WAVELETS IN REAL-TIME DIGITAL AUDIO PROCESSING: A SOFTWARE FOR UNDERSTANDING WAVELETS IN APPLIED COMPUTER SCIENCE

Claudia Schremmer[1], Thomas Haenselmann[1], Florian Bömers[2]
[1]Department of Praktische Informatik IV, University of Mannheim, Germany
[1]{schremmer,haenselmann}@informatik.uni-mannheim.de
[2]florian@bome.com

ABSTRACT

Today increasing hardware performance and the development of faster and more efficient algorithms allow the implementation of signal processors even on PCs. This article discusses real-time processing of digital audio with wavelets and multiresolution analysis. We present a tool that allows the real-time application and modification of wavelet filters. Thus, it allows the user to directly hear the effects of parameter settings. Simultaneously, it visualizes the time domain of the audio signal as well as the time-scale domain of the multiresolution.

Keywords. Real-time Digital Audio, Wavelet.

1 INTRODUCTION

The human perception of audio is still not totally understood. Many models exist, mostly agreeing that people have a range of audible frequency from about 20Hz to 20KHz. As human perception is the benchmark for digital audio, it is still difficult to judge parameter settings automatically. The best results are obtained when parameters are set manually, according to the specific data and to the specific purpose.

Today, processing of real-time digital audio on personal computers is becoming more and more common, making it affordable even for amateurs and small studios to work in the digital domain. Real-time audio processing allows modified audio to be "judged by hearing" while it is processed. Although needing much CPU power, real-time processing significantly improves digital audio: only when the effect of a changed parameter (e.g. volume, denoiser of an audio track) can be subjectively judged instantly, a best parameter setting can be singled out.

As it seems to be common that human perception of different signals (audio, image, video) relies more on the variation than on the absolute signal [7], signals are transformed into a frequency domain to analyze and extract or reject specific frequencies. This was traditionally done by the Fourier Transform. As sines and cosines are infinite functions, they have the disadvantage that they can hardly extract local features. The windowed Fourier transform promised help. But again, it is not ideal for signal analysis as the size of the window remains constant over time and frequency. The wavelet theory finally provides a huge range of analyzing functions that maintain fixed volume, but that change their size over time and scale. This feature leads to the notion of multiscale resolution and tags the wavelets to be the right tool for signal analysis.

After discussing some related work in Section 2, we present the basic concepts of digital audio, wavelet theory, and filter banks in Section 3. In Section 4, we present a framework for digital audio processing in real-time. With its real-time aspect, it allows to directly judge the effect of a parameter setting. Section 5 gives an outlook.

2 RELATED WORK

Applications of wavelet theory in signal processing have evolved in the 1980s and have seen a big boost in the mid-1990s. Wavelets have found their way into compression [10] and segmentation [8] as well as into automated quality judgement of processed data [2]. Some attempts to use wavelets for content analysis [13] and for noise reduction [3][9] have been made. An unusual project has been carried out at Yale School of music [1]: Wavelet analysis has been used to restore a battered recording of Brahms playing his own work in 1889. The original wax recording is one of the rare opportunities to analyze a major composer's interpretation of his own work. The copies of the wax cylinder were of such poor quality though, that no meaningful music was to be heard. Even if the wavelet-analyzed sound is not yet musically pleasant, enough of the recording has been reconstructed to prove that Brahms took considerable liberties with his own score. Contrary to our approach, none of the mentioned implementations focuses on real-time aspects.

3 BASIC CONCEPTS

3.1 Analog and Digital Audio

Sounds, as the ear can hear them, are small changes in air pressure, which stimulate the eardrum. In analog audio systems, these changes are captured and transformed into levels of electrical voltage which has a continuous range of amplitude levels. A computer can only handle *discrete* signals. The conversion analog-discrete is done by *sampling*: in short intervals the level of the current is measured (e.g. every 1/44100 second). The number of intervals per time is called *sample rate*. Once a sound has been sampled, its representation is a sequence of discrete values. These digital signals can be stored, copied, and processed without loss of quality.

3.2 The WT for Processing Real-Time Musical Signals

The Wavelet Transform (WT) provides some features especially useful for processing (digital) musical signals. Its multiresolution decomposition offers high temporal localization for high frequencies while offering high frequency resolution for low frequencies [5]. A high frequency event (e.g. a cymbal crash) will be analyzed by many "fast", i.e. short-time, and high frequency wavelets. Low notes will be analyzed by "slow", i.e. long-time, low frequency wavelets. Generally, this fails with Fourier analysis. As a wavelet is a compact function that vanishes outside a certain interval, the WT is specially adapted to analyze local variations "per constructionem", and is thus specially adapted to audio analysis. Furthermore, the logarithmic decomposition of the frequency bands of dyadic Wavelet Transforms resembles human perception of frequencies [7], and thus modifications in the wavelet's scale domain are similar to the corresponding (subjective) changes in human perception.
Parameters that influence every wavelet analysis of signals include (a) choice of the analyzing wavelet or the corresponding filter, (b) choice of the transform (continuous, dyadic, packet,...) and (c) choice of depth of the decomposition, i.e. how many levels (scales) are calculated.

