Pitch-Synchronous Multiresolution Analysis of Music Signals

César Alonso Abad

Master thesis submitted in partial fulfillment of the requirements for the degree:

Màster en Tecnologies de la Informació, la Comunicació i els Mitjans Audiovisuals

Supervisor: Dr. Xavier Serra

Department of Information and Communication Tecnologies
Universitat Pompeu Fabra
Spain
September, 2007

Abstract

In this thesis a novel multiresolution approach for note detection in a polyphonic mix is proposed. The idea is to use a set of wavelets whose lengths are adapted to the theoretical fundamental period of musical notes. Using the typical wavelet dyadic decomposition we can generate a set of wavelets that match the fundamental frequency (F0) of a given note in every octave. Therefore, using a set of 12 different wavelets, one per each semitone, we can represent the fundamental frequency of every note in every octave using one wavelet scale per each octave. The magnitude and phase continuity of wavelet coefficients across temporal frames is exploited to draw a special kind of spectrogram, namely Pitch-Synchronous Wavelet Spectrogram (PSWS). When the corresponding F0 and harmonics of a note are present in the signal, a special DC pattern appears in the PSWS, due to the aforementioned continuity. Any other harmonic signal or noise produces pseudo-periodic or random AC patterns. This way, by filtering the AC components, we can identify the DC patterns in the PSWS and state the presence of a given musical note at some moment in time, even if the signal is polyphonic. For the moment, the method only works satisfactorily when the harmonic peaks in the music signal are close to the theoretical position of the frequencies of the musical notes. Some techniques are suggested in order to improve the system and extend it to non-stable pitch musical instruments.

Acknowledgements

First of all, I would like to thank my tutor Xavier Serra for giving me the opportunity to join the Music Technology Group, and also to work as teaching assistant in the subject "Speech Processing" at the Pompeu Fabra University. Both experiences have been invaluable for me, and it has been a pleasure to learn and to be able to apply and transmit my knowledge in the exciting field of music technology.

Special thanks to Perfecto Herrera, my direct supervisor in this research work for his wise advises and comments about the organization and the contents of the thesis, specially in the introduction and the state of the art. Jordi Bonada and Emilia Gómez for their help and advises in the evaluation of the note detection system and the comparison with the HPCP approach.

Thanks also to all the people of the MTG, specially those working with me in Room 316, with which I had the opportunity to discuss ideas and learn from different points of view.

Finally, thanks to my classmates in the Master in Information, Communication and Audiovisual Media Technologies. Specially to Inês Salselas for being much more than a classmate and a colleague: a very good friend.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

The motivation for the elaboration of this thesis has to do with the time-frequency representation of harmonic signals using the Discrete Cosine Transform (DCT): one may expect that the evolution of the peaks with time follow more or less continuous curves, as it happens approximately in the Short-Time Fourier Transform (STFT). The DCT and the Modified Discrete Cosine Transform (MDCT) are mainly used for coding purposes because they provide a similar spectral representation as the modulus of the FFT (very interesting to compact the energy of harmonic signals in few coefficients, or to apply psychoacoustic models for perceptual coding) and also critical sampling and perfect reconstruction. But the strong correlation in time that harmonic signals present (which is exploited for looseless compression, for example) does not mean that consecutive time coefficients in a DCT (or MDCT) spectrogram present a similar correlation: on the contrary, the phase lags between consecutive time frames affect dramatically the magnitude and the signs of the DCT coefficients, even if the signal is harmonic or even purely periodic.

One way to achieve a smooth time evolution of DCT coefficients in the case of a harmonic signal is to make the analysis window length coincide with the fundamental period of the signal. In this case, the phase lags between consecutive frames are small, and the time evolution of the coefficients follows a smooth curve. But this only happens for low frequencies. For high frequencies, the minimum error in the pitch estimation to determine the length of the analysis window could mean hundreds of periods, thus making the phase vary in a practically unpredictable way from frame to frame.

Therefore, the only chance to accurately follow the phase variations of a harmonic signal from frame to frame, seems to use a window length that is adapted to the fundamental period of all the tones involved. This appears to be quite unpracticable a priori. But given the constraint that we are dealing with musical signals, two ideas encouraged me to think that the approach was

affordable:

1. The theoretical frequencies of the fundamentals and the harmonics of the musical notes are precisely defined and they are actually not so many.

2. The typical dyadic wavelet decomposition is such that it is possible to adapt the window length to the period of the musical notes in a very natural way: if the window length for the lowest scale coincides with the fundamental period of a musical note, the windows of the upper scales will coincide with the fundamental period of the same note in the upper octaves, since both the dyadic decomposition and the construction of the musical notes follow the same power of 2 law.

This means that using 12 different sets of wavelets, one per each semitone, we can cover all the theoretical fundamental frequencies of the musical notes, assigning just one coefficient to each one of them. This way, more coefficients (scales) would be spent to describe the low frequencies, and less for high frequencies, matching the nature of musical notes and also the resolution of our hearing system.

The next step was to try to apply these ideas to some problem in which multiresolution analysis could have some advantage against Fourier analysis of music signals. This issue is discussed in next section.

1.2 Problem statement and objectives

In this section we develop the ideas that motivated this thesis and suggest some problems related to music analysis in which they could be useful. The main hypothesis and objectives of the thesis are also presented as comments added in relation to the problems stated.

1.2.1 The need for multiresolution analysis

One of the main drawbacks in Fourier analysis is the trade-off between time and frequency resolution. Music signals in general are wide band, and present important frequency components in virtually all the audible spectra. But the musical notes, by its own nature, are multiresolution: the frequency range of a semitone can be a few tens of Hz at low frequencies, or several thousands of Hz at high frequencies. For this reason, we need much more resolution for low frequencies than for high frequencies in order to analyze music signals.

As the Discrete Fourier Transform (DFT) provides a constant frequency resolution analysis, long temporal windows are needed in order to achieve the necessary resolution for low frequencies. But the longer the time window, the worse the time resolution. The Short-Time Fourier Transform (STFT) is a very useful tool for the analysis of locally stationary harmonic signals. The magnitude of the STFT can be used for tracking the peaks of the partials of musical notes, taking advantage of the fact that the position in frequency and

the amplitude of these peaks vary slowly with time. But the same is not true with the phase information. Even if we choose the analysis window length to contain one or two fundamental periods of the signal being analyzed following a pitch-synchronous approach (like in PSOLA [3], for example), the phase will evolve following a smooth curve along time only for low frequencies, as explained in section 1.1. Several approaches have been developed for the improvement of the time-frequency representation of spectrograms using the phase information for the estimation of analysis and/or synthesis parameters. Some examples are [4, 5, 6].

What we propose in this thesis is the use of the dyadic wavelet decomposition to make a pitch-synchronous analysis of musical signals. The length of the analysis windows for the wavelet decomposition is chosen to coincide with the fundamental periods of musical notes. The idea is that with this multiresolution pitch-synchronous scheme we can take advantage of phase continuity across the spectrogram, for example for efficient note detection.

Wavelets and multiresolution analysis have been successfully applied to different areas related to signal processing. Also in sound processing and more specifically in speech and music processing. Yet in the 1980's, researchers realized that the particular nonlinear characteristics of both the music signals and the human auditory system could be better expressed using wavelets [7]. In particular, the resolution of the human ear has a logarithmic behavior, being better for low frequencies and worse for high ones. Musical sounds have also a logarithmic structure: the frequency range of an octave is wider as frequency increases, and so again less frequency resolution is needed for high frequency components. This suggests the use of wavelets for some problems previously studied using the Fourier Transform, like polyphonic music automatic transcription, source separation, music feature extraction, sinusoidal plus residual modeling, perceptual audio coding, etc.

Fourier analysis has been and is still very useful for signal analysis, synthesis and coding. Fourier base functions (sines and cosines) are very well suited for a large variety of signals. Especially for music sounds, which have in general a well-defined harmonic structure. The development of digital technology and particularly the discovering of a fast algorithm for DFT calculation (FFT) lead to an extensive use of Fourier Transform in every signal processing area. But there is an important drawback in Fourier Transform when applied to music signals: the time-frequency resolution is equal for every frequency.

As it has been said, both human auditory system and the nature of musical notes are multiresolution, that is, they have different properties at low and high frequencies. We may want to increase the frequency resolution at the expense of loosing some time resolution for low frequencies, and vice-versa: increase the temporal resolution for high frequencies at the expense of loosing some frequency resolution. This would match the nonlinear resolution of the auditory system and would match also the location of the fundamental frequency and harmonics of musical notes along the frequency axis.

Fortunately, there is a way to increase both frequency and time resolution in Fourier Transform: the use of the Short-Time Fourier Transform (STFT)

spectrogram, with suited windows of as many points as necessary to increase frequency resolution and as much overlapping percentage as necessary to increase temporal resolution.

But this solution has a very important drawback: overlapping means using each temporal sample several times in consecutive windows to build a detailed spectrogram. This has 3 direct consequences:

- There is an increment of computational load proportional to the overlapping factor.
- The signal is not anymore critically sampled (i.e.: the spectral representation uses more coefficients than the temporal representation).
- The increment of time resolution is not really true: even if the *hop size* (i.e. the separation in samples between consecutive windows) is small, each window can contain frequency information of a much bigger segment of signal. This means that frequency components that are really not present in a given segment of time can appear however in the spectrogram area corresponding to that segment of time, thus producing well-known artifacts, like the pre-echo effect in the SMS synthesis process [8].

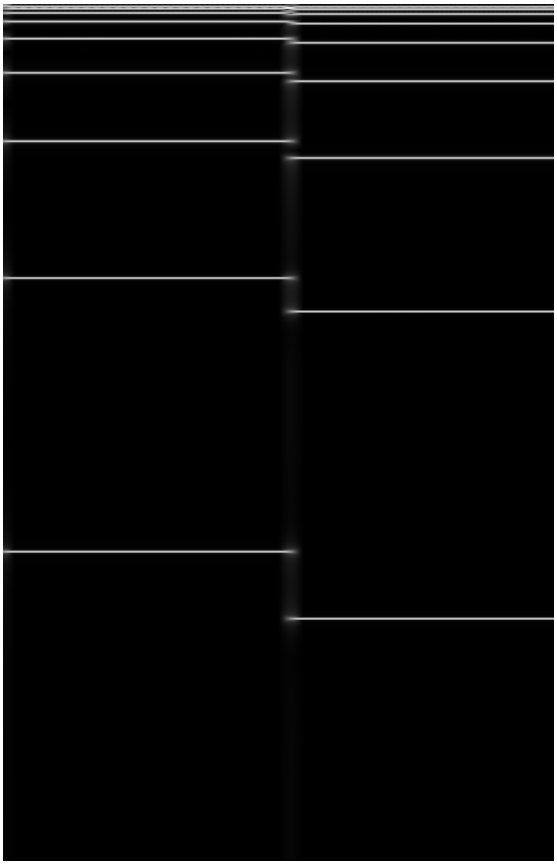
The first consequence is important for real time applications, or when there is any computation complexity constraint. The second one is important for synthesis and coding/compression purposes, for example. The third one is a serious problem when we try to make an accurate analysis or synthesis of the spectral components of a signal along time without adding false information or reconstruction artifacts. The time-frequency resolution trade-off in STFT is illustrated in Figure 1.1

Multiresolution tools (Quadrature Mirror Filter Banks [9], Constant-Q filters [10], Wavelet Transform [11]) provide a way to increase either time or frequency resolution depending on the characteristics of the signal subject to analysis. Wavelet analysis provides a great deal of flexibility to adapt the analysis tool to the actual problem considered. We can choose several features:

- The "shape" of the wavelet. We can choose the family of wavelets that best fits our problem, or even develop a new wavelet family.
- The time-frequency grid of the coefficients.
- Orthogonal, Biorthogonal or non-Orthogonal wavelets.
- Discrete Wavelet Transform (DWT) or Continuous Wavelet Transform (CWT).

Moreover, there are fast algorithms for the calculus of the DWT. Taking all this into account we may think that wavelets could help in the solution of the problems stated above, provided that the new time-frequency resolution trade-off is suited for the problem considered, which as explained before, is the case of musical signals analysis and their perception by humans.

(a) Window length: 128 samples; hop-size: 64 samples.



(b) Window length: 2048 samples; hop-size: 64 samples.

Figure 1.1: STFT spectrograms of a signal that consists of 9 octave-spaced tones. In the first half of the signal, this 9 tones correspond to the fundamental frequencies of the A note in every scale, while in the second half they correspond to the fundamental frequencies of the note B. Two different sets of analysis parameters are used for (a) and (b). In (a) the window length is not big enough to resolve the low frequency tones, while in (b) the exact point in time in which the signal changes cannot be clearly stated.

