Katholieke Universiteit Leuven

Departement Elektrotechniek

ESAT-SISTA/TR 1996-78

Applications of Wavelets in Audio and Music[1]

Peter De Gersem Bart De Moor Marc Moonen[2]

December 1996

Published in *Wavelet analysis: a new tool in signal and image processing*, KVIV Study-day, Antwerp, Belgium, Dec. 11, 1996

[1]This report is available by anonymous ftp from *ftp.esat.kuleuven.ac.be* in the directory *pub/SISTA/degersem/reports/report96-78.ps.gz*

[2]K.U.Leuven, Dept. of Electrical Engineering (ESAT), Research group SISTA, Kardinaal Mercierlaan 94, 3001 Leuven, Belgium, Tel. 32/16/32 17 09, Fax 32/16/32 19 70, WWW: *http://www.esat.kuleuven.ac.be/sista.* E-mail: *bart.demoor@esat.kuleuven.ac.be.* This work was supported in part by the Concerted Research Action MIPS of the Flemish Government, and the F.W.O.-Flanders.

Applications of Wavelets in Audio and Music

Peter De Gersem* Bart De Moor† Marc Moonen†

K.U.Leuven-ESAT/SISTA
Kard. Mercierlaan 94, 3001 Heverlee (Leuven), Belgium
Tel. +32-(0)16-32.18.07 Fax. +32-(0)16-32.19.70
E-mail: `Peter.DeGersem@esat.kuleuven.ac.be`
URL: `http://www.esat.kuleuven.ac.be/~degersem/`

Abstract

There are obvious links between the wavelet transform and music: both are time-frequency representations/phenomena, and in both cases the frequency-range of interest is perceived logarithmically. Therefore it seems feasible to look for applications of wavelets in music.

Research on music or audio processing using wavelets is mainly concentrated on the Continuous Wavelet Transform (that can be used to extract features from, and modify, musical signals), and on the Wavelet Packet transform (that can be used for compression, de-noising, and granular synthesis).

In this paper, we make an attempt at giving a research and literature overview.

*Research Assistant of the Fund for Scientific Research - Flanders (Belgium)(F.W.O.)
†Research Associate of the F.W.O.

1 Introduction

Since the very beginning of the wavelet-era, *wavelet analysis* has been associated with *music* (e.g. [5, 23]), since both are time-frequency phenomena. Music can be described with *notes*, which have a time-component (starting time, duration), and a frequency-component (pitch); a wavelet $\psi(\frac{t-b}{a})$ has time-component b and frequency-component a (the scale). More, wavelet-analysis applies a *constant-Q* filter bank to a signal, such that higher frequencies are studied with less frequency-resolution and more time-resolution than lower frequencies. This property is very similar to music, because of the exponential spacing of notes (raising a note one octave is equivalent to doubling its frequency). Enough reasons to start looking for applications of wavelets in music.

The first research efforts were carried out at the University of Aix-Marseille II, France, right there where Grossmann and Morlet had just developed basic wavelet-theory. And nowadays, the first commercial musical applications of wavelets become available, be it only as a preprocessing step for a neural network[1].

Research in this area belongs to the domain of *Computer Music* [19], encompassing the whole process of composing, analyzing, processing, and performing music with the help of computers. Aspects of Computer Music are sound-synthesis, sound-processing, time-frequency analysis of sounds, algorithmic composition, music software, the musicians interface, ... There is a lot of engineering work involved in all these aspects, but an important thing is that the final word is said by the composer/musician who will have to use the resulting products, or our ears that have to hear the sound.

The main body of this paper consists of three sections, which describe some musical applications of the different *wavelet transforms*: the Continuous Wavelet Transform (CWT), the Discrete Wavelet Transform (DWT), and Wavelet Packets (WP). Each section will contain some relevant wavelet-theory, of which more details can be found in [5, 23, 25].

This overview is by no means complete, and contains only some of the most prevailing material.

2 Continuous Wavelet Transform

The Continuous Wavelet Transform (CWT) transforms a one-dimensional signal $f(t)$ into a two-dimensional time-scale function (often called *image*)

$$\mathcal{F}(a,b) = \int_{-\infty}^{\infty} f(t)\psi^*_{a,b}(t)dt$$

where

$$\psi_{a,b}(t) = \frac{1}{\sqrt{|a|}}\psi\left(\frac{t-b}{a}\right)$$

[1] Prosoniq, `http://www.prosoniq.com/`.

