Context Sensitive Harmonic Processor

Andre Luiz Luvizotto[1] **, César Rennó Costa** [1 2 *]

[1]Núcleo Interdisiplinar de Comunicação Sonora - NICS
Caixa Postal: 6166, Campinas, São Paulo, Brasil

[2]Laboratório de Bioinformática e Computação Bio-Inspirada - LBiC
Caixa Postal: 6166, Campinas, São Paulo, Brasil

andre@nics.unicamp.br, cesar@nics.unicamp.br

Abstract. *We present here a new-fashioned audio processor directed to perform harmonic distribution manipulation. It's a tool that operates over timbre, being sensitive to pitch context. The process is based on a wavelet-based filter-bank that is dynamically tuned to the note's fundamental pitch. As a single entry process it acts as harmonic equalizer while with two entries it works as harmonic controller.*

1. INTRODUCTION

Helmholtz [Helmholtz, 1863] on his classic "On the Sensations of Tone as a Physiological Basis for the Theory of Music" defined tone as a combination of a fundamental and countless overtones. Followed by other studies, its consensus that harmonic distribuiton has a primary role over timbre perception of musical sounds. Thus, on music sounds, variations on partials implies on variations over timbre.

We present here a processor focused on the timbre manipulation of musical interesting sounds by means of partials equalization. It's based on dynamic wavelet filter bank proposed in [Beltran and Beltran, 2003] controlled by a pitch detection algorithm. A preview version limited to this application has been presented in [Luvizotto and Costa, 2007].

The processor can also work as a harmonic controller of inharmonic sounds when using a harmonic material to control pitch detection and applying the filter bank over another sound. In this sense it actually would analog to an automatic and dynamic subtractive synthesizer.

The Wavelets transform is a method that combines decomposition in elementary contributions and hearing like properties [Kroland-Martinet, 1988]. Its main feature for our application is the possibility of fast filter design with few parameters. Also, we consider the pitch detection which can be a very complex task due large bandwidth and inharmonic partials [Jehan, 1997].

The paper is outlined as follow: in the first section we treat the main aspects about wavelets analysis and the mathematical background, including the formulation of the used scale and shift parameters as well as the chosen mother wavelet. In the following section we handle the pitch detection algorithm with an overview of the Harmonic Product Spectrum method. In the last two sections there will be an explanation of the implementation and also a discussion about the results and performance of our model.

[*]Supported by Fapesp.

2. WAVELET ANALYSIS AND FILTER BANK

Usually the process of partial extraction is based on *Short Time Fourier Transform*, but as we know by the Heisenberg uncertainty principle an ideal time localization gives rise to a non-ideal frequency localization and vice versa[Chui, 1997][Mallat, 1998].

Wavelet Analysis can be seen as a powerful tool to fix this problem that provides a flexible time-frequency window and that is the main reason to choose wavelets to implement the filter-bank. In formulating a Continuous Wavelet Transform (CWT) a scale parameter is introduced to adjust the width of the sliding time window process. The mathematical basis of the Fourier Transform are sine waves whereas the basis of the CWT is a wavelet family generated by the mother wavelet ψ(t).

As proposed by Beltrán in [Beltran and Beltran, 2003] we are considering a complex generalization of the Morlet wavelet given by:

$$\psi(t) = C' e^{\frac{-t^2}{2}} \left(e^{j\omega_0 t} - e^{\frac{-\omega_0}{2}} \right) \tag{1}$$

for the mother wavelet.

This function as proposed by Kronland-Martinet et al [Kroland-Martinet, 1988] requires small corrections to ensure that the admissibility condition for an analyzing wavelet is satisfied. However, in practice taking $w_0 > 5$ is enough [Beltran and Beltran, 2003]. In this case the Fourier transform of the Complex Morlet's wavelet is:

$$\hat{\psi}(\omega) = C e^{\frac{-(\omega-\omega_0)^2}{2}} \tag{2}$$

C' e C are normalization constants in the time and frequency domain, respectively.