On the other hand, it is the intention of this master thesis to explore another point of view about the relationship between wavelets and music. That is the fact that the typical dyadic decomposition used in wavelets is actually very related to how musical notes are produced: if we double the frequency of a musical note, we get the same note but one octave higher, and vice-versa. Intuitively, we can adventure that by choosing a suitable set of analysis wavelets, the corresponding coefficients for each scale can give a measure of the projection of a musical signal in all the octaves of a note.

1.2.2 Exploiting *a priori* music knowledge

When dealing with music signals, we can take advantage of a lot of *a priori* knowledge: low level features like periodicities, frequency and time correlations, etc. and also higher level features like rhythm, tonality, etc. If we want to analyze, synthesize or code music signals, this *a priori* knowledge can be very useful and sometimes it can be transferred into a computer system in a successful way. For example in *Sinusoidal-Plus Residual Modeling* the *a priori* knowledge or assumption that a harmonic signal has strong approximately equally spaced frequency peaks sustained along the time can be used for analyzing, transforming, synthesizing or encoding the deterministic and stochastic parts separately.

Another interesting *a priori* knowledge is that once a musical scale is defined, the fundamental frequency of all the musical notes are related to one "standard" frequency by approximately fractional numbers. By its own nature, at least theoretically, the deterministic part of music sounds (i.e. the fundamental tones and their harmonics) will have a well-defined non-linear distribution in frequency. In practice, this is usually not the case. Specially in music instruments that can present vibrato, legato, pitch-bending, etc.

However, it is reasonable to assume that a multiresolution analysis in which each spectral bin represents one of the "musically interesting" frequencies (i.e. frequencies in which deterministic note components are likely to be located) could improve the results provided by using just STFT. This is one of the fundamental hypothesis of the present work.

To support the idea that these "musical frequencies" are more present in music than the rest of the frequencies, we can see Figure 1.2. The characteristic pattern that this figure presents shows that a for a typical piece of radio music (probably western music), a well-defined harmonically-related set of frequencies are clearly more relevant. This suggests that, even though in music signals we can find strong energy components at any frequency, statistically, there are some harmonically-related frequencies in which most energy is localized.

1.2.3 Automatic note transcription

The problem of automatic music transcription is one the most tackled in music technology. One of the most comprehensive and recent works on this matter is [12]. In it, Klapuri analyzes several approaches for the identification of musical notes in a polyphonic mix. Most of the methods make use of the STFT in

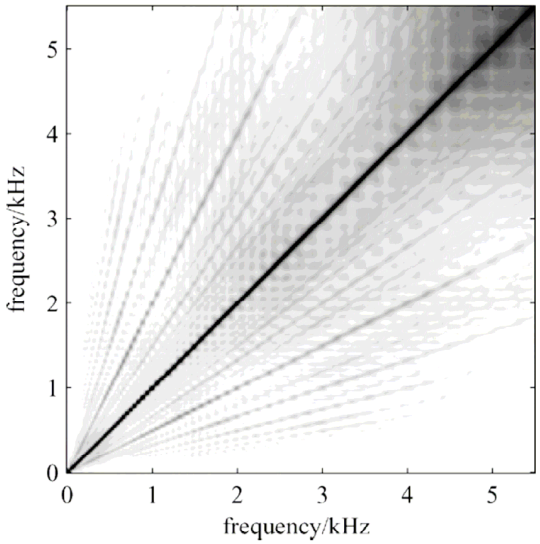


Figure 1.2: Cross-correlation of spectral magnitudes for several hours of music radio. The straight lines indicate correlations between harmonically related frequencies (figure extracted from [1]).

order to track the most prominent sinusoidal components (partials) of the signal, and then use some kind of artificial intelligence (A.I.) technique to cluster these partials into groups that form notes. These methods include: timbre models, Markov chains that model chord transition probabilities, Bayesian networks to integrate top down and bottom-up knowledge, Kalman filtering to estimate temporally continuous sinusoidal tracks, likelihood functions that represent perceptual grouping rules, etc.

To improve the performance, an iterative approach can be implemented: once a note is detected, it is removed from the mix and the estimation is repeated for the residual signal until all notes are detected.

One of the most important problems that all these methods have to solve is the *coinciding frequency partials*: "in Western polyphonic music, it is rather a rule than an exception that the partials of a harmonic sound overlap with the partials of other concurrent sounds. Thus, it does not suffice to merely group partials to sound sources but even individual partials need to be shared between sources", says Klapuri.

Fourier transform is very useful for frequency representations because every frequency bin represent signal components that are (ideally) orthogonal to each other. But the problem is that musical notes are *not* orthogonal: many of them share the same "building blocks", that is: sinusoids with the same frequency. Therefore, the spectral representations helps a lot, but does not solve *per se* problems like: *given a partial, which note does it belong to?*. Or in the case

of implementing an iterative approach: *I found a set of equally spaced partials that are supposed to belong to one note but, how much energy from each one should I remove without affecting the detection of other possible notes present in the signal which partials may coincide with the ones I have just detected?*. There is no answer to these questions at "Fourier analysis level", and to find a satisfactory solution to these problems we have to apply higher level knowledge about music: a model of the timber of the notes involved, the rules of harmony, etc. This is the reason why most note transcription systems make extensive use of A.I. techniques.

One of the hypothesis of this thesis is that taking into account not only the magnitude, but also the phase of the partials that form notes, using a pitch-synchronous multiresolution approach, the presence of a given set of harmonically related partials in a mix can be identified and labeled at "signal processing level" (i.e.: without the use of any further A.I. technique) even in the "worst cases": fifths, thirds, etc.

1.3 Organization of the thesis

The rest of this thesis is organized as follows: In chapter 2 an introduction to wavelets and multiresolution analysis is made, focusing on their relationship with audio and specially music signals. In the first part of chapter 3 a brief description of the state of the art in music transcription is performed. In the second part, we make a review of different works that have made use of wavelets for music analysis. In chapter 4 we present the Pitch-Synchronous Wavelet Spectrogram, a novel music representation tool that can be useful for some music-related applications, like automated note transcription or pitch class classification. In chapter 5 we analyze some examples of note detection using MIDI-synthetic and also real-world monophonic and polyphonic music signals. Finally, in chapter 6 we present the conclusions of this work and suggest some future work.

Chapter 2

Multiresolution analysis

In this chapter, a brief qualitative study of wavelets for audio processing is performed. A large number of more or less friendly introductions to wavelets can be found in Internet: in [13], for example, there is good compilation of links. For a deeper and comprehensive mathematical analysis on wavelets see, for example, [11] or [14].

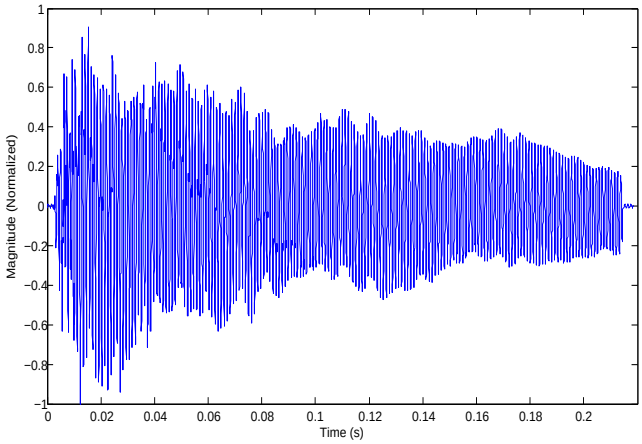
2.1 Motivation: Fourier analysis and music

The Fourier transform has been traditionally the most important tool for audio analysis. The Fourier frequency representation of signals is very useful for an intuitive understanding of operations like filtering, for example. For harmonic signals, the Fourier representation is much more compact and informative than time representation. To illustrate this we can see in Figure 2.1(a) that the waveform of the signal in time only gives a limited information about the signal (a piano musical note). On the other hand, the frequency representation shows the harmonic structure of the signal, and what is also important, the representation is much more compact since most of the coefficients are close to zero.

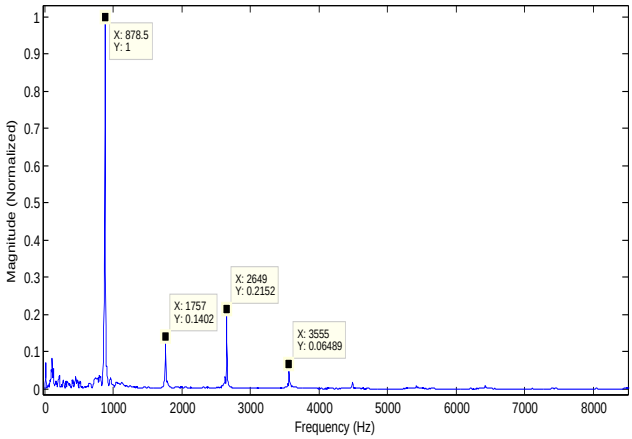
The frequency representation is very useful for some interesting manipulations in music. For example, if we want to transpose the A_5 note in Figure 2.1 into an A_4 note, we can just double the sample frequency. Of course, the larger the transposition in frequency is, the more unnatural the signal will sound. There are much more refined techniques to do this kind of transposition, but let this example be just an illustration of the good relationship between Fourier representation and music signals.

The question then is: why bother with wavelets when Fourier analysis is so very well suited for music signals? As it has been said in chapter 1, there are some time-frequency resolution problems that cannot be solved without a multiresolution approach. Also, the nature of our ear and the nature of the musical notes are also multiresolution.

To illustrate this last point, see Figure 2.2. It corresponds to the Fourier



(a) Time representation of an A_5 piano note



(b) Frequency representation of an A_5 piano note

Figure 2.1: Representation of a piano note in (a) time and (b) frequency. Note in (b) the characteristic harmonic pattern of musical notes: most of the energy is concentrated in the spectral bins around the fundamental frequency (in this case, approximately 880 Hz) and the bins around the multiples of this frequency.

Transform of a signal which contains all the notes of a wavetable-synthesized piano from A_0 to G_8. Note that the spectral peaks (harmonics) are much more concentrated in the low frequencies, while the peaks in high frequencies are more separated. By the own nature of the musical notes, if we want to describe the location of these peaks we will need much more resolution for low frequencies than for high frequencies. Note that between 9000 Hz and 12000 Hz only 5 different peaks can occur in this piano. The same number of peaks must be resolved between 300 Hz and 400 Hz. Using an FFT with a 1024-point window at a sample frequency of 44100 Hz we have a resolution of about 43 Hz. That is, we have 2 or 3 spectral bins available around 300-400 Hz, which is not enough to resolve the 5 harmonics, while between 9000 and 12000 Hz we will have about 70 spectral bins, far more than enough to resolve the same number of harmonics.

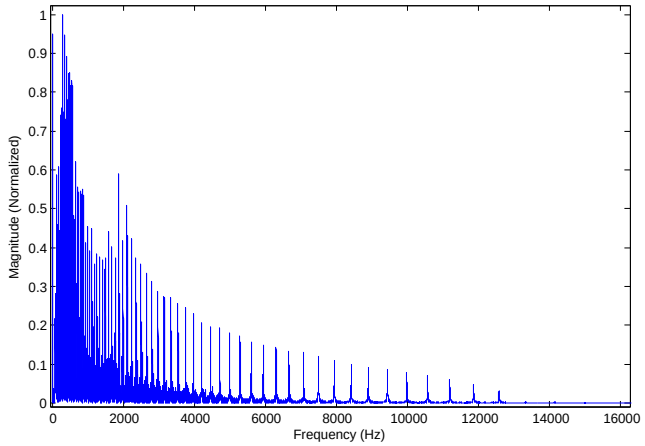


Figure 2.2: Frequency representation of a signal containing all notes from A_0 to G_8 in a wavetable-synthesized piano.

In other words: to achieve the necessary resolution to resolve all the harmonic peaks that can occur in this piano[1], we need to use a very large window; but the high frequency bins are somehow underutilized since no matter which note we play, there are large areas between high frequency peaks in the spectrum that will (at least theoretically) never present significant peaks, because of the structure of musical notes. These ideas apply to other musical instruments that have fixed note positions, like a flute or a guitar, for example. They could also be applied with caution to other musical instruments or to music in general as seen in Figure 1.2.

As we saw in Figure 1.1, the larger the analysis window, the worse the time resolution. The reason is illustrated in Figure 2.3. In this case, a 32-samples window length STFT tiling is used. The purpose of this figure is to underline

[1]In a "real" piano, larger deviations in the spectral peaks from their theoretical position may occur. Nevertheless, it is clear that even so, we need much more resolution at low frequencies than at high frequencies.

that we do not have any information about what happens in time between samples 0 and 32.