3.3 Wavelet Transform and Filter Banks

In signal processing filters denote an algorithm or device

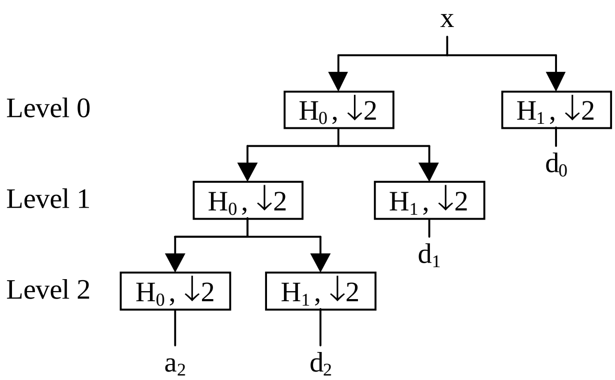

Figure 1: Wavelet filter bank for analysis of signal x, implemented via successive application of a 2-channel filter bank. H_0 denotes a low-pass filter, and H_1 denotes a high-pass filter. The symbol $\downarrow 2$ indicates the downsampling process. a_i and d_i stand for the approximations and the details of the different levels.

that alters selected frequencies of the signal. Two special filter types are *low-pass* and *high-pass* filters. Low-pass filters let all frequencies pass that are below a cut-off frequency, whereas the remaining frequency components are removed from the signal. High-pass filters work vice versa. The wavelet theory of multiresolution is implemented via high-pass filters (~ wavelets) and low-pass filters (~ scaling functions) [11][12].
The decomposition of a signal can be realized in a filter bank via parallel application of the appropriate filters, or via successive application of a 2-channel filter bank (cf. Figure 1) of high-pass and low-pass filters. The wavelet theory is represented by such a filter bank coding as outlined in Figure 1: the detail coefficients of every recursion step are kept apart, and the recursion starts again with the remaining approximation coefficients of the transform.
The synthesis of a filter bank works just the other way round: The detail and approximation coefficients of each level are upsampled and filtered with synthesis filters. The sum gives the approximation coefficients for the next level. The process is repeated until the original signal x is being reconstructed.
Concerning the implementation, the discrete WT is of (theoretical) complexity O(n). Because of boundary problems at the edges of the digital signals, computation is in reality more time-consuming than computing the FFT [14][4]. However, it is fast enough for real-time analysis and synthesis of audio data.
The choice of different wavelets and thus different filters has impact on how many coefficients are non-zero within the scale domain. Concentration of a signal's energy on some few non-zero coefficients, i.e. high number of vanishing moments, is preferred for audio filters. However, in general more vanishing moments and smaller transition bands between pass band and reject band lead to longer filters [6]. As the length of a filter directly affects computation time of analysis and synthesis, shorter filters are preferred for computational reasons. A compromise of filter length has thus to be found.

3.4 Wavelet Applications in Signal Processing

Wavelet *compression* is reached when part or all of the wavelet coefficients are small, so that they can be thrown away. Wavelet *analysis* asks for the specific content of the coefficients at the different levels in order to locate specific contents within the scale domain. Let us discuss the detection of white noise: White noise is a very regular, perturbing signal in all frequency bands (noise in radio broadcast, noise in TV scenes) that might have been added to a signal during its transmission. As it is located in all scales, it is relatively difficult to detect and eliminate. The solution is a comparison of coefficients in all different scales: the coefficients in scales that seem to contain no other information are cut off under a certain threshold.

4 THE WAVELET AUDIO TOOL

It is still most difficult to automatically determine a best choice of the analyzing wavelet, the kind of transform, and the depth of decomposition (cf. Section 3.2) automatically for arbitrary signals and for arbitrary purposes.

As after all, subjective judgement of a processed signal is still the most reliable one, we have developed a flexible tool for real-time digital audio processing [15] with the following main purposes:

- subjective judgement of the chosen settings by "hearing" the wavelet filters,
- "seeing" the wavelets and the effect of multiresolution analysis,
- becoming interested in wavelet analysis by "playing around" with the tool, and
- teaching/understanding wavelets.