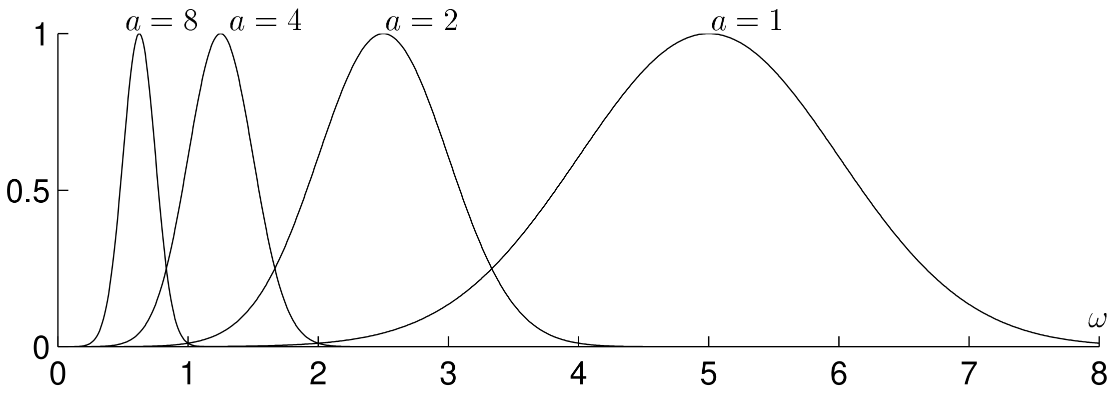


Figure 1: Constant-Q behavior of the Morlet-wavelet.

are the wavelets derived by translation (b) and dilation (a) from the mother wavelet $\psi(t)$. The mother wavelet $\psi(t)$ is a function limited in time and frequency, and has to satisfy an admissibility condition which basically forces $\int_{-\infty}^{\infty} \psi(t)dt = 0$. With these properties, a wavelet is typically a smooth (limited in frequency), and short (limited in time) wave (the integral must be zero). Typical examples are the Mexican Hat wavelet $\psi(t) = (1 - t^2)e^{-\frac{t^2}{2}}$, and the Morlet-wavelet (a modulated Gaussian) $\psi(t) = e^{j\omega_o t}e^{\frac{-t^2}{2}}$.

To illustrate the constant-Q behavior of the wavelet-transform, we can write the CWT-formula also in the frequency-domain:

$$\hat{\mathcal{F}}(a, \omega) = C_a \hat{\psi}^*(-a\omega)\hat{f}(\omega) .$$

Because $\psi(t)$ has a band-pass characteristic (e.g. the Morlet-wavelet has $\hat{\psi}(\omega) \sim e^{-\frac{(\omega-\omega_0)^2}{2}}$), the CWT for each a is calculated by filtering $f(t)$ with $\psi(\frac{t}{a})$, and the relative bandwidth of the filters for each a is the same (the center-frequency of the band-pass filter changes, but the width of the pass-band changes equally, See fig. 1).

The function $f(t)$ can be recovered from the CWT using

$$f(t) = C_\psi^{-1} \int_0^\infty \frac{da}{a^2} \int_{-\infty}^\infty \mathcal{F}(a, b)\psi_{a,b}(t)db ,$$

where C_ψ is a finite constant, dependent on the chosen wavelet $\psi(t)$. There are some approximate re-synthesis formulas available, of which

$$f(t) \simeq K_\psi \int_0^\infty \mathcal{F}(a, t)\frac{da}{a^{3/2}}$$

is most frequently used [15].

Obviously, an implementation of the CWT will have to work with a finite number of discrete data. Most probably, the signal to be transformed will be given already in discrete

