Pyramidal Algorithm for the Restoration of Audio Signal Corrupted by Wideband Noise.

Azaria Cohen[1], Itai Neoran[2]

[1] Waves, Tel Aviv, Israel.

azi@waves.com

[2]Waves, Tel Aviv, Israel.
itai@waves.com

ABSTRACT

Restoration of noisy audio recordings necessitates maximum suppression of noise with minimum degradation of program material. Spectral suppression methods perform best with high frequency resolution but deliver poor performance with transients. While wavelet-based algorithms attempt to mitigate the time-frequency trade-off, they suffer from frequency aliasing. The suggested pyramidal algorithm is a good candidate for resolving the time-frequency resolution trade-off while avoiding aliasing. In this study, an algorithm for removal of wideband noise from old audio recordings is evaluated. The algorithm is based on the pyramidal algorithm and on a spectral method for noise suppression. Results show enhanced conservation of onsets with efficient reduction of noise. The algorithm is implemented in real-time.

1. INTRODUCTION

Good restoration of noisy sound recordings means that the restoration process causes minimum degradation of program sound while achieving maximum suppression of noise. These criteria are subjective, as they depend strongly on our ability to hear and distinguish the sound and the noise. Therefore, by saying "minimum degradation of sound" we mean minimum human detectable degradation.

Coarse noise reduction is achieved by passing the signal through constant filters. Wideband noise reduction techniques include spectral methods, model based methods, transform methods and psycho-acoustic methods [1]. The most widely used method is the spectral method. The spectral method is based on the assumption that the music signal is composed of

compact energy concentrations in the time-frequency domain. Noise suppression is achieved with adaptive filters. The adaptive filters' coefficients estimation is usually done in the frequency domain, requiring estimation of the signal and the noise frequency spectra. If the noise and signal spectra are available, the optimal filtering method becomes a time varying Winner filter [1,2]. Oftentimes, noise statistics are unavailable and the noise power spectrum is estimated from silent periods. In such cases, restoration is often achieved through spectral subtraction and expanding methods [2]. This methods rely on the known statistical information about the noise and assume only that the wanted signal is uncorrelated with the additive noise. The noisy signal is divided into overlapped segments, after which the short-time Fourier transform of the noisy signal is multiplied by a gain function according to the noise power estimation. In this case, longer analysis segments result in higher frequency resolution, hence better signal-to-noise separation. On the other hand, using long segments on a short onset causes smearing of the energy in the frequency domain. Therefore, relatively short onsets may result in comparatively small Fourier coefficients. These coefficients are too close to the level of the noise's Fourier coefficients and may be eliminated in the noise filtering process.
Moreover, processing in the frequency domain using overlapped segments results in smearing of transients in the time domain.

The overlap frame size is subject to two contradictory design principles: on one hand, the frame size should be long in order to provide good frequency resolution; on the other hand, the frame size should be small to conserve short events.

Restoration using wavelet is a good candidate for overcome this time-frequency resolution contradiction [4]. The wavelet filter bank decomposes the noisy signal into sub-bands in which the noise reduction is carried out. However, noise suppression using wavelet algorithms suffers severe distortions due to the aliasing phenomenon which yields spurious, short-lived, isolated frequency components. These aliased components shuffle some isolated, noisy frequency components into a more audible frequency area, where unpleasant metallic residual noise may be perceived. The aliasing problem is the result of critically-sampled sub-bands. If we loosen this restriction, we can gain multi-resolution without aliasing.
The pyramidal algorithm provides good time-frequency resolution while avoiding the annoying aliasing products of the wavelet process [5]. The pyramidal algorithm mimics the human hearing system in that it provides different frequency resolution at different frequency bands.

Nevertheless, there is a lack of study concerning, wideband noise audio restoration that uses the pyramidal algorithm. This study discusses the main design principles and performance of the pyramidal algorithm.