Given this, it seems to be a good idea for music analysis and synthesis to use some kind of transform that has a good time resolution at the expense of a poorer frequency resolution for high frequencies (which is somehow unnecessary in music signals, as we have seen). That is, divide the time-frequency plane in *Heisemberg boxes* distributed in a more advantageous way. This is where the Wavelet transform enters the scene.

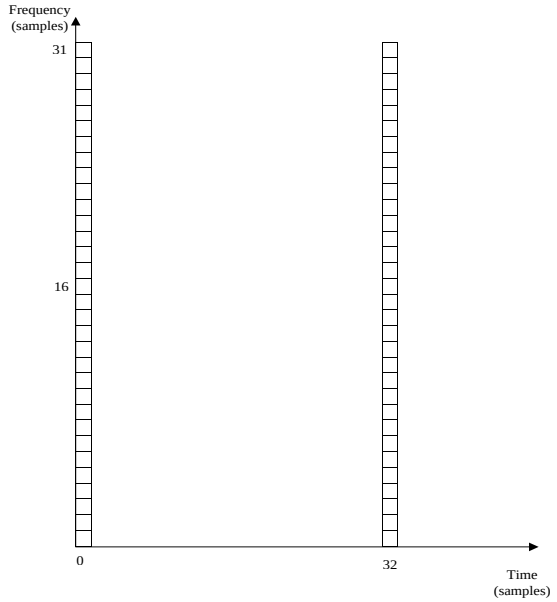


Figure 2.3: Short Time Fourier Transform tiling of Heisemberg boxes for a 32-point window length. Note that the more frequency resolution we need, the more time samples will be necessary to calculate the FFT, and so the less time resolution we will have.

2.2 The Wavelet Transform

Given the time-frequency resolution problems of Fourier analysis stated in previous section, and the characteristics of music signals, the solution seems to be straightforward: let us use larger windows for low frequencies and shorter windows for high frequencies, instead of a constant analysis window length. This is the idea of multiresolution analysis, but *how to do this* was very far from being easy, specially if we want to keep some very interesting properties of STFT analysis, like orthogonality, perfect reconstruction, critical sampling, etc.

The philosophy behind Fourier analysis is to project our signal over subspaces that consist of sines and cosines of different frequencies. The more our

signal is similar to one particular sine or cosine, the bigger the coefficient that corresponds to that frequency will be. In the continuous Fourier Transform we project on a dense an infinite number of infinitely long sines and cosines that represent all the possible frequencies from $-\infty$ to $+\infty$. In the discrete version, we only project on a finite set of sines and cosines of equally spaced frequencies. These frequencies are the result of dividing the sample frequency between the number of samples of the window.

The philosophy behind Wavelet analysis is to project our signal over subspaces that consist of some kind of "wavelets" of a particular shape with the purpose of providing good localization both in time and frequency. The Heisenberg uncertainty principle prevents wavelets from being arbitrarily compact both in time and frequency. The time-frequency tile of a wavelet decomposition is usually chosen to present short analysis windows (and thus, poorer frequency resolution) for high frequencies, and vice-versa. The dyadic decomposition is one of the simplest time-frequency tilings. It is represented in Figure 2.4 for a decomposition of level 5. It is very informative to compare this decomposition with that on Figure 2.3. In both representations, the vertical axis is related to frequency and the horizontal axis represents time. Each rectangle corresponds to one coefficient, and these, in turn, represent the correlation of the signal and the wavelet at a particular moment in time.

We see in Figure 2.4 that we have more (and larger) rectangles to represent low frequencies and less (and shorter) rectangles to represent high frequencies. The dyadic decomposition has also the particularity that the window length of the next scale is achieved by doubling the number of analysis points from the previous one. This has an interesting relationship with the octaves in music, as we will see in chapters 4 and 5.

It is important to distinguish between the Continuous Wavelet Transform (CWT) and the Discrete Wavelet Transform (DWT). Their relationship is similar to that between the Fourier Transform and the Discrete Fourier Transform. In a DWT representation we use more coefficients for high scales and less for low scales (as shown in Figure 2.4), while in a CWT representation we use the same number of coefficients in every scale: usually as many as original temporal samples. That means that if we have a decomposition of level 8 of a signal 1024 points long, we will get 8192 CWT coefficients. This is done by a proper overlapping of the analysis windows at each scale. So, the CWT is a highly oversampled representation and may require an important computational effort. In this thesis, when we refer to the wavelet transform we will always mean the DWT, unless otherwise stated.

Depending on the problem, we can choose from a large amount of different wavelets, most of them discovered in the past 30 years. The simplest wavelet was discovered by Haar in 1910. The expression of the Haar wavelet is:

$$\psi(t) = \begin{cases} 1 & if\ 0 \leq t < 1/2 \\ -1 & if\ 1/2 \leq t < 1 \\ 0 & otherwise \end{cases} \tag{2.1}$$

And the dilations and translations by integer numbers j and n of this func-

tion:

$$\psi_{j,n}(t) = \frac{1}{\sqrt{2}^j}\psi\left(\frac{t - 2^j n}{2^j}\right) \tag{2.2}$$

generate an orthonormal basis, such that any finite energy signal can be decomposed over this wavelet basis. The wavelet function ψ is associated with a high pass filter and so with the **detail level** of the decomposition.

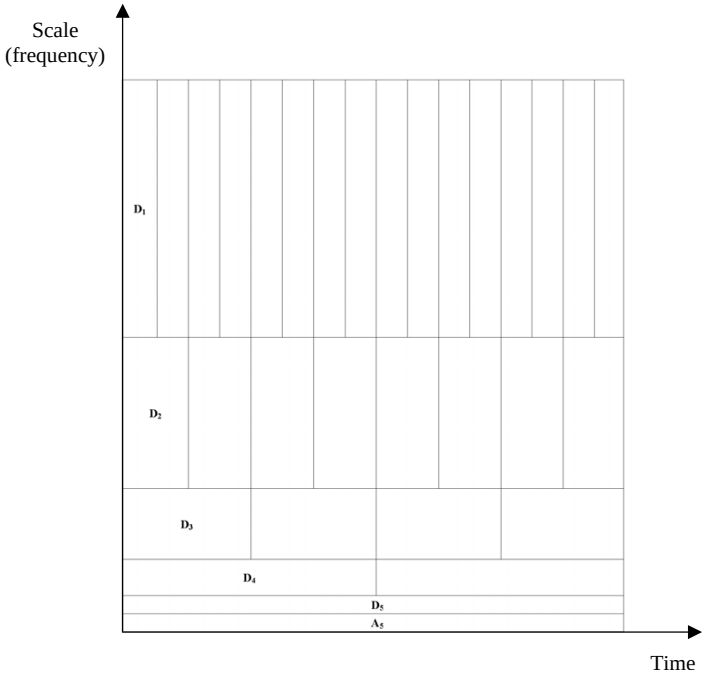


Figure 2.4: Typical dyadic time frequency decomposition. A stands for approximation level and D for detail level. This example is a decomposition of level 5.

There is another function ϕ, called *scaling function* that is associated with a low pass filter and so, with the **approximation** level of the decomposition. The Haar scaling function is:

$$\phi(t) = \begin{cases} 1 & if\ 0 \leq t < 1 \\ 0 & otherwise \end{cases} \tag{2.3}$$

The ψ and ϕ functions can be viewed as *quadrature mirror filters*. The dyadic decomposition is obtained by filtering the signal to give the level 1 *approximation + detail* decomposition, and then separating again iteratively the obtained approximation in further *approximations + details*.

To illustrate all this, we can use an impulse signal as an example. The same way the frequency response of a filter can be characterized by filtering a time impulse signal, we can see the shape of the ψ and ϕ functions by filtering an impulse in time. This is shown in Figure 2.5

Haar wavelet is simple but its usability is limited because it is very well localized in time (it is exactly zero outside the interval of definition: this is called *compact support*) but very bad localized in frequency because of the sharp discontinuities that it presents: as it is well-known, the Fourier transform of these kind of step functions is an infinitely long and not too fast decaying *Sinc* function.

Ingrid Daubechies [14] in 1987 found a way to design a very interesting family of wavelets. They were very celebrated not only because of their properties — orthogonality, compact support and easy implementation, among others— but also because of their startling fractal structure. The Haar wavelet is a particular case of Daubechies family. The family is ordered according to the number of vanishing moments of the wavelets: the higher the vanishing moments, the more smooth is the wavelet. Haar wavelet is the Daubechies wavelet of order 1. In Figures 2.6 and 2.7 we show the wavelet shapes of Daubechies wavelets of order 2 and 9 respectively.

To summarize this qualitative introduction to the wavelet transform, we can say that the wavelet functions play the role of the sines and the cosines in the Fourier transform. The advantage is that wavelets are constructed in such a way that they can achieve some interesting properties of the discrete Fourier transform, like perfect reconstruction, orthogonality, critical sampling, etc. while achieving a time-frequency distribution of the coefficients that can be more advantageous for some applications.

2.3 Scale and Frequency

When talking about wavelets, an interesting question usually arises: what is the relationship between *scale* and *frequency*? In this section we will try to clarify this issue a little bit.

Let us focus first on the Fourier Transform, or even better, on its real discrete counterpart the DCT. We can see this transformation as some kind of wavelet decomposition, with a very special family of wavelets: cosines of linearly increasing frequencies. We can see the first 8 cosines of this kind in Figure 2.8. If we make a 256-point decomposition, for example, we will have 256 functions analogous to these.

The spectrum[2] of these functions is shown in Figure 2.9. We can say that in the ideal case, we have a filter bank in which each band-pass filter represents each discrete frequency. Of course, this is not true in general, because of the side lobes of the filters. But it is a good approximation if enough care is taken

[2]We have chosen the frequency of the basis functions and the sample rate in such a way that this spectral representation seems to consist of perfect filter banks without any side lobes due to the "sinusoids on the clacks" effect.

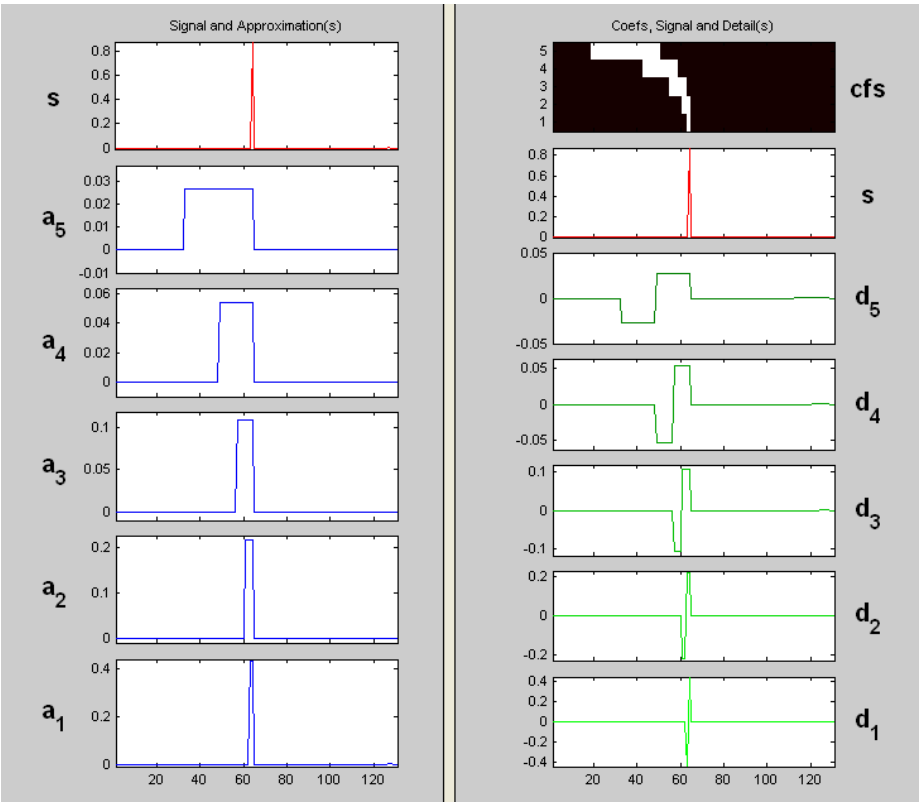


Figure 2.5: Haar decomposition of level 5 of an impulsive signal. We can see the different wave shapes of the ϕ functión (left) and ψ function (right). The "cfs" plot is the same time-frequency tiling shown in Figure 2.4 but upside-down. Note that as level increases, wavelets become shorter and more coefficients appear in the same segment of time.