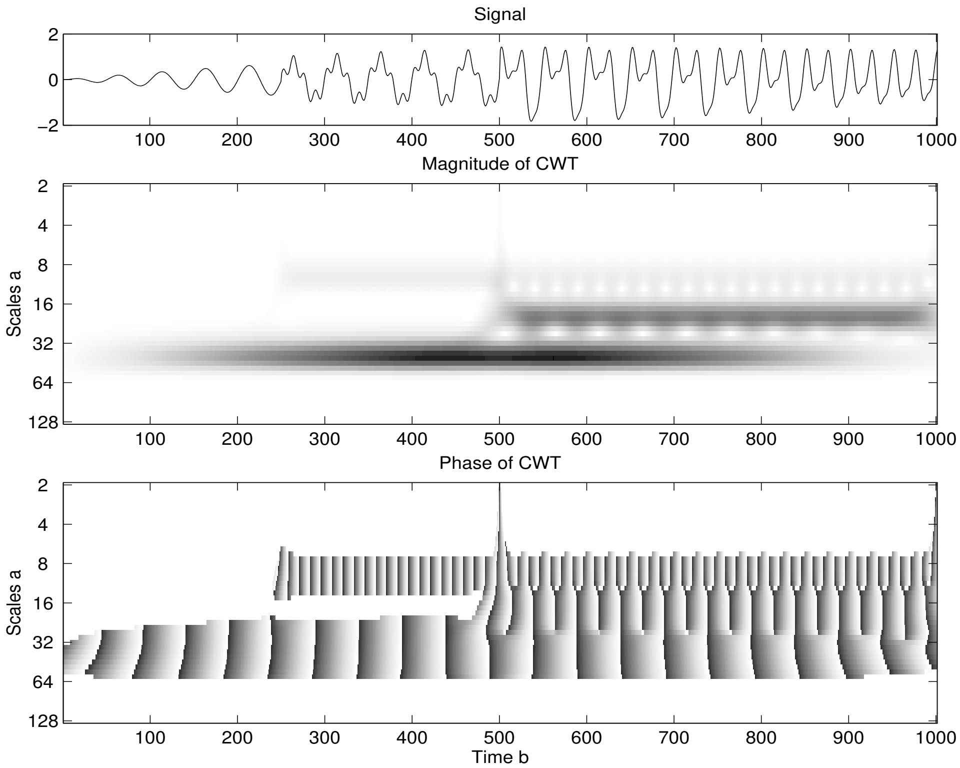


Figure 2: CWT image of a signal. The signal exists of three sine waves which start at different times, each with double frequency. The used wavelet is the Morlet-wavelet.

samples. To calculate the CWT of this, the a and b parameters should be discretized as well. For b, this is trivial, because the input-signal is discrete. Values of a are often chosen exponentially: $a_0^{m/q}$, with $a_0 = 2$, q the number of *voices/octave*, and m integers. The theoretic details of this discretization are covered by the theory of *frames* [5]. For each scale a, the CWT can be calculated as a convolution of the discrete signal and the discretized wavelet $\psi_{a,b}(t)$.

2.1 Analysis

The CWT originally served as an analysis tool for seismic data. Soon, engineers interested in music started applying the CWT for analyzing musical signals. One of the first papers dealt with "detection of abrupt changes in sound signals" [11]; indeed, detecting discontinuities in signals is generally considered as one of the major application areas of the CWT. Another early paper described some possibilities in "analyzing sound patterns" [14]. An example of a CWT is shown in fig. 2.

The CWT builds a time-scale image of a signal, similar to the Short-Time Fourier Transform (STFT), but with the following differences [18]. The STFT first extracts a window from the signal to be analyzed, and then takes the FFT of the selected time-interval. For each time-interval, the local frequency-content is analyzed through a set of linearly spaced band-pass filters. The CWT analyzes, for each scale, the whole signal through a constant-Q band-pass filter. For the interpretation of CWT-images, it is important to know that each scale corresponds to all the frequencies in the passband of the band-pass filter corresponding to that particular scale; also two neighboring scales in general have overlapping passbands (solving this by taking band-pass filters with a sharp cut-off, has the disadvantage of prolonging the impulse response of that filter, i.e. the wavelet). The main advantage of the CWT over the STFT is that the details show up where they should: high frequency-resolution at low frequencies, high time-resolution at high frequencies. Important for this is the Heisenberg-principle: it is impossible to obtain high time- and frequency resolution at the same time, there is a trade-off to be made (high time-resolution implies low frequency-resolution, and vice versa). Both the STFT and the CWT can be used for time-frequency-analysis.

Now for the analysis itself: music is a non-stationary phenomenon, its frequency content changes with time. Time-frequency analysis allows that time-varying frequency content to be displayed. A simple idea (but not so simple to realize) is trying to determine the pitch of each note played. On a deeper level, one could analyze the time-varying spectrum of one note of a particular instrument, e.g. the different harmonics[2] of such a sound can decay at different rates. Such applications are described in [15, 16]. Another type of analysis works by tracking the maxima of the CWT at each time-instant, which generates the *skeleton* of a signal. This can be used to extract the modulation laws used for generating the signal [7, 12]. Of course, several other analysis-applications can be conceived, e.g. [21].

2.2 Time-Stretching, Pitch-Shifting and other Processing

Analysis is one issue, but of course it is also possible to change parameters in the transformation-domain, re-transform, and listen to the resulting signal. A simple example consists in zeroing the coefficients corresponding to a particular scale, which amounts to band-reject filtering the signal.

A very puzzling application is *time-stretching* (and the equivalent *pitch shifting*), i.e. changing the length of an audio-signal without changing its pitch, or transposing the pitch without changing the length. This is by no means an obvious operation. The usual time- or frequency-scaling is done by resampling, which changes both aspects simultaneously (time-compressing a signal lets it sound higher).