2. THE PYRAMIDAL ALGORITHM

2.1 Basic Pyramid Unit

The sub-band processing of audio signals is used primarily in source coding applications for efficient transmission and storage. The basis for the sub-band decomposition is a critically sampled filter bank [4]. These filter banks allow a perfect reconstruction of the input, provided there is no processing within the sub-bands. The wavelet algorithm consists of an analysis filter bank for decomposition of the signal in critically sampled sub-bands and a synthesis filter bank for reconstructing the broadband signal output. Using a critically sampled filter bank as a multiband equalizer, the processing in the sub-bands would lead to aliasing in the output. In order to avoid aliasing, a multi-complimentary filter bank [5] which we call the "pyramidal algorithm" is presented. The pyramidal algorithm enables alias-free processing in the sub-bands and leads to perfect reconstruction in the output. The basic unit in the pyramidal algorithm is the two-band complementary filter bank, as depicted in Figure 1. The input signal is low pass filtered (LPF) with a finite response filter, serving as an anti-aliasing filter. The low passed signal is downsampled to yield the low sub-band. The low sub-band is upsampled and interpolated by an identical low pass filter. The interpolated signal is then subtracted from the delayed input signal. Thus, we obtain a decimated low frequency sub-band and a residual high frequency sub-band. From each of the sub-bands, the noise is filtered out. The processed low frequency sub-band is upsampled and interpolated. The two processed sub-bands are then combined to produce the broadband output signal. A delay buffer compensates for the delay of the low pass filters. The scheme depicted in Figure 1 can be applied recursively. The open circles on the right of the pyramid unit denote the option to cascade another pyramid unit.

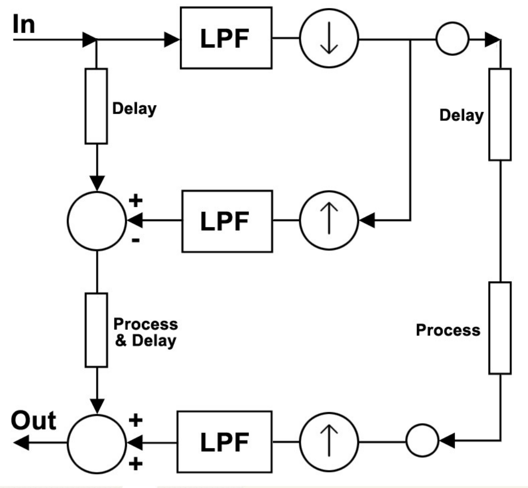

Figure.1: A pyramid unit.
The pyramid unit consists of a two-band complementary filter bank. The pyramidal algorithm is a cascade of pyramid units.

2.2 The Complete Pyramid

Pyramid units are indeed cascaded to provide multiband processing. The high frequency band of each level is filtered, while the decimated low frequency band is left for the next cascaded level to process. We call this structure of cascaded pyramid units a "pyramid". Each pyramid unit forms a "level" in the pyramid. The sampling frequency of each level is half that of the predecessor. Therefore, we obtain a multi-scale decomposition of the input signal. The open circles on the right of Figure 1 mark the unit's connection points in the cascade structure.

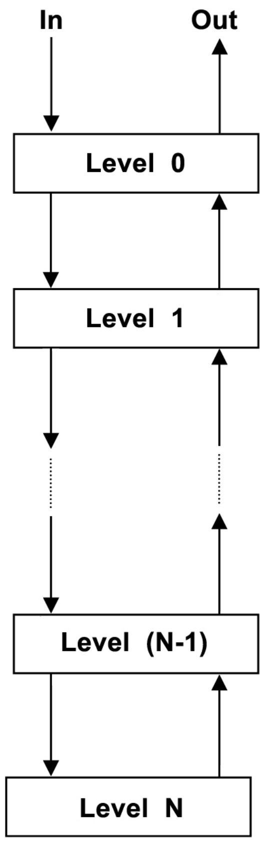

Figure 2: A chain of pyramid units. The parent pyramid unit transfers the downsampled, low frequency sub-band to the child pyramid unit. After processing, the low frequency sub-band is returned back to the parent unit, where it combines with the high frequency sub-band. The recursion is ended with a degenerate pyramid unit that contains only the noise suppression process.

The last level in the pyramid is a degenerate one which contains only filtering. Figure 2 depicts such a pyramid. We used a four levels-deep pyramid structure. Therefore we created four sub-bands whose frequency ranges were approximately: [20kHz-10kHz], [10kHz-5kHz], [5kHz-2.5k], [2.5kHz-0Hz].

Figure 3 depicts the general scheme of the sub-bands' frequency ranges. Each band is processed through overlap-and-add by a spectral method. The processing segment size is level independent. Therefore, we achieve high frequency resolution for low frequencies and low resolution for high frequencies. The time resolution acts in the opposite manner. We assume that most of the onset energy is concentrated in the high frequencies. The noise removal process thus reveals less erosion.