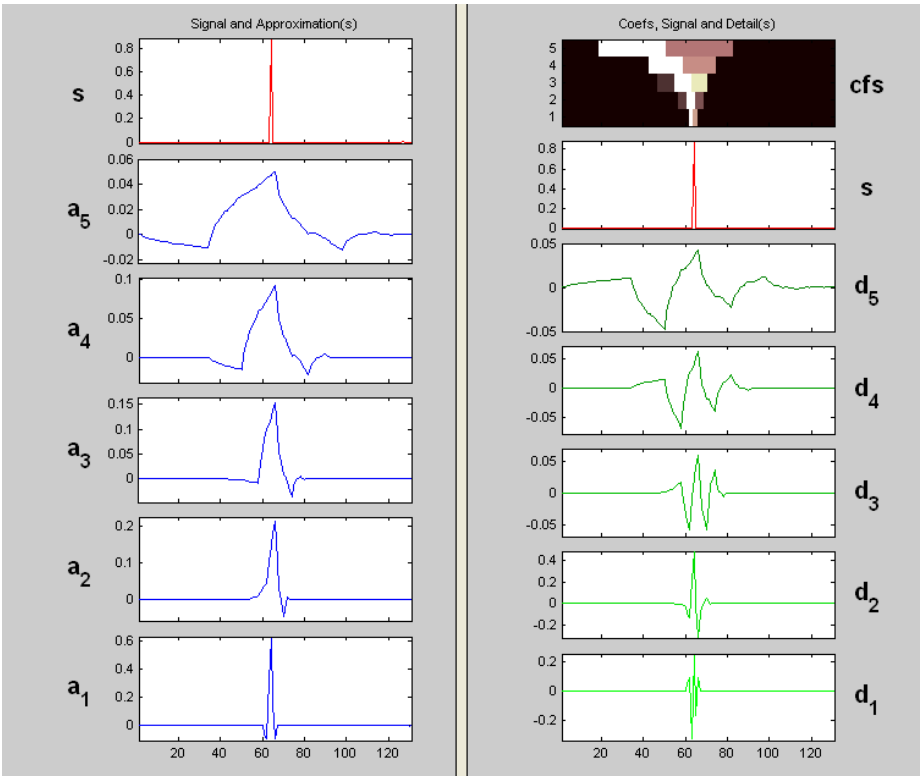


Figure 2.6: Daubechies decomposition of order 2 and level 5 of an impulsive signal. We can see the fractal-like shape of the ϕ and ψ functions.

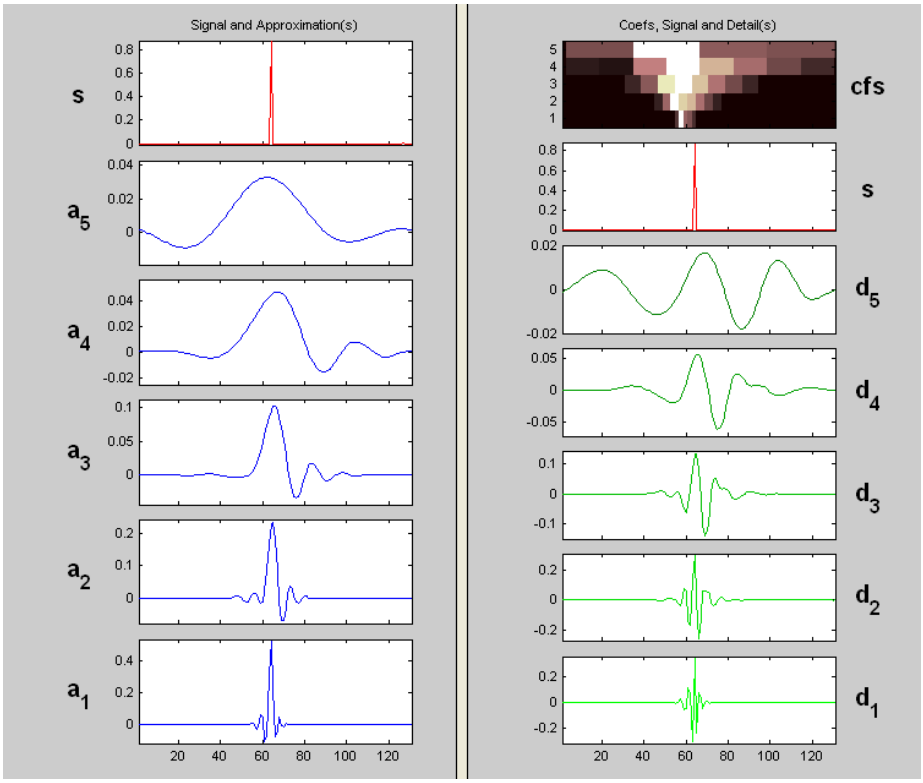


Figure 2.7: Daubechies decomposition of order 9 and level 5 of an impulsive signal. In this case the waveforms are much more smooth. These wavelets have a shape similar to that of a windowed sinusoidal function and so they are better localized in frequency.

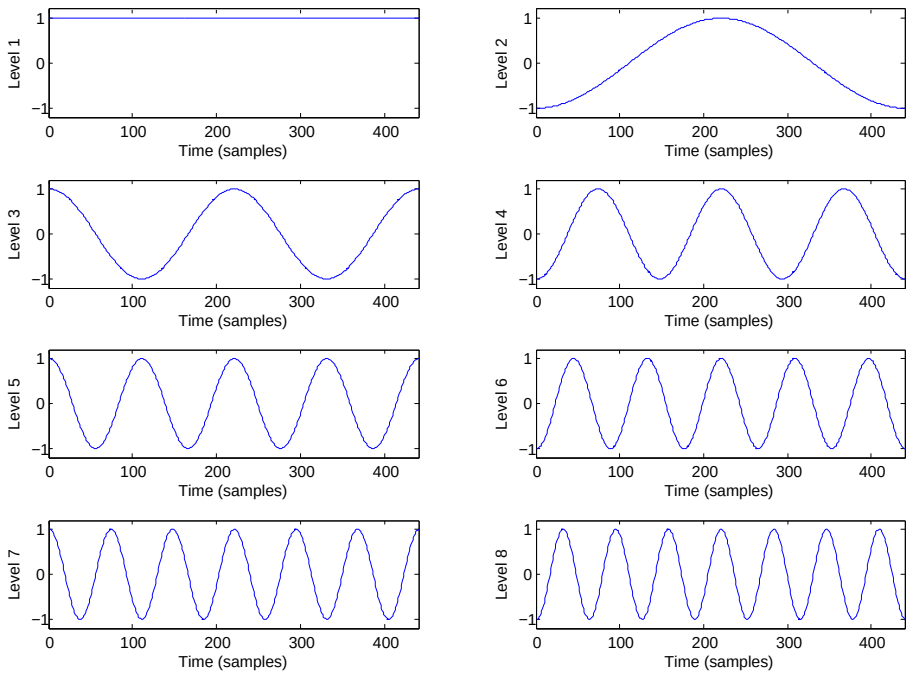


Figure 2.8: Waveforms of the first 8 basis functions of a DCT decomposition.

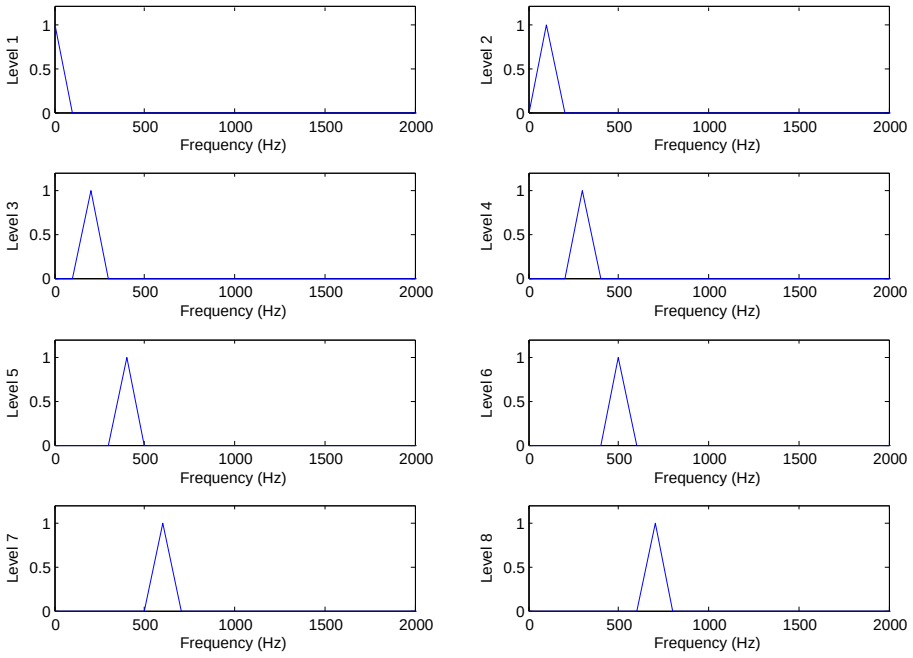


Figure 2.9: Spectral representation of the waveforms on Figure 2.9. We can see each waveform as a filter with a very good localization in frequency, and constant bandwidth.

when choosing the parameters of the transformation: window length, window shape, sample rate, etc.

We have chosen the term "Level" to label each filter to highlight the analogy between this decomposition and the wavelet decomposition. In Figure 2.5 we had the Haar basis fuctions of a level 5 wavelet decomposition. The wavelet function, that is, the ψ, plays the role of the cosines in a DCT transform (or the sines and cosines in a complex FFT transform). Therefore, we can see the wavelet decomposition also as a filter bank, but in this case the filters do not have constant bandwidth.

Let us see in Figure 2.10 the spectral representation of the 8 wavelets (ψ functions, like those on the rigth side of Figure 2.5) in a level 8 Haar decomposition. We see that the centers of the filters are separated by exactly one octave, and not linearly spaced as in the FFT or the DCT. This is a direct consequence of the dyadic decomposition used. We also see that, as we mentioned before, the filters are in this case very bad localized in frequency due to the sharp transitions that characterize the Haar wavelet. Note that pure tones will not be expressed in general by just one coefficient like in the Fourier Transform. As long as the filters are heavily overlapped, each tone will contribute to more than one wavelet coefficient.

Nevertheless, when the signal we want to analyze is such that its period is a multiple of the length of the analysis wavelets, very interesting properties occur, as we will see in chapter 4. One of these properties —but not the most interesting— is that in spite of the bad frequency resolution of the Haar wavelet transform, we can identify octave spaced pure tones just by looking at the coefficients: the 5th tone (that on the right most part) will only present non-zero coefficients after level 7; the 4th tone, after level 6, etc.

To complete this analysis of the relationship between wavelets and frequency, we show in Figures 2.11 and 2.12 the spectral representation of the higher order Daubechies wavelets shown in Figures 2.6 and 2.7 respectively. We can see that the higher the order (i.e.: the number of vanishing moments) the smoother the time representation, and the better the frequency localization (the sidelobes decay much more quickly). One disadvantage of Daubechies wavelets is that they are not critically sampled, except the Daubechies wavelet of order 1, that is, the Haar wavelet. This property of the Haar wavelet makes it interesting for adapting the wavelets length to the interesting periods (or frequencies) in a musical signal. We will see how to exploit this and other properties of the Haar wavelet for musical purposes in chapter 4.

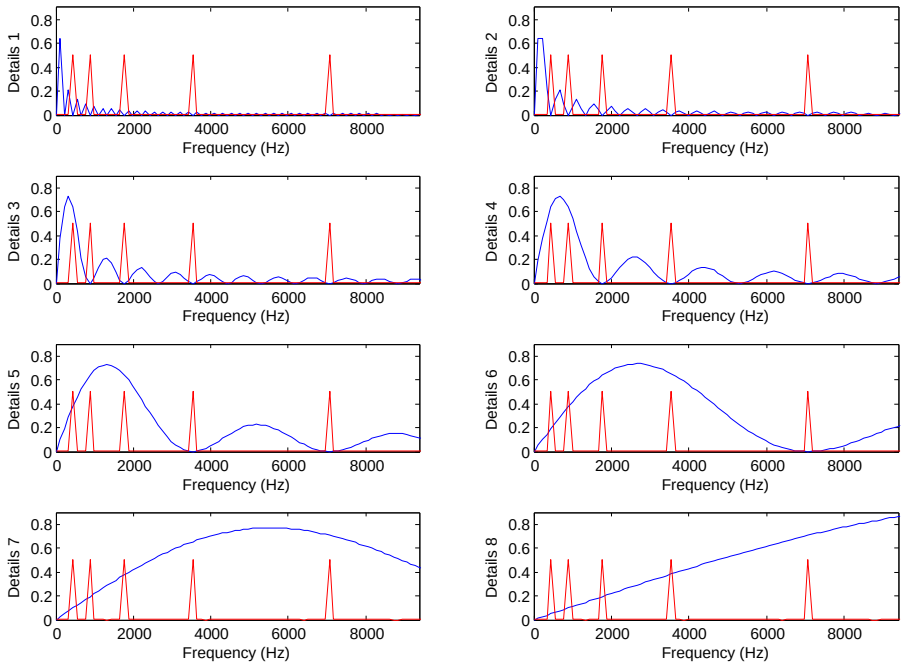


Figure 2.10: Spectral representation of the 8 wavelets of a level 8 Haar decompostition (blue). Compare this frequency representations of the ψ function with the time representations in Figure 2.5. Five octave-spaced tones are also shown to serve as a reference (red). The frequency of the reference tones is such that their period matches the length of the wavelets at the different scales. Note that the main lobe is always centered in the center of the corresponding octave, and that the nodes of the side lobes coincide with the boundaries of the next octave.

