{16} = 65536$ (or 1.5s). In order to yield an even number of trees, the signal was truncated to an integer multiple of the context length in samples, i.e. from a signal of length n with a decomposition depth of m we used the first $\lfloor \frac{n}{2^m} \rfloor * 2^m$ samples. We worked exclusively with monophonic sounds and extracted the left channel from sounds that have originally been recorded in stereo.

3.3.2 Model construction

The wavelet decomposition tree structure is mapped to a HMT by associating each coefficient i with a hidden state variable s_i that can take one of two discrete values, depending on the value of the parent's state variable $s_{\rho(i)}$ (see Section 3.2). Together with one normal observation distribution per state value, each node forms a Gaussian mixture model that approximates the statistics of a coefficient at a certain position in the tree. The model has the same number of nodes as a single wavelet tree and all the trees in a decomposition forest are regarded to be independent samples from the same underlying distribution which corresponds to the parameter tying *across trees* described in [8].

In order to simplify the model, we don't take the approximation coefficient corresponding to each corse scale coefficient into consideration, although an extension to a two-dimensional Gaussian mixture for the root node would be straight-forward.

3.3.3 Model initialization and training

Since the EM algorithm only converges to a local minimum of the likelihood function (see Section 3.2), it is important to find a good initial estimate of the model's parameters. Following [17], we initialize the conditional state probabilities and the Gaussian distribution parameters by fitting a two-state Gaussian mixture model (GMM) to each level of the wavelet decomposition (the corresponding levels of all trees are concatenated). Once the GMM parameters have been found by the EM algorithm for Gaussian mixtures [18], an initial estimate of the conditional state probabilities $\epsilon^{mr}_{i,p(i)}$ is found by averaging over the number of joint state occurrences for each tree node

$$\epsilon^{mr}_{i,p(i)} = \frac{\#(s_i = m \text{ and } s_{\rho(i)} = n)}{\#(s_{\rho(i)} = n)} \tag{10}$$

During training, each tree of the decomposition forest is presented to the HMT model as an independent training example. In the E-step, the probabilities $P(S_i = m|\theta, w_i)$ and $P(S_i = m, S_{\rho(i)} = n|\theta, w_i)$ are determined as described in Section 3.2.2. The M-step then proceeds to update the model parameters according to (7), averaging over all of the trees in the training set.

We trained our models until the training data log likelihood under the updated model in step $l + 1$ was within a margin t of 0.001 of the log likelihood under the model in step l or when a maximum number n of iterations had been reached:

$$r_l \equiv \frac{\ln f(\mathbf{w}|\theta^{l+1}) - \ln f(\mathbf{w}|\theta^{l+1})}{\ln f(\mathbf{w}|\theta^{l+1})} \tag{11}$$

terminate when $0 \leq t \leq r_l \vee l \geq n$.

3.3.4 Synthesis

Sampling from the model begins by choosing an initial state for the root of the tree based on the estimated probability mass function (pmf) and sampling a wavelet coefficient from the gaussian probability distribution function (pdf) associated with the node and the sampled state. The algorithm proceeds by recursively sampling state pmfs and observation pdfs at each node given the state of its immediate parent. After having sampled a number of trees from the model independently from each other—without any explicit tree sequence model—the resulting forest of binary wavelet coefficient trees is transformed to the time domain by the inverse wavelet transform.

4. RESULTS

For a first qualitative evaluation we selected two textural sounds, *fire* and *running water*, from a commercial collection of environmental sound effects [2].

Fig. 4 shows the spectrograms of the fire and the water sound, respectively (left column). The fire texture is composed of little micro-onsets stemming from explosions of gas enclosed in the firewood. Inter-onset intervals are in the range of a few milliseconds. The background is filled with hisses, little pops and some low frequency noise. The sound of a water stream on the other hand is characterized by its overall frequency envelope with a broad peak below 5 kHz and a narrow peak around 12 kHz, while the fine structure is not clearly discernible in the spectrogram.