The noise suppression itself involves filtering in the frequency domain of each sub-band. The filtering is preformed by a spectral method. The processed signal is overlapped and added to the individual sub-band output. We used a constant segment length, n, therefore each sub-band processed n samples at a time. Hence, the frequency resolution of the first level is $\Delta f_1 = fs/n$ where fs is the sample frequency. Due to the downsampling, the next level's frequency resolution is $\Delta f_2 = fs/[2n]$ and the k-th level's frequency resolution is $\Delta f_k = fs/[kn]$.

The time scale is an increasing function of the level number. The first level segment time duration is , n/fs ,while at the k-th level the time segment length is: kn/fs . Therefore, high frequency bands are less likely to smear onset energy than are the low frequency bands.

It is common to assume that onset energy is concentrated in the high frequency bands. As seen from the above example, the pyramidal algorithm is more suitable to conserve onsets than are conventional spectral methods. The frequency resolution for the low frequency bands is higher than in the high frequency bands. This is well in agreement with the human auditory system's frequency resolution, which suggests that we perceive better frequency resolution in the lower frequency range.

The interpolation filter is not necessarily identical to the anti-aliasing filter. We chose the same filters as a matter of simplicity. The interpolation filters in the subtraction and the summation paths must be identical in order to attain perfect reconstruction of the broadband output signal. The anti-aliasing and interpolation filters share the same constraints. They should pass only frequencies that are smaller than half of Nyquist frequency and their phase response must be linear. There are variety of design methods with which to construct these filters.

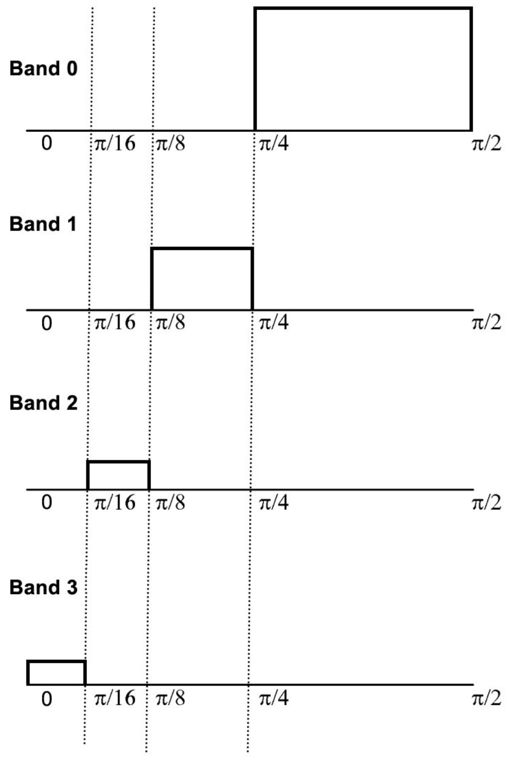

Figure 3: schematic description of the sub-bands' frequency ranges.

We designed the filter using a Chebyshev window with 131 coefficients. Figure 4 depicts the low pass frequency response. The cutoff frequency was prescribed to 0.44π . From Figure 4 it is clear that the stop band attenuation is more then $100[dB]$. The filter impulse response is symmetric, thus the filter is phase linear. Therefore, only a simple delay buffer is necessary to compensate for the filter's delay. The downsampling has a gain of 0.5, therefore an additional multiplication by 2 is needed at the interpolation filter.