As we are interested on a filter bank we need to have how to control the frequency resolution of each band. This way we can tune all the bands of our processor to the sound's partials of a given note. We should included a parameter k in the equation 2 to control the bandwidth of the filter. So we have in the frequency domain:

$$\hat{\psi}(\omega) = C e^{\frac{-(\omega-\omega_0)^2}{2k}} \tag{3}$$

The set of scale is employed to provide a logarithm-resolution on the frequency axis. We will also insert, in equation 3, a discrete factor s_i to slide the central frequency of the first band, i.e, changing this factor we will be sliding the first band and generating all the others. To prevent aliasing problems the filter bank generates first the bands with highest frequencies. That is, it gets the fundamental pitch f_f of the played note, calculates the last frequency band that will be used and then calculates the central frequencies ω_c of the remaining bands changing s_i as follow:

$$\omega_{c_1} = f_f N \tag{4}$$

and

$$\omega_0 = s_{min} \omega_{c_1} \tag{5}$$

We will call s_1 of s_{min} cause it can be considered a tuning factor related to the Nyquist criterion which fine tunes ω_0 to the desire first (highest) band's central frequency ω_{c_1}.

And for $2 \leq i \leq N$ we have:

$$s_i = \frac{N}{n_p} \tag{6}$$

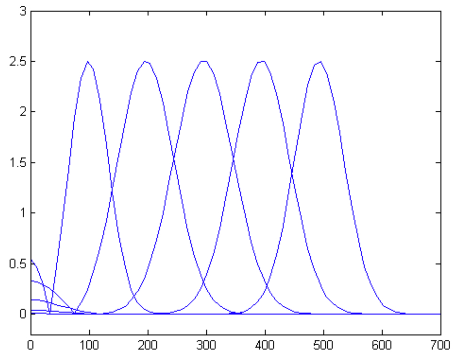

Figure 1: Filter bank generated by the mother wavelet presented in equation 9 with $\omega_c = 100$Hz, $k = 22000$ and s_{min} = 6.91 and s_i given by equations 6.

$$\omega_c = \frac{\omega_0}{s_i} \tag{7}$$

N is the number of bands and n_p is the band's number of the i-*th* frequency that is being calculated. We can see in the equation 7 the relation between the central frequency ω_c and the frequency ω_0 for all bands. Then we have in the frequency domain the follow equation:

$$\hat{\psi}(\omega) = C_s e^{\frac{-(s_i\omega - \omega_0)^2}{2k}} \tag{8}$$

where C_s is also a normalization constant. We have in equation 8 the main expression for the filter bank in the frequency domain. We need now to find its *time* representation to be able to use the convolution representation to filter the input signal. The follow expression was obtained from the equation 1, taking into account properties of Fourier transforms and by changing the continuous factor a for the discrete one s:

$$\psi(t, s) = \frac{\sqrt{k}}{s} C' e^{\frac{-kt^2}{2s^2}} \left(e^{\frac{j\omega_0 t}{s}} - e^{\frac{-\omega_0}{2\sqrt{k}}} \right) \tag{9}$$

The equation 9 presents the mother wavelet that will be used with the frequency slide parameter s and bandwidth parameter k.

Now we'll expose the pitch detector algorithm.

3. PITCH DETECTION

Pitch detection algorithms can be classified in two separate categories, time-domain based on period detection and spectral-domain. The first kind of detection seems to be the most straightforward idea, which consists in looking to the input signal as an amplitude fluctuations in the time domain and try to find repeating patterns from the waveform that could lead us to its periodicity. However due to its limitations we will use in this paper the second category of pitch detection algorithms. The chosen method is *Harmonic Product Spectrum*, or HPS.