Informally evaluating the synthesis results by listening [3] shows that the HMT model is able to capture key dependencies between wavelet coefficients of the textural sounds. In the case of fire, the model built from an analysis with the longer wavelet function with ten vanishing moments is not able to reproduce the extremely sharp transients present in the signal. All three fire reproductions capture the overall perceptual quality of the original. This coherence is ensured by the HMT model by capturing the *across scale* coefficient dependencies. The temporal fine structure however can deviate significantly from the original: In all three cases the onset patterns are denser than in the source sound and lack sequential coherence. This can be explained with the fact that our model doesn't capture temporal, i.e. *within-scale* dependencies of wavelet coefficients explicitly. This missing feature roughly corresponds to the autocorrelation feature found to be important for the perception of textures in both image and sound [15, 3].

Similar to the sounds of fire, the synthesis of the water sound shows an overall similar spectral shape to the original, although an important spectral peak is missing from around 12 kHz and the high frequency content is more noisy in general (Fig. 4). In this sound, clearly noticeable *bubbles* form an important part of the temporal fine structure, and this feature is missing from the synthesis. We attribute this, as in the case of fire, to the missing autocorrelation feature in our synthesis model.

All of the synthesis examples show a repeating pattern with a length close to the wavelet tree size, i.e. directly related to the decomposition depth, although there is some minor within-loop variation. This result is an indication that the model is overfitted to the source material and can be explained with the relatively low number of training examples per tree model (around 7 wavelet trees per 10 seconds of source material). We could alleviate the overfitting effect in two ways: firstly, by using a significantly larger training set, and secondly, by *tying* parameters of correlated wavelet coefficients and thereby reducing the number of states and the number of mixture components. Simply tying parameters within one level of the wavelet decomposition however was found to be inadequate, because temporal fine structure gets lost and the synthesis result resembles a noisy excitation with roughly the spectral envelope of the original.

In order to quantitatively assess the synthesis quality, we conducted a small listening experiment with eleven subjects. We selected three sound examples for each of the five texture classes *applause*, *crowd chatter*, *fire*, *rain* and *running water* from the Freesound database [4]. We trimmed the sounds to the first 20 seconds, selected the left channel and downsampled this sound portion to a uniform sample rate of 22.5kHz. We then trained a model for each of the sounds using a wavelet tree decomposition of a depth of 16, i.e. an analysis frame length of 1.5s, and stopping training after 40 iterations. By sampling from the models we synthesized an eight second audio clip for each original sound file and presented the examples in random order. In a forced choice test, the subjects had to assign each synthesized sound to one of the five texture classes.

Table 1 shows the confusion matrix of the listening experiment and Table 2 lists the per-class accuracy. Apparently our model adequately captures the key perceptual properties of the respective sound classes except in the case of *water* and *rain*. The rain/water confusion can be explained with the missing "larger-scale" fine structure in the water examples (bubbling, whirling) that draws them closer to the noisy nature of the synthesized rain. While *applause* gets confused with rain on a surface because of the perceptual similarity between the micro-onsets that comprise those texture sounds, the vocal quality of the *crowd* chatter is a clearly distinguishing feature, even if poorly synthesized.

5. CONCLUSIONS

In this work we approached the problem of sound texture synthesis by application of a multi-resolution statistical model. Our contribution is a model that is able to capture key dependencies between wavelet coefficients for certain classes of textural sounds. While the synthesis results highlight some deficiencies that need to be addressed in future work, a parametric probabilistic approach to sound texture modeling has important advantages:

[2] *Blue Box SFX*, `http://www.eastwestsamples.com/details.php?cd_index=36`, accessed 2010-04-27.

[3] The synthesis results of our experiments are available on the web for reference, `http://mtg.upf.edu/people/skersten?p=Sound%20Texture%20Modeling`, accessed 2010-06-14