A rigorous problem-formulation is difficult, because there is no *model* that describes music or audio in general. For most music, the effect we want to achieve is clear: we want the

[2]The sound of a pitched instrument is periodic, and can be decomposed in its Fourier-series. Each frequency that is a multiple of the base frequency is called a *harmonic*, *partial*, or *overtone*. The relative strength of each harmonic is one of the determining factors in how the instrument sounds (timbre). The other aspect of timbre is the time-variance of these harmonics.

same piece of music (the same notes), but played at a different tempo, or transposed. For speech, there is already an extra difficulty, because shifting all frequencies up (or down), changes the *formants*: not only the voice sounds higher, but also the timbre changes, and in most cases, we just want the same voice but with adjusted pitch. A similar badly defined aspect is the attack-transient of a note: while stretching the duration of a note, we would like the attack-transient to remain constant, because that is what happens in reality too. The general problem does not have a rigorous solution, the result will be signal-dependent, and our ear will have to decide if the result is acceptable. In practice, some good algorithms already exist, some even work in real-time. Note that time-stretching and pitch-shifting are equivalent, one can go from one to the other by resampling the (discrete) signals.

For a simple model (that of music played by perfectly pitched instruments, each with its own timbre), we can describe the signal more or less by

$$f(t) = \sum_i A_i(t) \sin(\omega_i(t)t) \,.$$

Frequency stretching by a factor c in this case would consist in the transforming $f(t)$ into

$$g(t) = \sum_i A_i(t) \sin(c\omega_i(t)t) \,.$$

The idea with wavelets is to stretch or compress the time-scale *image* produced by the CWT (cf. fig. 2). E.g. for pitch-shifting, we change the scale of the a-axis by the stretching factor, and re-transform. Indeed, the time-scale image we expect from a pitch-shifted sound must be like our stretched image. What happens *under the hood* is a decomposition of the signal in the wavelet-domain, as a coordinate with respect to a set of basis-functions (the wavelets $\psi_{a,b}(t)$); stretching the scale-parameter by factor c is equivalent to replacing the basis-functions $\psi_{a,b}(t)$ by their pitch-shifted versions $\psi_{a/c,b}(t)$ (which is valid, because a scaled wavelet is still a wavelet).

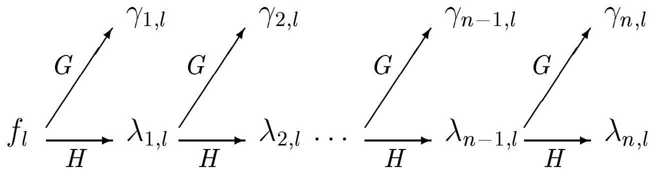
In [15, 16], this solution is worked out with the Morlet-wavelet, which is complex-valued. This has the consequence that the phase of the transform-parameters (which are complex because we analyze a real signal with a complex wavelet) must also be adjusted to allow the solution to be *possible*[3]. Other publications where this problem is described are [1, 2].

3 Discrete Wavelet Transform

Going from the CWT to the Discrete Wavelet Transform (DWT) is not such a big step. The first thing we should notice, is that wavelets on a small scale have a shorter duration than wavelets on a large scale. So it seems not such a bad idea to adapt the discretization

[3]In general, it is highly probable that an arbitrary time-scale image can not be generated by a CWT of any function $f(t)$; that is because one transforms a 1-dimensional function into a 2-dimensional function, thus introducing an extra degree of freedom. All *proper* time-scale images belong to the *reproducing kernel Hilbert subspace* corresponding to the wavelet. The interested reader is referred to [5] for details!

a)



- f_l $(= \lambda_{0,l})$ is the discrete input signal
- H and G form the subband filtering scheme followed by downsampling (by factor 2), corresponding to the wavelet ψ: H is the high-pass filter, G is the low-pass filter
- $\lambda_{m,l}$ are calculated by applying filter H to $\lambda_{m-1,l}$
- $\gamma_{m,l}$ are the wavelet coefficients on scale 2^m, and are calculated by applying filter G to $\lambda_{m-1,l}$

b)

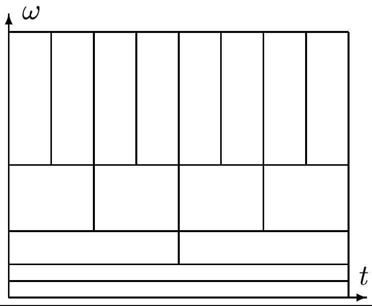


Figure 3: a) Principle of DWT operation, and relation with subband filtering, b) Tiling of the time-frequency plane.

of the time-steps b to the current scale a: for large scales, we can work with a sparser time-sampling than for small scales. To cut a long story short, the most frequently used sampling is $a = 2^m$ and $b = n2^m$, and the wavelet transform becomes $\mathcal{F}(m,n) = \int_{-\infty}^{\infty} f(t)\psi^*_{m,n}(t)dt$, with $\psi_{m,n} = \psi(2^{-m}t - n)$, and m, n integers. In general, this does not allow us to re-transform easily. To obtain an easy reconstruction, we must be able to write

$$f(t) = \sum_{m,n} c_{m,n}\psi_{m,n}(t) \, .$$

This is possible through a theory called *Multiresolution Analysis* (MRA, for details check [5, 23]), and this also leads to the Fast Wavelet Transform (FWT), which has strong links with perfect reconstruction subband filtering. The MRA can be described for continuous signals, and its extension to discrete signals is straightforward.