If the input signal is a note, with a well defined pitch, then its spectrum should consist of a series of peaks, corresponding to harmonic components that are integer multiples of the fundamental frequency. The HPS algorithm measures the maximum coincidence for these harmonics [de la Cuadra et al., 2001].

If we compress the spectrum a number of times (downsampling), and compare it with the original spectrum, we can see that the strongest harmonic peaks line up. The first peak in the original spectrum coincides with the second peak in the spectrum compressed

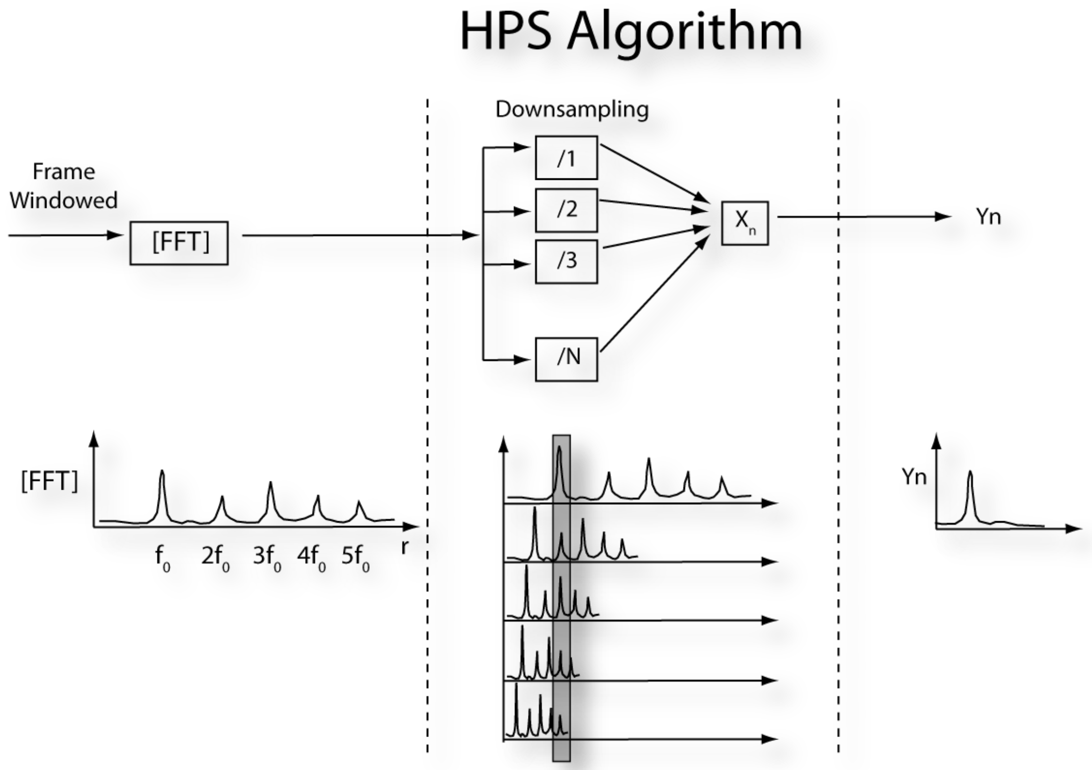

Figure 2: Overview of the HPS algorithm.

by a factor of two, which coincides with the third peak in the spectrum compressed by a factor of three. Hence, when the various spectrums are multiplied together, the result will form clear peak at the fundamental frequency. An comprehensive overview of this method can be found in [Galembo and Askenfelt, 1994]. Figure 2 demonstrates the HPS algorithm graphically.

4. IMPLEMENTATION

The algorithm was implemented with *Matlab* according to the diagram of the figure 3:

The input signal first reaches the pitch detector (that we call by Dr. Pitch) which outputs the fundamental frequency's value f_f to the filtering stage. Then the filter parameters ω_0 and s_i are calculated and inserted into the equation 9. As we know each band of the filter is related to a wavelet with bandwidth factor k and central frequency ω_c which is sampled with a sampling frequency f_s. As a results we have a matrix $M_{m\times n}$ where m is the number of sampled points and n the number of filter bands N.