A framework for digital audio processing has been developed that is implemented in C++ [3][15]. The framework's core classes provide the interfaces and some implemented classes. All sources, destinations and modifiers of the data flow are implemented as independent extensions. This allows to easily extend the audio framework by new extensions. The extensions implemented so far include filters to read audio from soundcard/file and to write the processed audio to soundcard/file.
A graphical user interface (gui) has been developed for the Windows platform (cf. Figure 2). It does not use all of the framework's possibilities: a simple chain with 2 readers and 2 writers is used. Any number of filters can be put in-between. The flow of audio data is symbolized with arrows. With radio buttons, the sources and destinations can be activated. In Figure 2, the setting is as follows: A difference listener is started before a noise generator is added. The forward Wavelet Transform paves the way for a wavelet denoiser. The display of the coefficients in the time-scale domain (cf. Figure 4) is added before the Wavelet Transform is reversed and the difference listener ends.

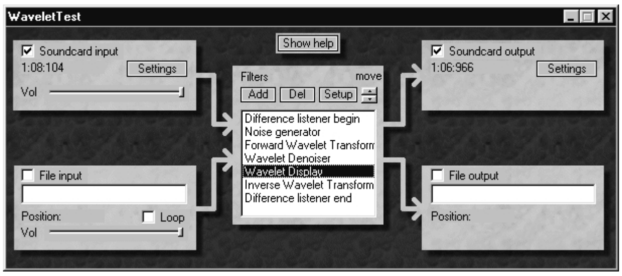

Figure 2: Graphical user interface for the wavelet audio tool. Audio data can be read from soundcard/file and be written to soundcard/file. A number of filters is applied in-between: The filters themselves are configurable, and their order can be varied.

The choice and the order of the filters is very flexible: The user can add implemented filters, and the dialog box in Figure 3 opens. For each selected functionality, a window pops up (cf. also Figure 5) where the parameters can be set. The filters are applied to the input audio stream in the order shown in Figure 2, from top to bottom. The order of the selected filters can be adjusted (cf. Figure 2: The "Wavelet Display" is marked) by moving them up or down. The results of all actions are directly sent to the soundcard/output file.

In our example setting, the entire set of actions is surrounded by the difference listener. This allows to hear the difference between original and denoised signal. If denoising was perfect, the difference signal would be complete silence. Any existing sound in the difference signal corresponds to side-effects of modifications done by the denoising filter. The presented setup allows finding the right parameters for the denoiser efficiently. A slightly different setup, with the "Difference Listener Begin" applied *after* the noise generator, would allow listening to the noise that has been removed by the denoiser.

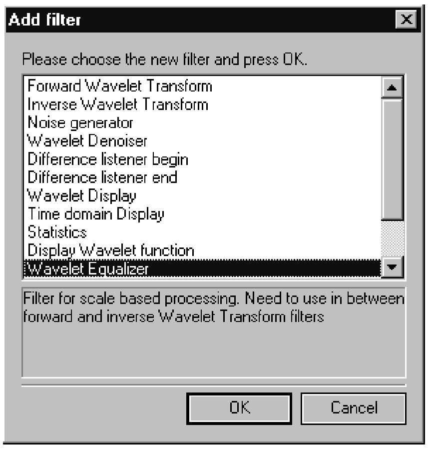

Figure 3: Add filter dialog box.

For the forward and inverse Wavelet Transform, all standard wavelet filters are implemented (Daubechies, Coiflets, Symlets, Biorthogonal, Battle-Lemarié, Spline). The choice of handling the (mathematically undefined) boundary problem for the audio signal is also let to the user: he has the choice between zero padding, mirror padding, padding with history, and wrapping.

Figure 4: The filter to display the wavelet domain visualizes the multiscale analysis of the wavelet decomposition. The higher the wavelet coefficient in a scale is, the darker the area. The x-axis denotes the time within the audio playback.

Another implemented functionality of the tool is a display for the multiscale analysis (cf. Figure 4): A window is created that displays all wavelet coefficients in the time vs. scale domain. Here, the amplitude of a coefficient is represented by colors. As this is done in real-time, every modification of the wavelet coefficients can directly be monitored. In our example setting of Figure 2, the effect of different denoising parameters can be followed visually in Figure 4.

The visualization of the chosen wavelet function ψ for forward and inverse Wavelet Transform is reached by a simple trick: setting all wavelet coefficients to zero, except one single coefficient will result in the time-domain wavelet function created by the inverse transform. This filter allows controlling the scale, temporal position and amplitude of the single non-zero coefficient (cf. Figure 5). By adding a time domain display filter after the inverse WT, the wavelet function can be seen and explored. Changing scale and time shows the effects of dilation and translation (cf. Figure 6). This is an effective means to better understand wavelets and their implementation as filter banks.

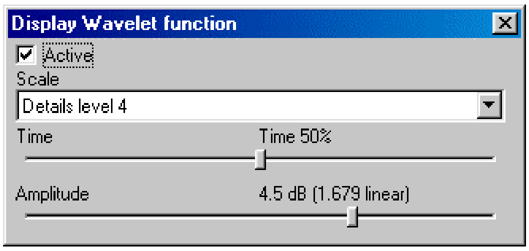

Figure 5: Filter to set all but one wavelet scale parameter to zero. In this screenshot, only the coefficients of level 4 will remain.