For this section, it is important to understand the link between subband filtering and the FWT (which is just the fast implementation of the DWT), see fig. 3a. The FWT-algorithm can be described as follows: given a discrete signal, we first calculate the wavelet transform on the smallest scale, the highest level of detail, this corresponds to the high-pass filtering plus down-sampling step (H) in the subband filtering scheme; the signal is also low-pass filtered by the conjugate subband filter, and down-sampled (G); this procedure is iterated on the low-pass filtered signal, until the largest scale is reached. Some other important things to remember: the MRA restricts the choice of the wavelet ψ (it is impossible to get a wavelet with all good properties, one has to trade-off regularity, compact-support, orthogonality, vanishing moments, $\ldots$); the MRA is described completely by its filter-coefficients h; a signal of N samples is transformed into N samples, so there is no redundancy in the wavelet transform.

3.1 Traditional Applications

The DWT has several applications, mostly valid for all kinds of signals, including audio. Because of their generality, they will only be described briefly. Note that section 4.1 extends this description using Wavelet Packets.

Two straightforward applications of the DWT are *de-noising* and *compression*. Both work in principle by discarding the smallest coefficients in the transform, and it is a well known property of the wavelet transform that it transforms many "real-life" signals very efficiently, i.e. with a lot of insignificant (close to zero) coefficients.

Recovering the original signal out of noisy data using wavelets was pioneered by Donoho and co-workers [8, 9]. Their technique is called *wavelet-shrinkage*, and is very easy to describe: first, a coefficient-*threshold* is calculated from the standard-deviation of the noise on the signal; all wavelet-coefficients smaller than the threshold are replaced by zero, and all other coefficients are shrunk by the threshold[4]. The re-transformation is under general assumptions the optimal reconstruction of the original signal. This technique is the basis for more recent and advanced noise-reduction, especially for image-processing.

Discarding the insignificant coefficients introduced in the wavelet-transform gives a certain amount of data-reduction, without too much distortion in the reconstructed signal. This method resembles subband coding: apply a filter bank to the signal, and code each subband as efficiently as possible before transmitting; the receiver reconstructs the signal with the data it gets. Most implementations in this area however use Wavelet Packets, because this simple extension of the DWT offers a lot of advantages. See section 4.1 for more details and references.

3.2 Computer Music Applications

As far as we know, there is no *typical* application of the DWT in computer music. There are a lot of possibilities however. Much like the CWT, one can use the DWT for *analysis* of

[4] $\eta_t(w) = \mathrm{sgn}(w)\max(|w| - t, 0)$, with w a wavelet-coefficient, t the threshold, $\eta_t(w)$ the shrunk wavelet coefficient.

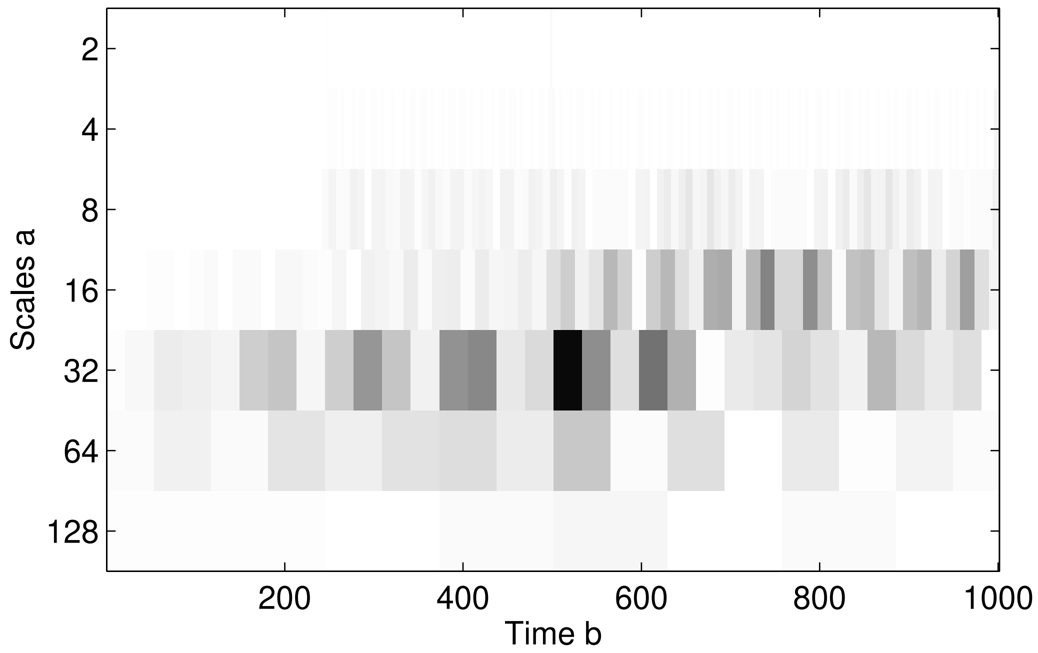


Figure 4: DWT image of the same signal as in fig. 2, with the D4-wavelet (Matlab Wavelet Toolbox).