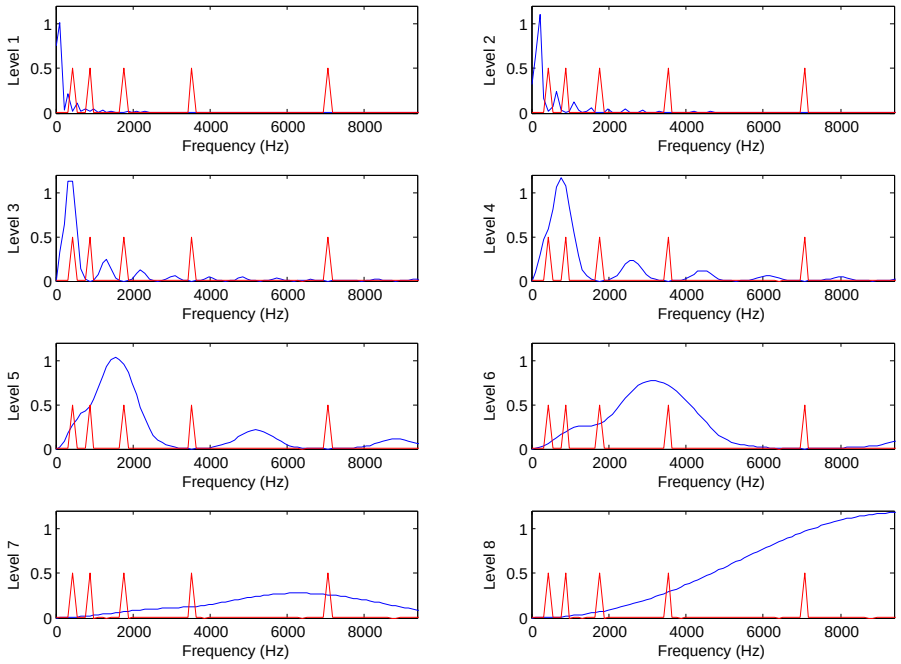


Figure 2.11: Spectral representation of the 8 wavelets of a Daubechies decompostition of order 2 and level 8 (blue). Compare this frequency representations of the ψ function with the time representations in Figure 2.6. Five octave-spaced tones are also shown to serve as a reference (red).

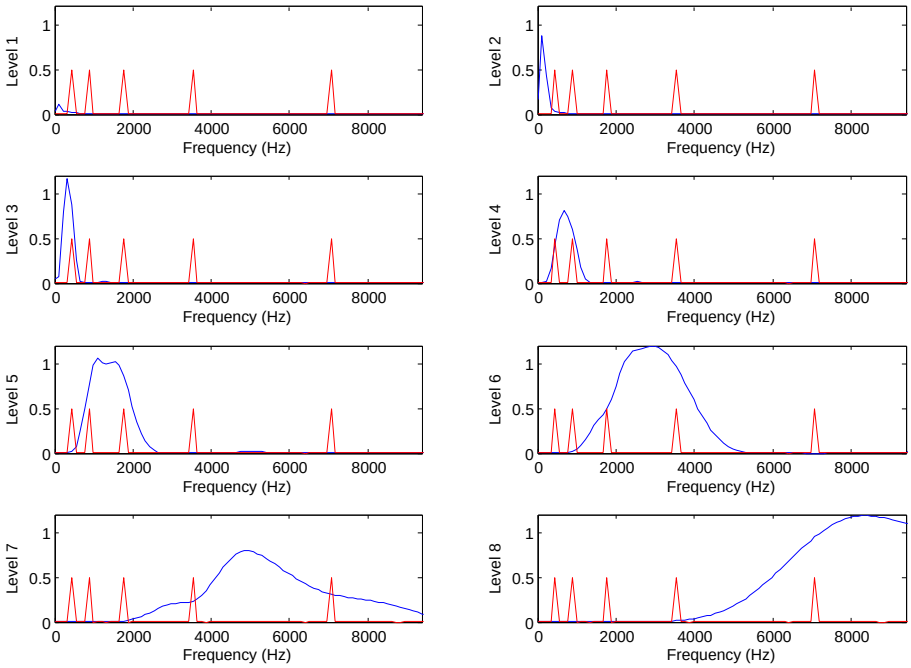


Figure 2.12: Spectral representation of the 8 wavelets of a Daubechies decompostition of order 2 and level 9 (blue). Compare this frequency representations of the ψ function with the time representations in Figure 2.7. Five octave-spaced tones are also shown to serve as a reference (red).

Chapter 3

State of the art

3.1 Polyphonic music transcription

One of the most straightforward applications of the multiresolution analysis tool that will be presented in this thesis is for monophonic and polyphonic music transcription. Nevertheless, the objective is not to measure or to compare the multiresolution tool developed herein against other music transcription methods. This would be undoubtedly interesting, but it is beyond the scope of this thesis. In any case, it is interesting to make here a brief revision of the most important techniques for music transcription to have an idea of different approaches to the solution of this problem.

Very comprehensive works about music transcription can be found in [15, 12, 16]. The following classification and discussion is based on these and other works by Klapuri and Hainsworth.

First, let us define Music transcription as the process of seeking to determine all the notes sounded in a musical audio ample and notating these in a conventionally recognisable form. Historically, there have been three approaches to the problem: bottom-up heuristic methods, blackboard methods and model based methods.

3.1.1 Bottom-up methods

The early approaches in the 1970s belong to this category. This methods were based on looking at the "greatest common harmonic" or finding the relationship between all the pairs of partials found at each step in order to classify them in groups forming notes. A detailed review of these methods can be found in [16].

3.1.2 Blackboard Methods

The Integrated Processing and Understanding of Signals (IPUS) [17] was developed at the University of Massachusetts. The philosophy of this approach is to extract information from the audio signal with a frame by frame scheme

using a series of signal processing algorithms, and then find the parameters that describe this signal, via the use of competing *knowledge sources* and multilevel hypotheses. That is: combining top-down priors with bottom-up information flow. However, its dependency on the heuristics makes it very dependent on the design characteristics. Its performance for polyphonic audio is not satisfactory.

3.1.3 Model-Based Algorithms

In this case, the designer explicitly sets down the structural relationships which are assumed to exist within the data and formulates the model with these in mind. The better the model reflects the content of the data, the more successful will be the performance of the system. Early approaches used Bayesian formulations, combining peak detection and tracking with peak grouping techniques. Nearly 80% of accuracy for 4 note polyphony was reported in [18].

Other approaches include Markov chains, Monte Carlo methods or generative models for producing a MIDI type representation from a score, thereby including expressive performance characteristics. Recent transcription systems also use psychoacoustically motivated analysis principles, models of the human auditory periphery and sparse coding methods.

As a summary, we can say that in most transcription systems, temporally continuous sinusoidal components, or *sinusoidal tracks*, are used as the elementary low-level descriptors that are extracted from the signal at a first step. The detection of sinusoidal components is usually carried out using an STFT. Then, a combination of bottom-up (given the components extracted, we try to cluster them in notes, notes in chords, etc.) and top-down techniques (chords predict notes and notes in turn predict components) is used to extract higher level descriptors.

The basic problems of polyphonic note transcription (or more precisely, of multiple-F0 estimation), according to Klapuri [12], can be clasified in four cathegories:

- The grouping problem: given a mixture of harmonic sounds, how to organize them according to their sound sources. Due to the *inharmonicity phenomenon* (i. e.: harmonic partials are not always equally-spaced in real-world signals) the components of a harmonic sound cannot be simply assumed to reside at ideal harmonic positions in the spectrum.
- Computing the saliences of different F0 candidates given the partials of a sound.
- Achieving noise robustness: against percussive instruments, for example.
- Coinciding frequency partials: in western polyphonic music often the partials of a harmonic sound overlap with the partials of other, concurrent, sounds, thus making it difficult to identify the source of each partial.

On the other hand, music transcription methods are constantly being improved thanks to 3 factors:

- The fast increasing of computer power that facilitates, for example, computationally intensive techniques like Monte Carlo methods.
- The gradual improvement of the low-level processing techniques, i.e. the time-frequency mathematical waveform analysis tools.
- The accumulation of knowledge about the human brain and the cognition and processing of music.

The analysis tool proposed in this master thesis is intended to be a contribution to the second factor, providing a good low-level F0 discrimination, and so making it possible to improve the performance of higher-level techniques for the note transcription task. In particular, we propose an approach that is not based on the STFT, but on a pitch-syncrhronous multiresolution analysis tool adapted to the theoretical F0 of the musical notes. The solution proposed contributes specially to solve the problem of the *coinciding frequency partials*.

3.2 Wavelets and music

The idea of multiresolution has been the framework for many different approaches to the analysis, synthesis or feature extraction of polyphonic music since the late 1980's (see [19, 8, 20, 21, 22], to cite some).

A special *harmonic wavelet* was developed by Newland [23, 24, 25] to improve the time-frequency resolution for harmonic signals, although he applied his results more in mechanics than in music.

The work of Evangelista, Cavaliere and Polotti on *frequency warped wavelets* [26, 27, 28, 29] is an effort to adapt wavelets to harmonic signals in which frequency (or pitch) varies in a continuous way. Concepts like HBWT (Harmonic band wavelet transform), PSWT (Pitch-Synchronous Wavelet Transform) or FAS (Fractal Additive Synthesis) are proposed by these authors, and are applied to music analysis, synthesis, transformation and coding. In particular, the PSWT is conceptually very related to the present work, but there are important differences. Maybe the most important is that Evangelista proposes a system that is based on the estimaton of the pitch to analyze harmonic signals. In this thesis, on the other hand, we propose a representation system, Pitch-Syncrhonous Wavelet Spectrogram (PSWS), which do not try to estimate the pitch of the signals, but to adapt the wavelets to the theoretical pitches of musical notes. PSWT relies on the accuracy of the pitch detection, while PSWS relies on the approximation of pitches of real-world music sounds to their theoretical positions.

Another interesting approach is the one proposed in [2]. In it, the use of the Continuous Wavelet Transform (CWT) is suggested in order to automatically detect musical notes in a polyphonic signal. The idea is to compute the CWT using a narrow-band wavelet (i.e. a wavelet whose Fourier transform is as similar as a frequency peak as possible), in this case the Morlet wavelet, and then

look at the scalogram in search of components corresponding to the fundamental frequency of the musical notes. In Fig. 3.1 the scalogram of a pure tone signal is plotted. The difference of this representation with respect to the STFT spectrogram is clear: in the vertical axis we have the fundamental frequencies of the musical notes *equally spaced*. That is a direct consequence of a suited multiresolution analysis.

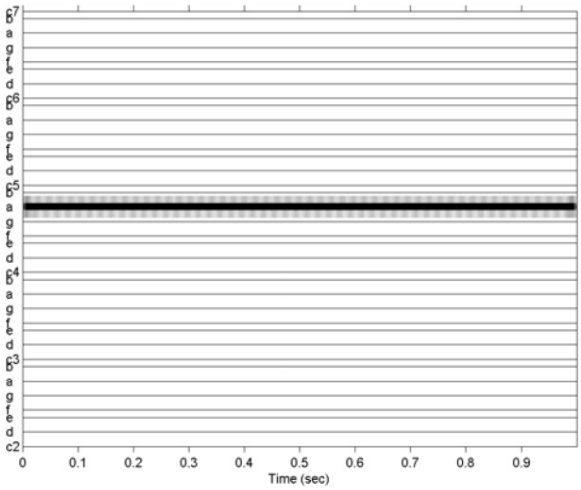


Figure 3.1: 2D time-frequency plot of CWT coefficients using a complex Morlet 1-5 wavelet. The signal being analysed is a pure tone of 440 Hz. (Figure extracted from [2]).