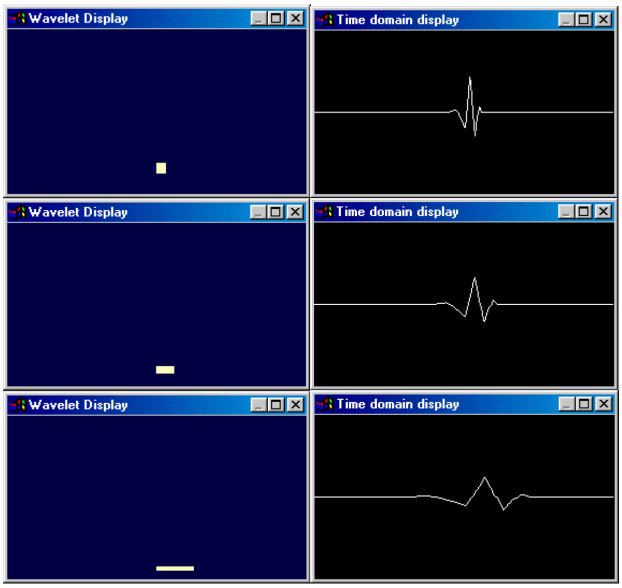

Figure 6: With the dialog box shown in Figure 5, wavelet coefficients of level 4, 5, and 6 have been selected accordingly. The inverse Wavelet Transform thus results in the display of the wavelets of the different scales.

5 CONCLUSION

We have presented the basic concepts of digital audio processing and why we have chosen the Wavelet Transform for real-time audio applications. The tool that we have presented in Section 4 [15] is outstanding to help understanding the concepts presented in Section 3. It is flexible in adding, removing, or changing the order of a lot of wavelet filters that are applied to an audio stream. Its only application at the moment is noise reduction. We have begun developing tools for audio content analysis. As analysis of the scale domain is much more powerful if one has the possibility to pick out specific scales of interest, this approach is realized by the implementation of wavelet packet analysis. A precise application of audio content analysis will be realized as phoneme detection: an audio stream of (spoken) audio will be decomposed automatically into its different phonemes.

6 REFERENCES

[1] Jonathan Berger, Charles Nichols. *Brahms at the Piano*. Leonardo Mus. Journal, 4: 23-30, 1994.

[2] Franziska Bock. *Analyse und Qualitätsbeurteilung digitaler Bilder unter Verwendung von Wavelet-Methoden*. Phd Thesis, Shaker Verlag, Aachen, 1998 [German].

[3] Florian Bömers. *Wavelets in Real-Time Digital Audio Processing: Analysis and Sample Implementations.* Master Thesis. University of Mannheim, May 2000.

[4] Corey Cheng. *Wavelet Signal Processing of Digital Audio with Applications in Electro-Acoustic Music.* Master Thesis, Hanover, New Hampshire. http://www.eecs.umich.edu/~coreyc/thesis/thesis-_html, 1996.

[5] Don Cross. *Time domain filtering techniques for digital audio.* http://www.intersrv.com/~dcross/-timefilt.html, 1998.

[6] Ingrid Daubechies. *Ten Lectures on Wavelets.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 1992.

[7] E. Bruce Goldstein. *Sensation and Perception.* Wadsworth Publishing Company, CA, USA, 1989.

[8] Thomas Haenselmann, Claudia Schremmer, Wolf-gang Effelsberg. *Wavelet-based Semi-automatic Segmentation of Image Objects.* In Proc. Signal and Image Processing (SIP 2000), Las Vegas, USA, 2000.

[9] Maarten Jansen. Wavelet Thresholding and Noise reduction. Phd Thesis, Katholieke Universiteit Leuven, Belgium. http://www.cs.kuleuven.ac.be/-~maarten/publications/PhD/index.html, 2000.

[10] JPEG 2000 Part 1 Final Committee Draft Version 1.0. Project 1.29.25444. ISO/IEC JTC 1/SC 29/WG 1. March 2000.

[11] Alfred Karl Louis, Peter Maß and Andreas Rieder. *Wavelets*. B. G. Teubner, ISBN 3-519-12094-1. Stuttgart, 1998 [German].

[12] Stéphane Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, ISBN 0-12-466605-1. San Diego, CA, USA, 1998.

[13] Silvia Pfeiffer. *Information Retrieval from Digitized Audio Tracks of Films.* Phd Thesis, Shaker Verlag, Aachen, March 1999 [German].

[14] Curtis Roads et al.: *The Computer Music Tutorial.* The MIT Press, 1996.

[15] Claudia Schremmer. The Wavelet Tool. URL: http://www.informatik.uni-mannheim.de/~cschremm/-wavelets/index.html, 2000.