signals. Each subband represents the signal in a certain frequency-range, with more time-resolution and less frequency-resolution (broader subbands) for higher frequencies (fig. 3b). But because of the down-sampling (critical sampling, no redundancy), the interpretation of the subbands is more difficult than with the CWT (fig. 4).

In [10], Evangelista describes how computer musicians can *play* with the different parameters to obtain new sounds. E.g. one could change (filter) the coefficients in one subband to obtain a different version of a given signal. It must also be possible to combine the subbands of two different sounds, the result of which can be a new sound combining properties of the two originals (morphing).

Another possible application could be the storage of samples in a sample-playback synthesizer (the most common type today). Ideally, each different key of the keyboard should be assigned its own sample, but for high-quality samples, the resulting memory requirement is too high. Therefore, commercial synthesizers store only a few different notes of a particular instrument in memory, and derive the other notes by interpolation of the wave-form (proper resampling is not used, because the ratio of the sample rates is different for each interval, and it is too computationally intensive). To obtain a realistic sound, the order of the interpolation-routine should be high enough, but that raises the computational cost. A new method could be to store the wavelet transform of the samples in memory (even obtaining some compression), and to interpolate each subband linearly before reconstructing the sample.

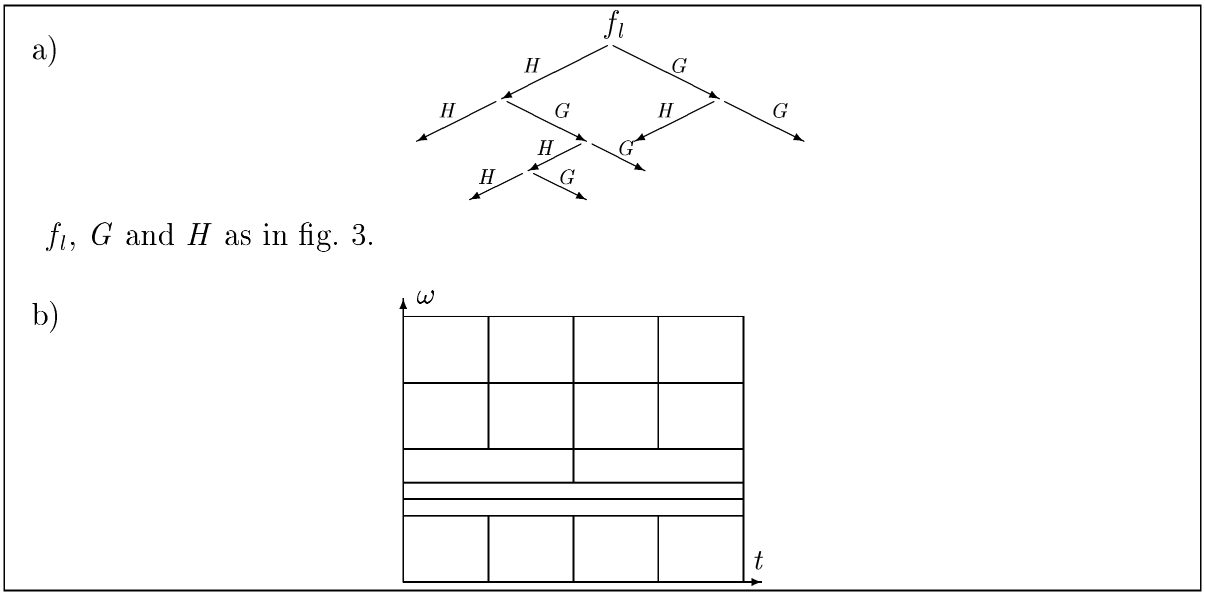


Figure 5: a) Example of a Wavelet Packet tree, b) Tiling of the time-frequency plane (for this particular tree).

4 Wavelet Packets

Wavelet Packets (WP) are simply an extension of the DWT. Instead of iterating the subband filter on the low frequency part, now the iteration is done on both subbands. This leads to a set of decompositions (by stopping the iteration at each possible level), that can be represented very well by a (binary) tree (fig. 5). The basis-functions of each subband are called *wavelet-packets*, because they still look like wavelets (small oscillating waves), but do not necessarily combine high frequencies with small time intervals, or vice versa. There exist very efficient algorithms to decompose a signal in all possible combinations, and to determine the most efficient combination of subband decompositions (efficiency is measured w.r.t. a particular criterion, often the *Shannon-entropy*). More information is available in [25]. Coifman and Wickerhauser from Yale University, the pioneers of this method, call it *Adaptive Waveform Analysis*.