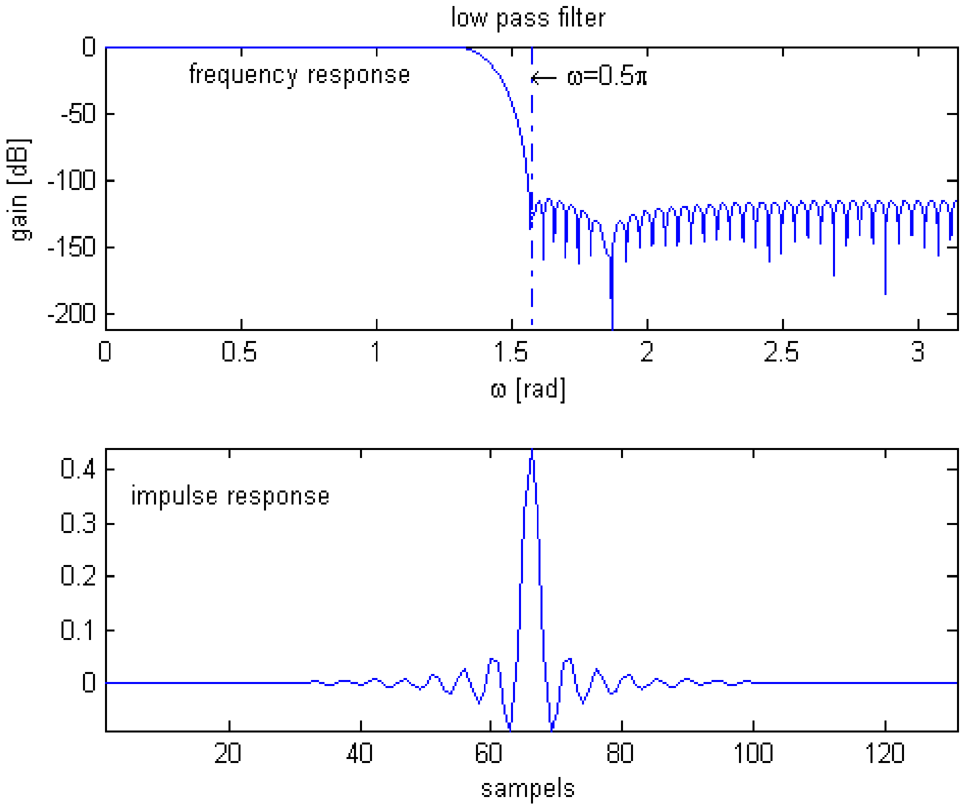

Figure 4: Anti-aliasing and interpolation filter's impulse response and frequency response magnitude. The phase response is linear. The filter is designed using a Chebyshev window.

There are two delay components in the pyramid unit as shown in Figure 1. One component is in the subtracting path, which yields the residual signal, the high frequency band. The second delay component is in the summing path, which yields the broadband output signal.

2. Delays

The low pass filter is a symmetric filter and its coefficient number equals $l = 2k+1$. Therefore its delay equals k samples. The process of downsampling and then upsampling does not change the overall delay.

We add another k delay samples due to the interpolation filter. Therefore, the subtraction path delay equals 2k samples. Comparison of the subtraction and summation path suggests that an additional process buffer is included in the summation path. This process buffer, where the noise suppression is preformed, is m samples long.

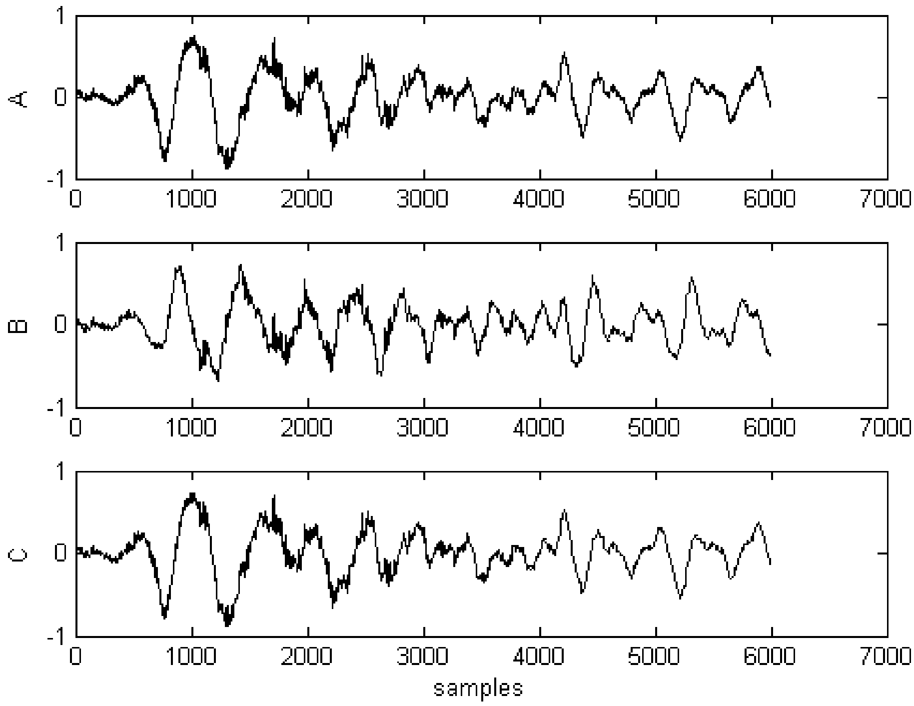

Figure 5 - A: a drum signal corrupted by noise. B: restored signal using a general spectral method. C: restored signal using the pyramidal algorithm.