This kind of music representation has the advantage of considering the frequency axis directly as "musical note axis". But it has also important disadvantages with respect to the STFT representation. The most important is the prohibitive cost of CWT calculations. As the authors say in [2], "performing the CWT calculations for a signal with hundreds of thousands of samples over several octaves (each octave having 12 scales) was extremely demanding, even for a fast computer with lots of memory". They propose, however, some strategies to reduce the computational cost, such as finding the highest important frequency in each frame and lower down the sample rate accordingly, but "in practice, downsampling an audio signal that much produced unlistenable results [...]. However, in this case, the signals were used only for calculations an were not reconstructed for listening purposes".

Sinusoidal analysis has also been subject of multiresolution approaches. Levine, Verma and Smith III propose in [8], and in more detail in [30], four approaches to multiresolution sinusoidal analysis:

- To input a signal through an octave-spaced, critically sampled, wavelet filter bank. This has a low complexity but there is no known way to eliminate all aliasing between channels in the filter bank.

- To use a parallel bank of constant-Q bandpass filters as a front-end and perform sinusoidal modeling on each bandpass filter output. The problem of this approach is that the structure is highly oversampled, what increases the complexity and data storage needs.

- To take multiple sliding FFTs of the same audio input with a different window length. Again, the complexity of this algorithm increases with the number of octaves (sets of FFTs) taken.

- To use an octave-spaced complementary filter bank. This approach is claimed to overcome the drawbacks of previous ones, but does not provide critical sampling for synthesis due to the alias-free subbands constraint imposed by the filter bank.

Bello includes in his PhD. thesis [31] a very interesting review of different methods for time-frequency analysis. Multiresolution analysis (constant Q, wavelets) is discussed but finally the author decides to use a phase-vocoder approach for his analysis tool. The reasons for this choice against a multiresolution approach are that "although time resolution is worse, frequency resolution is more important, for note detection for example. The time resolution problem can be solved by using an additional onset detector. Phase-vocoder (based on STFT) is a very well-known technique and there are efficient implementations available".

We can conclude from this revision of works about the relationship between wavelets and music that many researchers suspect that multiresolution analysis should improve the way we treat music signals. Nevertheless, even though wavelets had been successfully applied in the 80's and the 90's to a wide variety of problems, when dealing with music signals, Fourier analysis seems to be still the preferred choice.

Chapter 4

The Pitch-Synchronous Wavelet Spectrogram

4.1 Introduction

We have seen in chapter 2 that the dyadic decomposition provides only approximately one octave bandwidth filter banks. This is of course not enough to resolve every *musically interesting* frequency as defined in chapter 1. There are many other possible decompositions. One way to choose the most adequate is to use *wavelet packets*. But it is a difficult mathematical problem to design wavelets that are centered in arbitrary frequencies. Specially if we want them to have properties like orthogonality, good time-frequency localization, etc.

Given that the dyadic decomposition is very related to the octave-based frequency distribution of musical notes, but it does not provide enough resolution to resolve the 12 semitones within every octave, the approach we propose in this master thesis is to analyze musical signals by using 12 different sets of dyadic wavelets: one per semi-tone.

4.2 A multiresolution pitch-synchronous approach

As we mentioned in chapter 2, very interesting properties occur when the length of the wavelets (basis functions) coincide with the fundamental period of a harmonic signal. The wavelet coefficients, as well as the Fourier coefficients, are a measure of the similarity between the signal subject to analysis and the basis function chosen. This similarity is calculated through the correlation (or projection) of the signal and the correspondent basis function.

Let us suppose that we divide the signal in short segments that are one period (or an entire number of periods) long. Let us project one of these segments on a basis function of the same length. We will get one coefficient that measures their similarity. We do not care for the moment about the degree of similarity between

them, that will depend on their shape and their phase lag. The important thing here is that when we go to the next signal segment it will contain again one period of the signal. And if the signal is locally stationary, it will be very similar to the previous one. That means that the wavelet coefficient will be very similar to the previous one too.

What happens if we divide the signal in arbitrary segments without taking into account the intrinsic periodicities of the signal? In this case consecutive segments will be *out of phase* and the correlations of the basis function and the signal will be pseudo-random, or even pseudo-periodic. That means that the coefficients will not show temporal continuity even if the signal is stationary, or even purely periodic.

This is exactly what happens if we try to make the spectrogram of a pure tone (whose frequency is not related to the window length) using the DCT: the modulus of the coefficients will express approximately the frequency content of the signal, but the signs (i.e. the phase) will be pseudo-random (in fact, pseudo-periodic) and the spectrogram will not present smooth horizontal lines as one may expect. This is specially true for high frequencies in which the window length can represent hundreds of periods, and so, they are extremely sensitive to phase lags introduced by the temporal segmentation of the signal.

A STFT spectrogram usually expresses only the magnitude of the FFT. Therefore, the pseudo-random phase changes between consecutive segments do not affect the representation as in a DCT spectrogram. The problem in this case (and also in the DCT) is that when the frequency of the signal does not coincide exactly with that of the frequency bins (the sinusoids "on the cracks" artifact) the representation will present more spectral leakage and the peaks will not appear so clear. Looking at the phase evolution, it is possible to make a better estimation of the frequency content of a signal in the STFT spectrogram [4, 5, 6]. The problem is that the more frequency resolution we need, the bigger the time window must be. And therefore, the "continuity" in time of consecutive frequency coefficients is again not as smooth as one may expect because they might be quite separated in time (as shown in Figure 2.3), and the the signal could have changed significantly. That is: the signal cannot be considered locally stationary for large analysis windows.

A multiresolution approach could help in the task of achieving smooth frequency representations of periodic signals. As we saw in the dyadic time-frequency representation of Figure 2.4, the analysis window for low frequencies is much bigger than for high frequencies. For the same period of time, we calculate many level 8 coefficients (high frequencies) and few level 2 coefficients (low frequencies), for example. Therefore, the changes due to variations in the signal that may occur between consecutive coefficients will not be too sharp at any scale. Nevertheless, the changes due to the phase lag between segments will, in general, be very important indeed, unless we divide our periodic signal into frames that contain an entire number of periods. In such a case the wavelet coefficients will show a high degree of temporal correlation, even if the signal is not purely periodic but pseudo-periodic, and also even if consecutive frames present small phase lags, no matter which scale (frequency) the coefficients belong to.

We will not use in this thesis the typical representation that is shown in Figure 2.4, or the "cfs" subplot in Figures 2.5, 2.6 and 2.7. Instead, we will make use of an approach that is closer to an STFT spectrogram, putting all the wavelet coefficients that belong to a given segment of time in one column. One important remark is that for these wavelet spectrograms to be fully useful, the length of the analysis windows must be chosen taking into account the properties of the signal subject to analysis. In our case, as we are going to deal with musical signals, the length of the analysis windows will be integer multiples of the fundamental frequency of each musical note.The term *Pitch Synchronous Wavelet Spectrogram* or PSWS will be used in the remaining of this thesis, instead of *scalogram* to distinguish between the representation proposed and the traditional scalogram wavelet representation.

4.3 Properties of the PSWS

Let us analyze the properties of each one of the columns of a PSWS. We will use the MATLAB function *wavedec*. The output of this function is an "structure" of coefficients that contains one sub-vector for the approximation coefficients, and another 8 sub-vectors for the details. If we use a level 8 decomposition we have the distribution of coefficients shown in Table 4.1. Note that the Haar discrete wavelet decomposition is critically sampled: we obtain 256 wavelet coefficients from 256 temporal samples. If we place these coefficients ordered by level in one column at a time, we can build a "spectrogram-like" scalogram in which the frequency information is quite coarse (we only have 8 different levels to distinguish frequencies, each level centered in one different octave) but nevertheless can be very useful as we will see.

	Number of coefficients
Approximation	1
Level 1	1
Level 2	2
Level 3	4
Level 4	8
Level 5	16
Level 6	32
Level 7	64
Level 8	128

Table 4.1: Structure of coefficients in a DWT dyadic decomposition using the *wavedec* MATLAB function.

.

We show in Figure 4.1 the output of the Haar decomposition of several 256-point time segments corresponding to different pure tones. The first one —Figure 4.1 (a)— has a frequency of 440 Hz, and the frequencies of the rest are integer multiples of 440 Hz. The horizontal axis is divided using vertical lines

that delimit each sub-vector in the structure of coefficients provided by *wavedec*. In each of the segments between these vertical lines we find the coefficients that belong to each detail level. As it is usual in a wavelet decomposition, we have a better resolution for low frequencies than for high frequencies. Coefficients within each level represent different moments of *time*, but all of them represent the same frequency area. To see the frequency area covered by each decomposition level, see Figure 2.10. As it is shown there, the frequency resolution is very poor, but the good news is that when the frequency of the analyzed signal is such that its period coincides with any of the wavelet analysis windows for each level, *all the temporal coefficients within the corresponding level take the same value.*

Let us analyze an example in which we are interested in detecting the fundamental frequency of the A note at any octave (Figure 4.1). The first thing we have to do is to choose the sample frequency, which has to be an integer multiple of 55, 110, 220, 440, 880, 1760, 3520, etc. that is, all the fundamental frequencies of the A note in the whole audible spectra. Let this sample frequency be 28160 Hz. As the segment of the signal analyzed is 256 samples long, the number of coefficients for each detail level will be 128, 64, 32, 16, 8, 4, 2 and 1 respectively, as shown in Table 4.1. To calculate the 8 coefficients of level 4, for example, the analysis "sub-window" will cover a temporal area of $256/8 = 64$ samples. That is exactly the period in samples of a pure tone of 440 Hz sampled at 28160 Hz. In the same way, the analysis sub-window for level 8 coefficients is 128 samples long, and it covers a temporal area of $256/128 = 2$ samples. That is the period of a pure tone of 3520 Hz sampled at 28160 Hz.

Given what we have just stated, it is not surprising that in Figure 4.1 (a), (b), (d) and (h), coefficients within the corresponding level have *exactly* the same value: they express the correlation of the Haar sub-windows with *exactly* the same piece of signal, since it is periodic. In other words: the fundamental frequencies of the A note in every scale, produce a "continuous pattern" (let us say DC pattern) of Haar wavelet coefficients within the corresponding level, provided we choose the sample frequency as being a multiple of the highest of these frequencies. All other frequencies are expected to produce "alternative patterns" (AC).

To take advantage of this DC pattern we can simply low-pass filter the wavelet coefficients applying a suitable filter for each level. The result is shown in Figure 4.2. Another possibility is to take the mean value of the coefficients, instead of filtering. But low-pass filter is more convenient for non-ideal situations in which the coefficients within each level are not exactly the same, but experiment slow variations that may eventually cross the zero level.

If we take into account that each level represents (roughly) one different octave (remember Figure 2.10), in principle, with 256 points and 8 levels we will be able to analyze 8 different octaves. In practice, the first levels do not give so much information because there are too few coefficients involved. Anyway we can use bigger analysis segments (1024 points for example), or take into account the correlation between consecutive temporal frames in the spectrogram, specially for the lowest levels. For the sake of clarity and simplicity, we will only

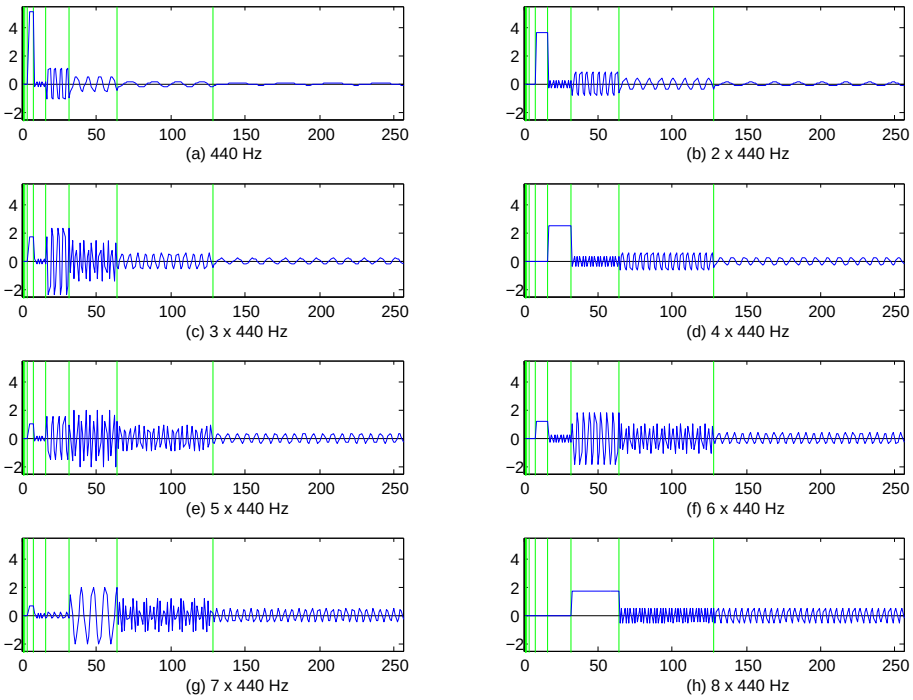


Figure 4.1: Haar wavelet coeficients of a 256-point temporal segment of pure tones of different frequencies. The frequency of the corresponding tone is stated under each plot. The vertical lines delimit the coefficients that belong to each decomposition level.