4.1 Traditional Applications

Again (cf. section 3.1), *compression* and *de-noising* are the best known general applications of the WP transform, and both are valuable in music/audio processing as well. The operation-principle is similar to the DWT (the WP transform introduces lots of insignificant coefficients that can be discarded), but there is one extra parameter: one can choose the best WP transform out of the set of all possible transforms.

A fine example of audio de-noising is [3]. Berger, Coifman and others tried to reconstruct a wax-cylinder recording of Brahms himself playing one of his Hungarian Dances. Their technique is basically just a WP transform, in which the best basis was searched using the Shannon-entropy. The WP transform coefficients are divided in a coherent part and a noisy part by thresholding, and the same WP transform was iterated on the noisy part to get as much information as possible from the original signal. A big part of their work existed of finding good segmentation methods for the signals, because a large signal has to be threatened in parts, and the combination of the parts should not introduce new artifacts. The final result was not very clean, but the improvement in sound quality provided for a better idea of how Brahms interpreted his own work, something that was definitely not possible from the original. It has to be remarked that the same technique was also performed using local trigonometric bases instead of WP, and this resulted in musically more pleasing results.

Audio compression is a very hot topic nowadays (according to the number of proposed papers at major signal processing conferences). Commercial algorithms (MPEG, Dolby AC-3) use combinations of techniques, including subband filtering, perceptual coding, etc. But methods using the WP transform are now being developed with comparable or even improved performance. Again, the first steps were made at Yale University by Coifman and Wickerhauser [4, 24]. Since then, a great deal of research has been carried out, resulting in too publications to list here. In [20], a clear description is given of an audio codec using the WP transform, but with a fixed decomposition tree (namely one that matches the auditory critical bands quite closely). The adaptation is present in the choice of the mother wavelet: the algorithm chooses the wavelet in a given class that most efficiently codes the subbands. This is combined with some other techniques typical for compression, and the resulting coder achieves very low bit-rates, at the expense of a highly complex coder (the decoder is much faster). This work is continued in [22] and other publications, where, among other things, a DSP-implementation is described of a simplified version of this codec. In [13], a different strategy is proposed: in a first step, the harmonic content of the signal is coded, second, the remaining transients are coded by means of WP, and finally only noise remains, which again is coded very efficiently. With this separation principle, various perceptual masking effects can be exploited.

4.2 Granular Synthesis and other Computer Music Applications

The WP transform is very efficient in transforming musical signals, i.e. a few WP basis functions often suffice to accurately describe a sound. This observation can be used in sound-synthesis: a sound can be described by its (few) transform coefficients, as in compression, and the sound is synthesized again by a WP generator, triggered by the coefficients. This synthesis-method belongs to the category of *granular synthesis* [19], and is briefly described in [24], again from Wickerhauser. Also Kudumakis used a similar method in [17].

5 Conclusions

This paper describes some real and some possible applications of the different wavelet transforms for the processing of audio and music. The described methods look promising, but (to the best of our knowledge) there are no real, commercial, applications yet. Let us conclude optimistically by saying that wavelets are no doubt a valuable tool for the analysis, synthesis and compression of audio and music signals.

6 Acknowledgment

This research work was carried out at the ESAT laboratory of the Katholieke Universiteit Leuven, in the frame of the Concerted Research Action MIPS ('Model-based Information Processing Systems') of the Flemish Government. The scientific responsibility is assumed by its authors.

References

[1] D. Arfib, "Analysis, transformation, and resynthesis of musical sounds with the help of a time-frequency representation," in De Poli [6], pp. 87–118.

[2] D. Arfib and N. Delprat, "Musical transformations using the modification of time-frequency images," *Computer Music Journal*, vol. 17, no. 2, pp. 66–72, Summer 1993.

[3] J. Berger, R. Coifman, and M. Goldberg, "Removing noise from music using local trigonometric bases and wavelet packets," *Journal of the Audio Engineering Society*, vol. 42, no. 10, pp. 808–818, Oct. 1994. `http://www.music.yale.edu/research/brahms/brahms1.html`.

[4] R. Coifman, Y. Meyer, and M. Wickerhauser, "Wavelet analysis and signal processing," in *Wavelets and their Applications* (M. Ruskai *et al.*, eds.), pp. 153–178, Boston: Jones and Bartlett, 1992.

[5] I. Daubechies, *Ten Lectures on Wavelets*, vol. 61 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. SIAM, 1992.

[6] G. De Poli, ed., *Representations of Musical Signals*. MIT Press, 1991.