The delay length is pyramid level dependent. The last pyramid level, the degenerate one, holds exactly m samples. These m samples are upsampled, making the total delay $2m$.

The total path delay is the sum of the process delay and the filters' delay. Therefore, the total delay is equal to $2m+2k$. The subtraction path has already been compensated by $2k$ samples in the delay block (Figure 1). Therefore, we have an additional $2m$ delay samples to be compensated. This delay compensation is done in the process and delay block (Figure 1).

We can compute all the delays in the pyramid in a recursive way. For example, the preceding level's process length is $2m+2k$. Therefore, this level reflects its preceding level by $2[2m+2k]+2k$ samples and so on.

Finally, the pyramid unit's delay becomes the elements of the following series: $m, 2m+2k, 2[2m+2k]+2k, \ldots$. The increasing growth speed of this series infers the pyramidal algorithm's most pronounced shortfall: this algorithm is a heavy memory consumer.

2.4 Noise Suppression

The noise reduction is carried out by filtering each of the sub-bands in the frequency domain. First, the estimated noise spectrum is acquired. Assume for example, that we can sample a time segment that contains only noise. Next we acquired its noise spectrum. The sub band's signal is processed by an short time overlapped FFT algorithm [6]. Each of the overlapped segments is windowed using a Hanning window, and Fourier transformed. The signal spectrum is then compared to its matched estimated noise spectrum. According to the signal to noise ratio, a gain function is constructed for each frequency bin, as in [6]. For example, assign r to the ratio between the signal and the noise. The gain is given by $g = min(r^{\alpha}, 1)$, where α is a real bigger than 1. The output segment is the result of multiplying the input segment by the gain g. An inverse Fourier transform, output windowing, and adding the output segment in an overlapped manner to the output complete the noise suppression process.

2.5 Real Time Implementation

The noise reduction process in each of the pyramid levels' sub-bands require considerable CPU and memory resources. The segmentation process on each of the bands results in a CPU peak occurring with a segment length periodicity. Synchronization of these peaks may produce a very high CPU peak even though the average CPU load is quite small. We spread the CPU peaks by desynchronizing the noise reduction operation between different levels. We used the extra memory resulting from the delay compensation to construct different delays for each of the levels in the pyramid. The result was to spread and therefore reduce the CPU peaks.

The overall latency is equal to the first pyramid levels' delay. If the pyramid has LV levels, then the total latency is described by : $[2^{LV}]m + [2^{(LV+1)} - 2]k$ where m is the process length in samples and k is half filter length. The CPU usage of the proposed pyramidal algorithm was approximately 10% for a Pentium-4 1.7GHz processor. It is possible to save latency by applying shorter filters and processes. Another possibility is to apply IIR filters and IIR noise reduction. The latter will decrease m and k.

3. RESULTS

The pyramidal noise reduction algorithm was implemented as a real-time audio plug-in. A comparison between the pyramidal algorithm and a general spectral algorithm performance was performed by both quantitative analysis and subjective listening test. In this paper we describe a test case of a recorded drum that has been corrupted by artificial additive white noise at -40dB. This audio file was restored by a spectral method and by the pyramidal algorithm .

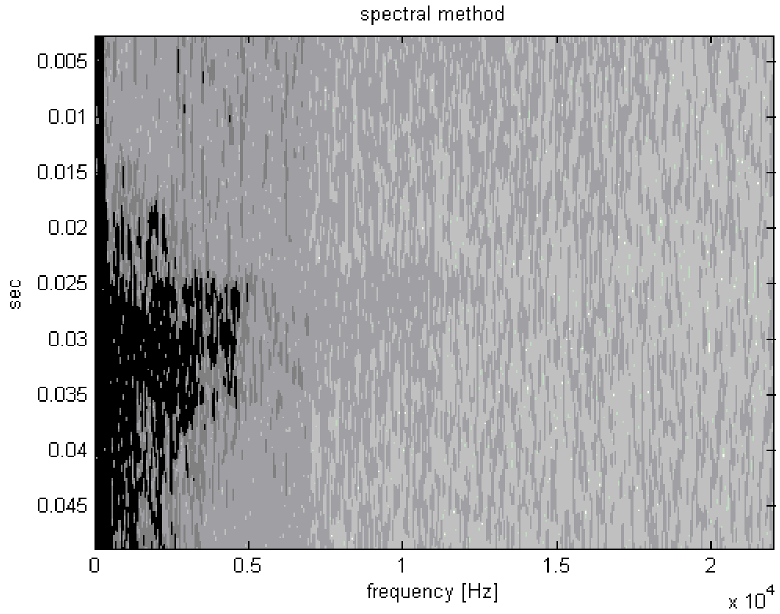

Figure 6: Spectrum of a restored signal using a general spectral method.