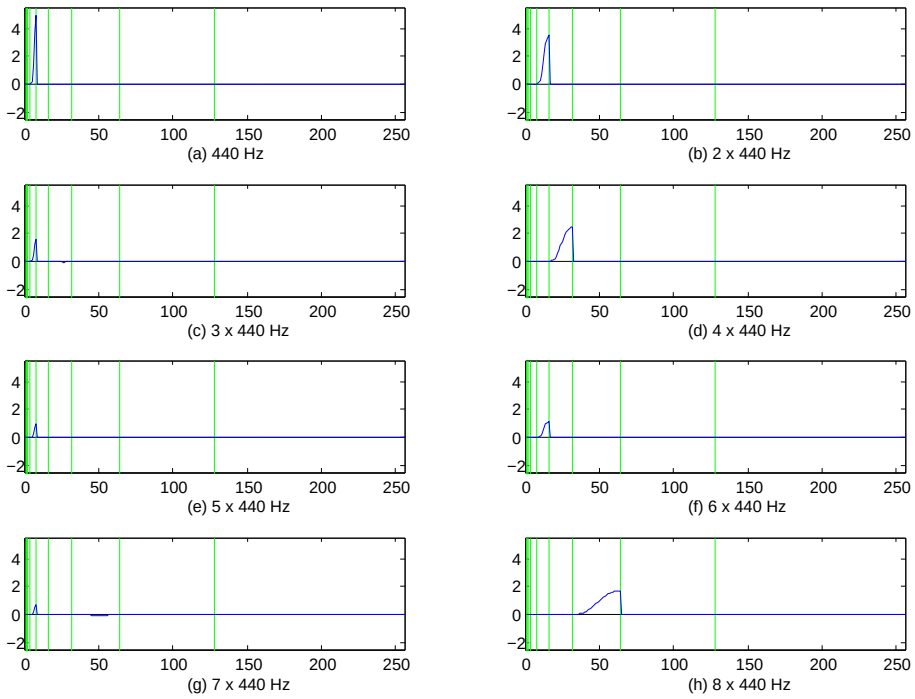


Figure 4.2: The same coefficients in Figure 4.1 low-pass filtered. Note that we can identify to which octave belongs the tone which period coincides with the wavelet analysis sub-window of the corresponding level: (a), (b), (d) and (h).

use in this work 256-point time frames and level 8 Haar decompositions, which seems to be enough to analyze a range of at least 4 or 5 octaves.

Whenever the fundamental frequency of the "A" note appears at any octave, it will be easily identified after filtering the wavelet coefficients, as we see in Figure 4.2. We would like our decomposition to be completely orthogonal, that is: after filtering, only the fundamental frequency of the A notes will stand (if they are present in the signal subject of analysis) and all the others would be removed by the filter. This is usually the case for any frequency that is not multiple of the fundamental frequency of any A note: they create pseudo-random or periodic patterns in the wavelet coefficients that have zero mean and they are removed by the filtering.

4.3.1 The *reinforcement of the fundamental* effect

There is a very interesting effect that is produced in the PSWS with harmonic signals. We can see in Figure 4.2 that even though most of the coefficients that do not represent the F0 are filtered, a small spurious DC pattern also appears in lower scales when the signal analyzed is (or contains) a multiple of the F0. In particular, at the level that corresponds to the maximum common divisor between the F0 of the note and the harmonic. For example, the 2nd harmonic of the A_4, which approximately coincides with the fundamental frequency of the E_5 note —around 1320 Hz depending on how we define the musical scale—, creates a small spurious DC pattern in the 3rd level (Figure 4.2 (c)), which corresponds to a frequency of 220 Hz, the maximum common divisor between 1320 and 440 (of course, apart from 440 itself). The same happens for all even partials (Figure 4.2 (e) and (g)). Whenever an "A" note occur, in most musical instruments all these harmonics will appear, and so this spurious effect. But in this case they all contribute to reinforce the fundamental partial and it is not a problem for *monophonic* notes detection, since the fundamental is usually also present. What is more: even if the fundamental partial is not present and only its harmonics appear, this sum of spurious effects might help in the determination of the note that is being played. An analogous effect occurs with the odd partials. The reinforcement of the fundamental frequency effect is illustrated in Figures 4.3, 4.4 and 4.5.

The problem of the spurious effect comes when dealing with *polyphonic* notes detection because it creates an ambiguity between whether there is, for example, an A_4 note or an E_5 note, or both at the same time. Like in many note transcription schemes, the problem of distinguishing between fifths is usually the hardest. But nevertheless, as we will see in next chapter, the PSWS + low-pass filtering alone is quite good at this task without the use of any additional post-processing or top-down techniques. The nature of the musical notes produced by most instruments favor the discrimination capability of the PSWS. In particular, the most energetic partial is often the fundamental and the rest of the harmonics use to have decaying energies. This fact, and the reinforcement effect of the fundamental frequency, normally result in remarkable differences between the "real" fundamental tones and the "fake" spurious tones.

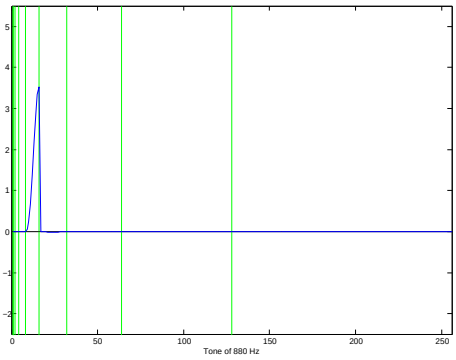


Figure 4.3: A tone of 880 Hz produces a DC pattern at level 4.

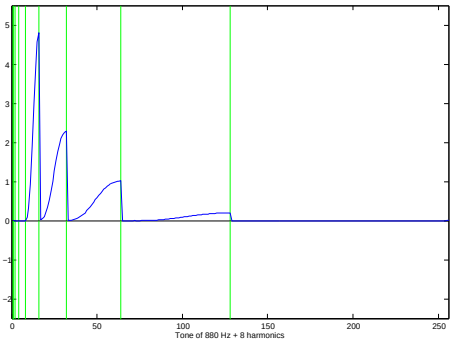


Figure 4.4: We add 8 harmonics with linearly decreasing amplitudes to the 880 Hz pure tone in Figure 4.3. Note that the first, the third and the fifth harmonics are the fundamentals of the A note in octaves 6, 7 and 8 respectively, producing the corresponding characteristic pattern at levels 5, 6 and 7. Note also the reinforcement effect: the other harmonics have contributed to raise the level of the coefficients at level 4 too.

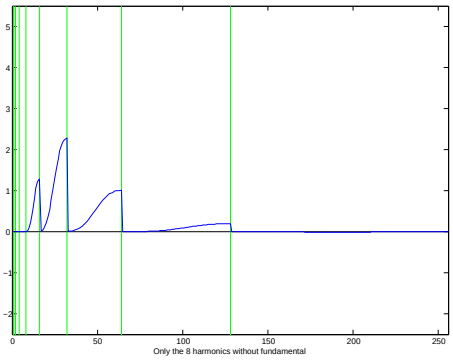


Figure 4.5: In this case the fundamental has been removed and only the 8 harmonics are present. Note that even so, the DC pattern at level 4 indicates the presence of the fundamental of an A_5 note, even though no tone with 880 Hz is present in the signal. When the fundamental is present, the two DC contributions sum, and coefficients at the corresponding level are raised accordingly, as we see in Figure 4.4.

The reason for the reinforcement effect is very easy to see when we use the Haar wavelet. Let us plot in Figure 4.6 the time representation of the Haar wavelet atoms for decompostion level 6 superimposed to the same tones of Figures 4.1 and 4.2. Note that we can intuitively estimate each of the 4 wavelet coefficients of this level of decomposition by just multiplying the wavelet and the tone signal point by point in each 64-points segment.

In Figure 4.6 (a) we can see that the period of the wavelet and the period of the signal coincide, and the 4 coefficients will be the same (in particular, they will take the maximum possible value). If we double the frequency, we see in Figure 4.6 (b) that all the coefficients are zero because the positive and the negative contributions cancel. This was predicted in Figure 2.10. Multiplying the fundamental frequency by 3 we get the representation in Figure 4.6 (c). Here we can see the reason why there exist a DC pattern of wavelet coefficients at level 6 for this frequency: there are 3 positive lobes and only 2 negative in each half of the window (6 "positive contributions" and 4 "negative contributions" in total), so the overall sum is positive. A similar situation can be found in Figures 4.6 (e) and 4.6 (g). But the positive balance is smaller and smaller and so the corresponding coefficients in Figures 4.1 (e) and 4.1 (g) are also smaller.

4.3.2 The *orthogonality* effect

In Figure 4.6 (a) we see that the corresponding wavelet coefficients will be maximum because the positive and the negative parts of the signal and the wavelet coincide and then the correlation is maximum. But what would have happened if the signal was a cosine instead of a sine? The answer is that positive and negative contributions cancel and the corresponding coefficient is zero. An

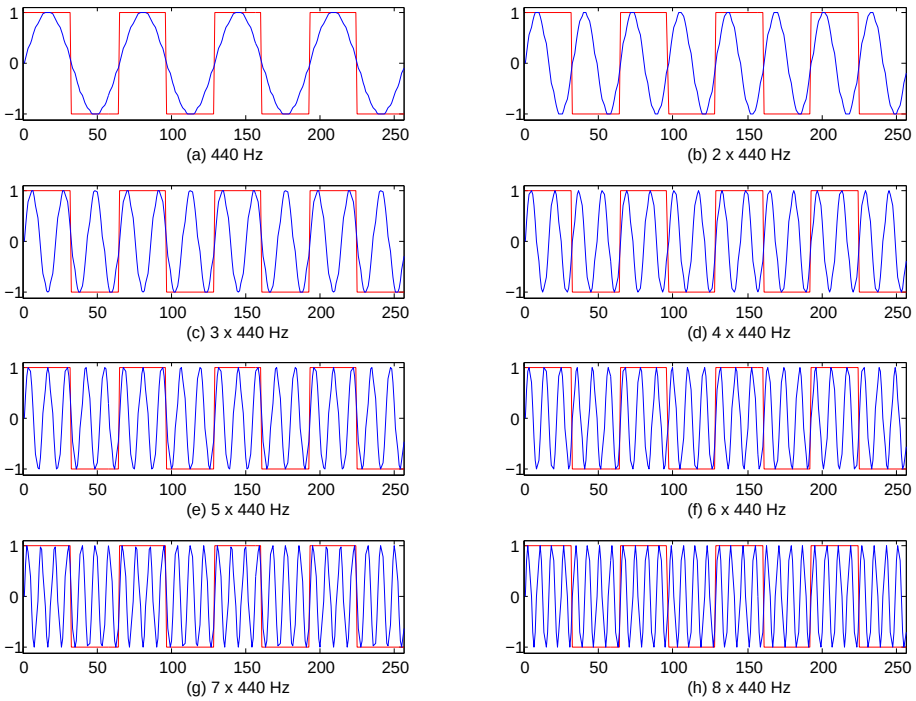


Figure 4.6: Haar wavelet atoms for decomposition level 6 (red) superimposed to the time domain representation of the same tones generated for Figures 4.1 and 4.2 (blue).

analogous problem arises if we try to use the DCT instead of the FFT to find frequency peaks: if the frequency of a cosine wave coincides exactly with the frequency that corresponds to one of the DCT frequency bins there will be a perfect spectral peak in this bin. But if the signal is not a cosine but a sine there will not be a peak in that bin anymore: surprisingly enough, there will be a *zero* exactly at this bin because the analyzed signal and the basis function are orthogonal, and the energy will be distributed among the neighboring bins.