In practice M is left-side multiplied by a weight's matrix $P_{1\times n}$ that has the informations about the gain of each band, i.e, values between 0 and 1 came from the processor' slides. At last this resulted matrix is time convoluted with the input sound vector providing the filtered output. When the next signal comes Dr. Pitch sends the new f_f to the filter stage. The parameters are actualized to generate the up-to-date wavelets and go on like before.

We used a Hanning window with 1024 and also 2048 samples with overlap of 50%. The window size depends on the lowest frequency (large period) of the input signal. We are looking for other methods of pitch detection that could be more efficient and computationally better.

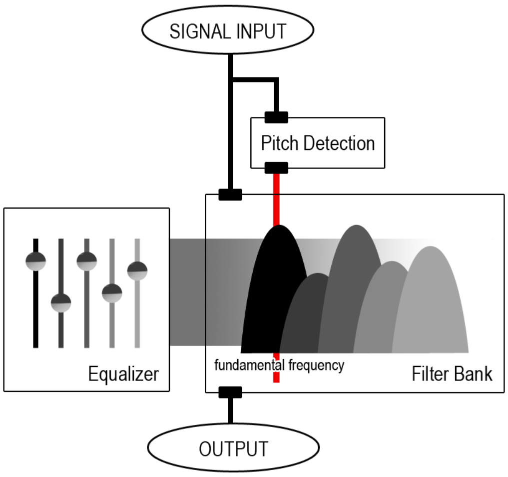

Figure 3: Implementation Block Diagram

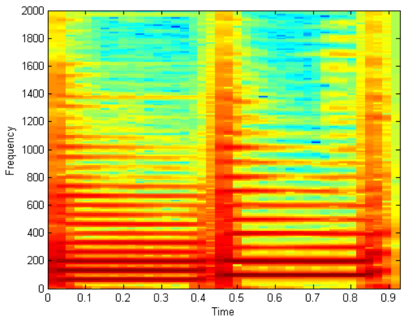

Figure 4: Sonogram of the input signal.

5. RESULTS

We applied the proposed processor with ten and twelve bands to many different sounds. Guitar, bass, percussion instruments, some different guitar tunings, for example, with the 6th string tuned to C1 since we were interested in evaluate how Dr. Pitch would work at extreme conditions. These examples can be find at www.nics.unicamp.br/∼andre/processor.

We can see on the figure 4 that Dr. Pitch worked very well, tracking both of notes correctly. We can also observe the difference between the spectrograms of figures 4 and 5 considering the number of partials presented. Only the first five partials were letting pass in figure 5 while all others five were filtered.

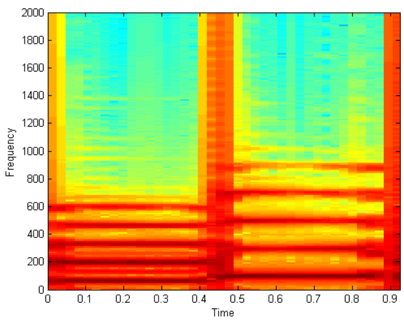

Figure 5: Sonogram of the output signal.

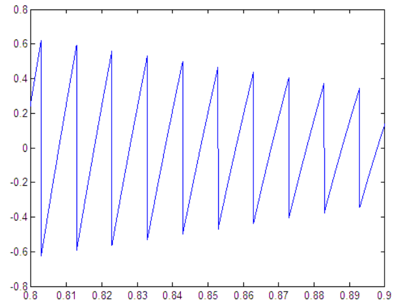

Figure 6: Original sawtooth wave.