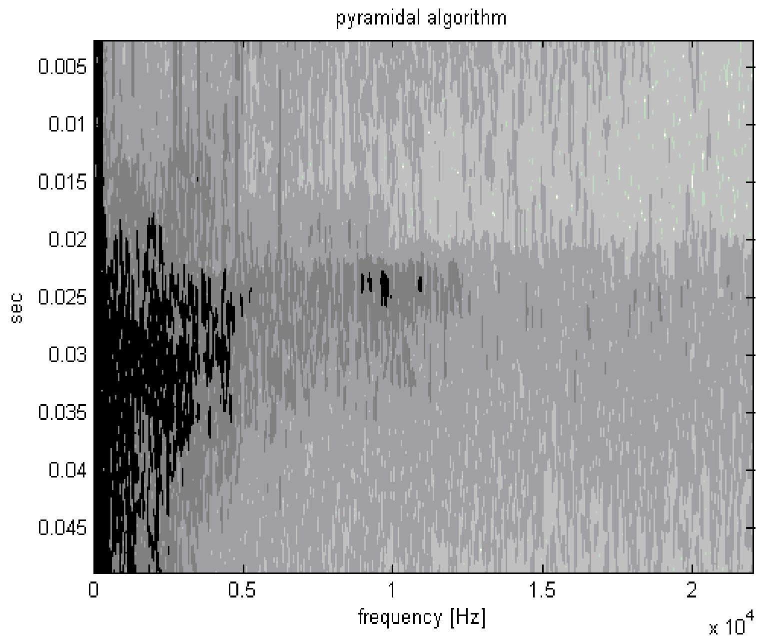

Figure 7: Spectrum of a restored signal using the pyramidal algorithm.

Figure 5 depicts a comparison between a spectral method and the pyramidal algorithm in the time domain. Figure 5A depicts a noisy recording of a drum. Figure 5B depicts the general spectral method restoration output and Figure 5C depicts the output of the pyramidal algorithm. It is quite obvious that the pyramidal algorithm conserved the onset better than the reference general spectral method.

A more complete understanding arises from a comparison in the frequency domain. Figure 6 depicts a spectrogram of a restored signal using a general spectral method and Figure 7 depicts the same audio signal restored using the pyramidal algorithm. Again, it is easy to notice that an extra energy concentration exists in the pyramidal algorithm spectrogram while in the spectral method this concentration is absent.

3. CONCLUSIONS

The pyramidal algorithm has shown better performance than known spectral methods. Its computation and memory requirements are, however, greater than the general spectral or wavelet methods. Today's technology enables us to implement the pyramidal algorithm in a real-time.

The pyramidal algorithm is a good candidate for solving other time-frequency problems in a way that requires a mimicking of the human auditory system.

4. ACKNOWLEDGEMENTS

We would like to thank Meir Shaashua from Waves for his great support, Dana Massie for his helpful advise, and all the colleagues from Waves who helped programming and testing of the audio plug-in.

REFERENCES

[1] Godsill S.J and Rayner J.W. Digital Audio Restoration a Statistical Model Springer Verlag ISBN 3540762221 . 1998

[2] Vaseghi S Advance Signal Processing and Digital Noise Reduction. Jointly Wiley and Sons and Teubner ISBN 0471958751. 1996.

[3] Mallat S. A wavelet tour of signal processing. Academic Press. ISBN 012466051. 1998

[4] Strang G and Nguyan T. Wavelet and filter bank. Wellesley Cambridge Press. ISBN 0961408871. 1997.

[5] Udo Zölzer. Digital Audio Signal Processing. John Wiley & Sons. ISBN: 0471972266. 1997.

[6] Udo Zölzer. DAFX Digital Audio Effects. John Wiley & Sons. ISBN: 0471490784. 2003.