[7] N. Delprat, B. Escudié, P. Guillemain, R. Kronland-Martinet, P. Tchamitchian, and B. Torrésani, "Asymptotic wavelet and Gabor analysis: extraction of instantaneous frequencies," *IEEE Transactions on Information Theory*, Special Issue on Wavelet Transforms and Multiresolution Analysis, vol. 38, no. 2, pp. 644–664, Mar. 1992.

[8] D. Donoho, "De-noising by soft-tresholding," Tech. rep. 409, Stanford University, Department of Statistics, Nov. 1992.
`http://playfair.stanford.edu/reports/donoho/denoiserelease3.ps.Z`.

[9] D. Donoho, "Wavelet shrinkage and W.V.D. – a ten-minute tour," Tech. rep. 416, Stanford University, Department of Statistics, Jan. 1993.
`http://playfair.stanford.edu/reports/donoho/toulouse.ps.Z`.

[10] G. Evangelista, "Wavelet transforms that we can play," in De Poli [6], pp. 119–141.

[11] A. Grossman, M. Holschneider, R. Kronland-Martinet, and J. Morlet, "Detection of abrupt changes in sound signals with the help of wavelet transforms," in *Inverse Problems: An Interdisciplinary Study* (P. Sabatier, ed.), Advances in Electronics and Electron Physics, supplement 19, pp. 289–306, Academic Press, 1987.

[12] P. Guillemain and R. Kronland-Martinet, "Characterization of acoustic signals through continuous linear time-frequency representations," *Proceedings of the IEEE*, Special Issue on Wavelets, vol. 84, no. 4, pp. 561–585, May 1996.

[13] K. Hamdy, M. Ali, and A. Tewfik, "Low bit rate high quality audio coding with combined harmonic and wavelet representations," in *Proceedings of the 1996 International Conference on Acoustics, Speech, and Signal Processing (ICASSP'96)*, Atlanta, Georgia, pp. 1045–1048, May 7-9, 1996.
`ftp://oz.ee.umn.edu/pub/tewfik/1996/audio/audio96.ps.Z`.

[14] R. Kronland-Martinet, J. Morlet, and A. Grossman, "Analysis of sound patterns through wavelet transforms," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 1, no. 2, pp. 97–126, Aug. 1987.

[15] R. Kronland-Martinet, "The wavelet transform for analysis, synthesis and processing of speech and music sounds," *Computer Music Journal*, vol. 12, no. 4, pp. 11–20, Winter 1988.

[16] R. Kronland-Martinet and A. Grossmann, "Application of time-frequency and time-scale methods (wavelet transforms) to the analysis, synthesis, and transformation of natural sounds," in De Poli [6], pp. 45–85.

[17] P. Kudumakis and M. Sandler, "Synthesis/coding of audio signals based on inverse wavelet packet algorithms," in *Proceedings of the U.K. Symposium on Applications of Time-Frequency and Time-Scale Methods*, University of Warwick, Coventry, U.K., pp. 92–99, Aug. 30-31, 1995.
`http://www.eee.kcl.ac.uk/member/pgrads/p_kudumakis.html`.

[18] O. Rioul and M. Vetterli, "Wavelets and signal processing," *IEEE Signal Processing Magazine*, vol. 8, no. 4, pp. 14–38, Oct. 1991.

[19] C. Roads, *The Computer Music Tutorial.* MIT Press, 1996.

[20] D. Sinha and A. Tewfik, "Low bit rate transparent audio compression using adapted wavelets," *IEEE Transactions on Signal Processing*, Special Issue on Wavelets and Signal Processing, vol. 41, no. 12, pp. 3463–3479, Dec. 1993.

[21] L. Smith, "Modelling rhythm perception by continuous time-frequency analysis," in *Proceedings of the International Computer Music Conference*, Hong Kong, 1996. `http://www.cs.uwa.edu.au/~leigh/Research/Papers/ICMC96.ps.Z`.

[22] A. Tewfik and M. Ali, "Enhanced wavelet based audio coder," in *Conference Record of the 27th Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, California, pp. 896–900, Nov. 1993. `ftp://oz.ee.umn.edu/pub/tewfik/1993/audio/audio_asilomar93.ps.Z`.

[23] M. Vetterli and J. Kovačević, *Wavelets and Subband Coding.* Signal Processing Series, Prentice Hall, 1995.

[24] M. Wickerhauser, "Acoustic signal compression with wavelet packets," in *Wavelets – A Tutorial in Theory and Applications* (C. Chui, ed.), pp. 679–700, Academic Press, 1992. `http://www.math.yale.edu/pub/wavelets/papers/acoustic.tex` and `ftp://wuarchive.wustl.edu/doc/techreports/wustl.edu/math/papers/acoustic.tar.gz`.

[25] M. Wickerhauser, *Adapted Wavelet Analysis from Theory to Software.* Wellesley, Massachusetts: A K Peters, 1994.