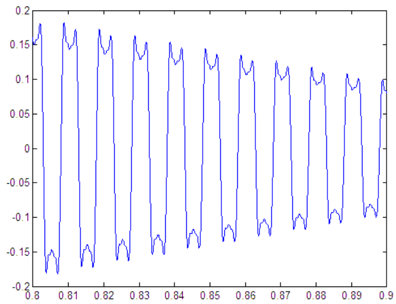

Figure 7: Filtered sawtooth wave. Notice that it is closed to a square waveform.

As a second example we have a sawtooth wave on figure 6 where its first four even harmonics were filtered. We can see on figure 7 that the resulted waveform is closed to a square wave, as expected. The more we increase the number of the filtered partials the more waveform matches to the square waveform.

Finally a square wave, figure 8, were filtered letting pass just the fundamental frequency. As we can observe on figure 9 the resulted waveform is closed to a simple sine.

6. CONCLUSION AND FURTHER IMPLEMENTATIONS

We presented the first results of a new-fashioned sound processor. A method of partials extraction was proposed as well as a pitch detector based on the autocorrelation function. The essential mathematical tools was revised in the first sections and a briefly discussion of the pitch detection in the section three.

This first prototype made with *Matlab* could give us a good idea of the sonic

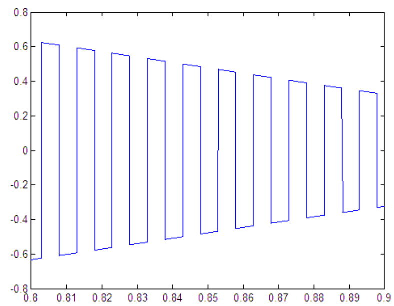

Figure 8: Original square wave.

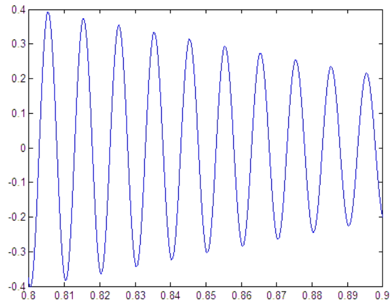

Figure 9: Filtered square wave. Notice that it is closed to a simple sine.

potential of the model. The best results were obtained in low frequencies sounds for example with bass drums, bass and lower guitar sounds.

We are currently developing a low latency C++ implementation to be used as a cross-platform plugin. We are also verifying the possibilities of working with other pitch detection methods, for example a wavelet based algorithm, and use it as another parameter on the graphical interface that could be selected by the user depending which kind of sound it wants to filter for example bass, brass, clarinet, piano etc.

7. ACKNOWLEDGMENT

We would like to thank Professor Adolfo Maia Jr. for helping us with the mathematical revision, formalization and kindly tips.

References

Beltran, J. R. and Beltran, F. (2003). Additive synthesis based on the continuous wavelet transform: A sinusoidal plus transient model. In *Proc. of the 6th Int. Conference on Digital Audio Effects*.

Chui, C. K. (1997). *Wavelets: a mathematical tool for signal processing*. Society for Industrial and Applied Mathematics.

de la Cuadra, P., Master, A., and Sapp, C. (2001). Efficient pitch detection techniques for interactive music. *Proceedings of ICMC*.

Galembo, A. and Askenfelt, A. (1994). Measuring inharmonicity through pitch extraction. *STL-QPSR*, 35(1):135–144.

Helmholtz, H. V. (1863). *On the Sensations of Tone as a Physiological basis of the Theory of Music*. Dover Publications.

Jehan, T. (1997). *Musical Signal Parameter Estimation*. PhD thesis, Université de Rennes 1.

Kroland-Martinet, R. (1988). The wavelet transform for analysis, synthesis, and processing of speech and music sounds. *Computer Music Jornal*, 12(4):11–20.

Luvizotto, A. L. and Costa, C. R. (2007). Harmonic processor. In *Convenção Nacional da AES Brasil*.

Mallat, S. (1998). *A Wavelet Tour of Signal Processing*. Academic Press.