[4] `http://freesound.org`

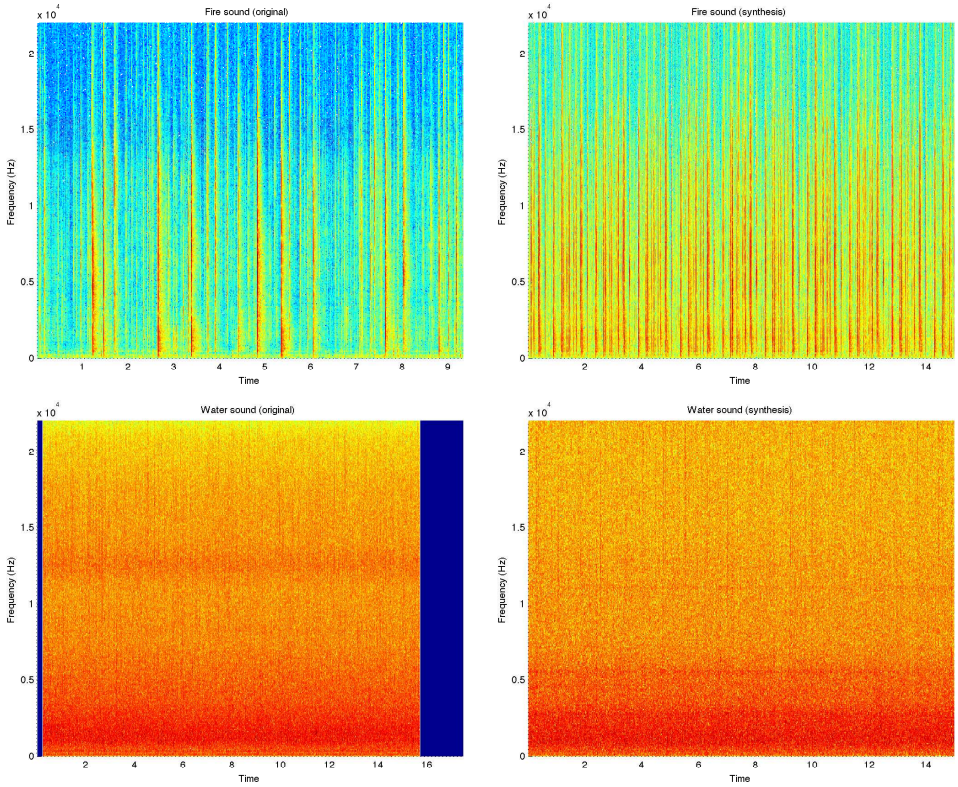


Figure 4: Spectrograms of a fire sound (top left), its synthesis (top right) a water stream sound (bottom left) and its synthesis (bottom right). Both sounds were recorded at a sample rate of 44100 kHz. The spectrum analysis was performed with a window size of 1024 and a hop size of 256.

		Predicted				
		applause	crowd	fire	rain	water
Actual	applause	13	1	0	7	1
	crowd	0	24	1	5	2
	fire	0	0	30	0	3
	rain	1	1	2	17	12
	water	6	0	2	19	6

Table 1: Confusion matrix for the listening experiment's results with five sound classes of three examples each and eleven subjects. Due to an error during the model building process, the applause class contains only two examples. One user classification for the crowd class was not submitted.

	Class				
	applause	crowd	fire	rain	water
Accuracy	0.59	0.75	0.91	0.52	0.18

Table 2: Class accuracies obtained in the listening experiment.

- *Probabilistic priors* can be used to deal with insufficient training data or to expose expressive synthesis control parameters.
- The model can be applied to *inference tasks* like classification, segmentation and clustering.

When comparing the synthesized sounds to their original source sounds it becomes evident that the model fails to capture some features that are crucial for auditory perception of texture, most notably the intra-scale autocorrelation feature. Another major limitation is the inadequate representation of infinite time series, because our model divides the signal into *blocks* of a size determined by the model tree depth, thereby introducing artifacts caused by the position of the signal relative to the beginning and the end of the block.

The most intuitive approach to overcome these limitations is to modify the graphical tree model itself, by allowing additional conditional dependencies between nodes on the same hierarchy level. Because graphs that satisfy certain conditions on their structure, and in particular on the cycles formed by their edges, still allow for efficient parameter estimation in the EM framework—see [19] for a thorough treatment—it is possible to model within-scale coefficient dependencies without resorting to Markov-chain Monte-Carlo or other simulation methods. The significant increase in the number of parameters needs to be addressed by aggressive tying, i.e. by using the same parameters for

a set of variables in the model that exhibit the same statistics. While tying within tree levels yields unsatisfactory results for the model described in this paper, a modified model might be able to capture just enough temporal correlations to make this tying scheme feasible. By explicitly modeling dependencies across time, the wavelet decomposition depth wouldn't be the only way to capture temporal context any longer and could be decreased significantly, resulting in a vastly reduced set of parameters.

6. ACKNOWLEDGMENTS

Many thanks to Jordi Janer, Ferdinand Fuhrmann and the anonymous reviewers for their valuable suggestions and to those who kindly participated in our listening test. This work was partially supported by the ITEA2 Metaverse1 Project [5]. The first author is supported by the FI-DGR 2010 scholarship of the Generalitat de Catalunya, the second author holds a Juan de la Cierva scholarship of the Spanish Ministry of Science and Innovation.

7. REFERENCES

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

[2] N. Saint-Arnaud, *Classification of Sound Textures*. Master thesis, Massachusetts Institute of Technology, 1995.

[3] J. H. McDermott, A. J. Oxenham, and E. P. Simoncelli, "Sound texture synthesis via filter statistics," in *Proceedings IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, (New Paltz, NY), Oct. 2009.

[4] 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*, vol. 22, pp. 38–48, July 2002.

[5] D. ORegan and A. Kokaram, "Multi-Resolution sound texture synthesis using the Dual-Tree complex wavelet transform," in *Proc. 2007 European Signal Processing Conference (EUSIPCO)*, 2007.

[6] A. Kokaram and D. O'Regan, "Wavelet based high resolution sound texture synthesis," in *Proc. 31st International Conference: New Directions in High Resolution Audio*, June 2007.

[7] D. D. Po and M. N. Do, "Directional multiscale modeling of images using the contourlet transform," *IEEE Transactions on Image Processing*, vol. 15, no. 6, p. 16101620, 2006.

[8] M. Crouse, R. Nowak, and R. Baraniuk, "Wavelet-based statistical signal processing using hidden markov models," *Signal Processing, IEEE Transactions on*, vol. 46, no. 4, pp. 886–902, 1998.

[5] http://www.metaverse1.org

[9] G. Strobl, G. Eckel, D. Rocchesso, and S. le Grazie, "Sound texture modeling: A survey," 2006.

[10] A. A. Efros and T. K. Leung, "Texture synthesis by Non-Parametric sampling," in *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, p. 1033, IEEE Computer Society, 1999.

[11] A. Willsky, "Multiresolution markov models for signal and image processing," *Proceedings of the IEEE*, vol. 90, no. 8, pp. 1396–1458, 2002.

[12] H. Choi, J. Romberg, R. Baraniuk, and N. Kingsbury, "Hidden markov tree modeling of complex wavelet transforms," in *Acoustics, Speech, and Signal Processing, 2000. ICASSP '00. Proceedings. 2000 IEEE International Conference on*, vol. 1, pp. 133–136 vol.1, 2000.

[13] J. Durand, P. Goncalves, and Y. Guedon, "Computational methods for hidden markov tree models-an application to wavelet trees," *Signal Processing, IEEE Transactions on*, vol. 52, no. 9, pp. 2551–2560, 2004.

[14] D. H. Milone, L. E. D. Persia, and M. E. Torres, "Denoising and recognition using hidden markov models with observation distributions modeled by hidden markov trees," *Pattern Recogn.*, vol. 43, no. 4, pp. 1577–1589, 2010.

[15] G. Fan and X. G. Xia, "Wavelet-based texture analysis and synthesis using hidden markov models," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 50, no. 1, p. 106120, 2003.

[16] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Communications on Pure and Applied Mathematics*, vol. 41, no. 7, pp. 909–996, 1988.

[17] G. Fan and X. G. Xia, "Improved hidden markov models in the wavelet-domain," *IEEE TRANS SIGNAL PROCESS*, vol. 49, no. 1, p. 115120, 2001.

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

[19] H. Lucke, "Which stochastic models allow Baum-Welch training?," *Signal Processing, IEEE Transactions on*, vol. 44, no. 11, pp. 2746–2756, 1996.