This means that in real transformations, like the DCT and the Haar wavelet (and many other wavelet transforms), the phase of the signal can affect dramatically the amplitude of the coefficients. One solution to this problem could be the use of a complex wavelet in an analogous way as we use complex sines and cosines in the Fourier Transform, and take into account the modulus of the coefficients.

In this thesis we will not care about this "worst case" in which the signal and the wavelet are orthogonal, and we will use only real wavelets. The amplitude of the coefficients is not as important as the fact that they present a DC pattern. So, unless the phase of the signal is such that all these coefficients are *exactly* zero (which is not so probable), the DC pattern, whichever its amplitude be, will point out the presence of a given musical note. Nevertheless, in real examples of note detection —we will see some examples in next chapter— this issue will affect the performance. In particular, it makes more difficult the selection of normalization values and a suitable threshold to determine if a DC pattern belongs to a note or it is just an spurious effect. Therefore, the *orthogonality* effect must be taken into account if a reliable note detection system is to be built based on the ideas proposed in this thesis.

4.4 An example of PSWS

Now we have analyzed in depth the characteristics of each column in a Pitch-Synchronous Wavelet Spectrogram, we are ready to understand the spectrogram representations that we use for automatic note transcription in next chapter. As an example, we show in Figure 4.7 the PSWS of a 1760 Hz pure tone (the fundamental frequency of an A_6 note) sampled at a rate of 28160 Hz. It is informative to see that the energy in a wavelet spectrogram of a pure tone is not concentrated in one (or few neighboring) coefficients as in an FFT, but it is spread all over the different levels. In this case, the window length that coincides with the period of the signal in samples is that of level 5. From Figure 4.7 and taking also into account Figure 2.10, we can observe several important things:

- Level 5 coefficients are the ones that concentrate more energy, even though we have nonzero coefficients at levels 6, 7 and 8 too.
- Because of the particular frequency response of the Haar wavelet, coefficients of levels 1, 2, 3, 4 and 5 are practically zeros.
- Coefficients in level 5 are constant in each column and also across rows.

This only happens if the signal is purely periodic. In "real world" approximately periodic signals, coefficients are approximately constant within the corresponding level (the level whose analysis window coincides with the period of the signal), and change slowly across rows as time evolves. This will be seen in more detail in next chapter.

As coefficients in levels 6, 7 and 8 follow pseudo-periodic patterns with zero mean, we can apply a low-pass filter with a very low cut-frequency to the rows of the PSWS and take the energy of the spectrogram by raising the coefficients to the power of 2. The result of these 2 operations is shown in Figure 4.8. Let us call this representation *Filtered Pitch Synchronized Wavelet Spectrogram: FPSWS*. After this simple post processing, we can identify clearly the pure tone in the area of the spectrogram that corresponds to level 5 coefficients. All other signals with any period other than that of the fundamental frequency of the A note in every octave will be efficiently removed by the filter —which only lets pass DC or very slowly varying components—. In particular, harmonic signals will present periodic patterns with zero mean, and noise signals will present random patterns with approximately zero mean too.

Finally, we show in Figure 4.9 the same FPSWS of Figure 4.8 but seen "from above", as a contour field. Horizontal lines that delimit the boundaries between levels have been added. Note that each level will correspond to the fundamental frequency of the A note in one different octave.

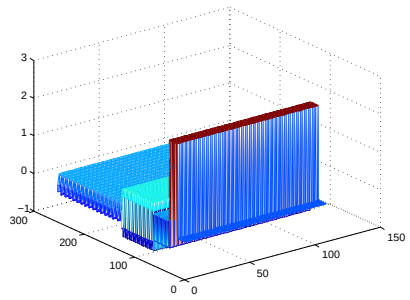


Figure 4.7: PSWS of a pure tone of 1760 Hz. Note that although there are nonzero coefficients at levels 5, 6, 7 and 8, only level 5 coefficients have nonzero mean within each column.

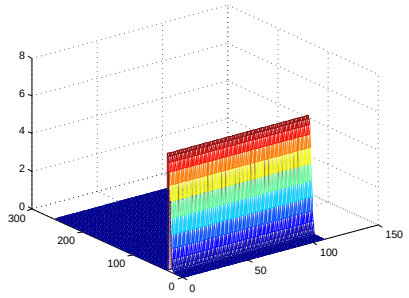


Figure 4.8: Filtered PSWS (FPSWS) of a pure tone of 1760 Hz. All coefficients with zero mean have been removed by a low pass filter that operates locally at each level across the columns.

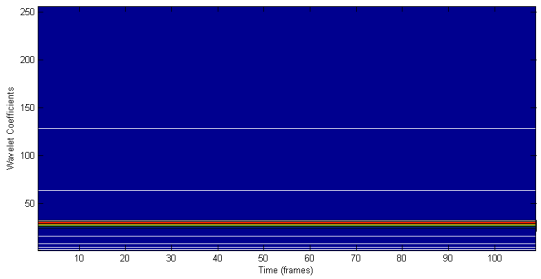


Figure 4.9: The same FPSWS in Figure 4.8 but seen from above as a contour field. The philosophy of this representation is that, whenever a pure tone of any frequency that coincides with the fundamental frequency of the A note at any octave is present in a signal, it will appear as nonzero coefficients in the corresponding level of the FPSWS contour field representation. In any other case, the FPSWS would be ideally flat.

Chapter 5

Application of PSWS for Music Transcription

In this chapter we will make use of the FPSWS representation directly as a note detector for musical signals. Each and every short frame of the signal is projected on a subspace of wavelets which are related to one particular musical note. The procedure is the following:

1. We take a segment of 256 points from the signal.

2. We resample the segment in such a way that the new sampling rate is a multiple of the fundamental period of the notes C at any octave. Thanks to the dyadic decomposition, the wavelet analysis window length will coincide with the period of the C note at the different octaves. Whenever a pure tone of a frequency that coincides with the fundamental of the C note at any octave is present in the signal, it will create a DC pattern in the spectrogram (PSWS), as seen in chapter 4.

3. We filter the spectrogram in order to remove the AC patterns and raise the remaining coefficients to the power of 2 to highlight the DC patterns.

4. We repeat these steps for the 12 semitones in a equal-tempered musical scale.

Musical notes, in general, present harmonically related spectral peaks that are approximately equally spaced in frequency. The method suggested above is useful to detect, in principle, only the fundamental frequency of a note, and reject as much as possible any other signal component: other harmonics, noise, percussive sounds, etc. One of the most important problems is that harmonics of one note usually coincide with fundamentals of other notes (i.e.: fifths, thirds, etc.) [15]. Anyway, as we saw in chapter 4, the harmonic tones tend also to reinforce the coefficients that represent the fundamental in a Haar wavelet

representation. The result is that the coefficients that represent the fundamental frequency of a note use to have much more energy than the coefficients that represent other partials.

5.1 Monophonic note detection

Let us analyze a signal that is a sequence of 24 notes, played in order from C_5 to B_6 in a wavetable-synthesis piano. The spectrogram of this signal is shown in Figure 5.1.

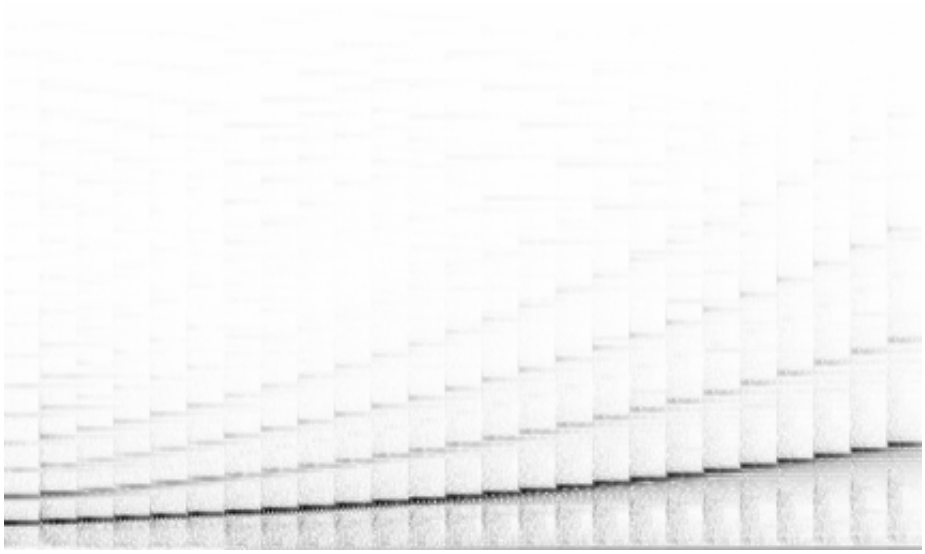


Figure 5.1: Spectrogram of a sequence of 24 notes, played in order from C_5 to B_6 in a wavetable-synthesis piano. FFT widow size: 512 samples; hop size: 256 samples.

In Figure 5.2 we can see the 12 FPSWS's that correspond to the 12 semitones of an equal-temperd scale for the signal on Figure 5.1. We can interpret the decomposition as the *projection* of the signal on the note C, $C\#$,D, $D\#$, E, etc. Whenever the fundamental frequency of these notes is present in the signal, the coefficients of the corresponding level of the FPSWS will be large, and the contour field will point out the octave to which the note belongs and the approximate moment in which it has been produced. Note that the horizontal axis represents the number of frames of the signal, but as we use a different sample rate for each note, these number of frames will be all different even though the signal is the same. In any case, the representation is normalized in time so that the events are approximately ordered in their correct moment of occurrence.

It is important to highlight here that it is not the aim of this thesis to go

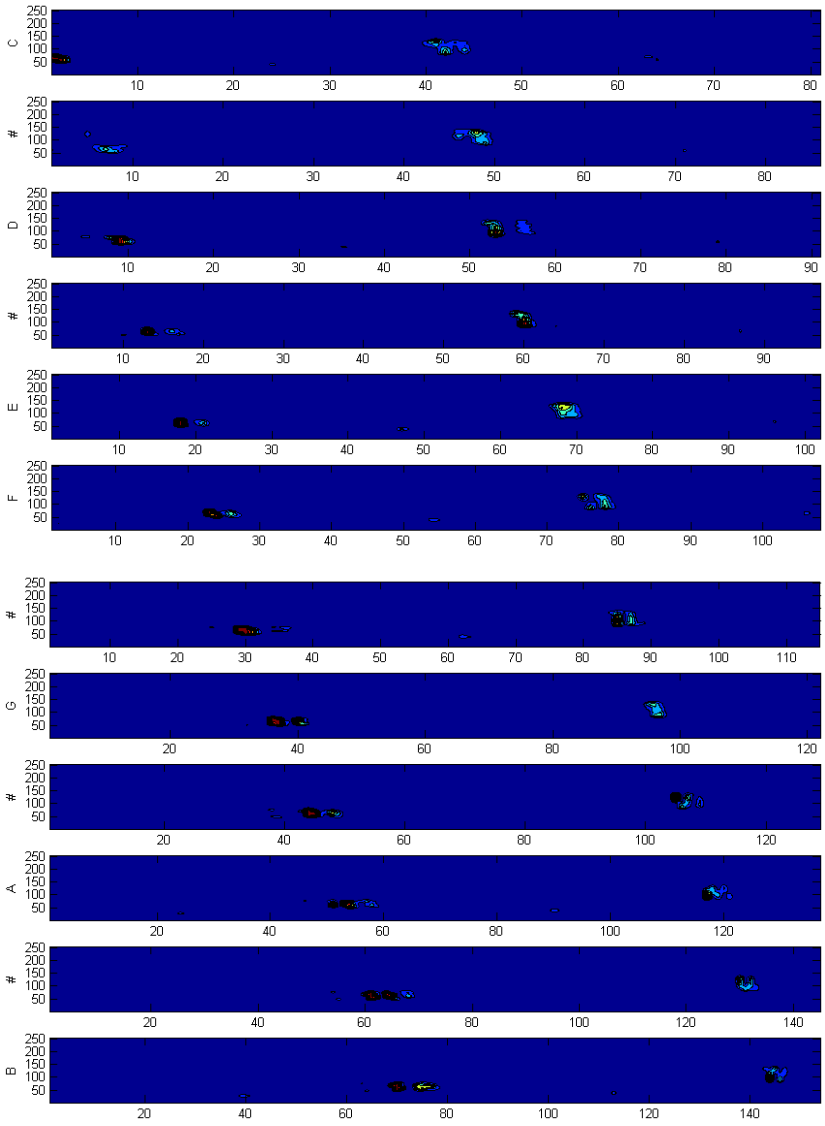


Figure 5.2: FPSWS's pitch synchronized with the fundamental frequency of the 12 semitones of an equal-tempered musical scale. The signal represented is a sequence of 24 notes played in order in a wavetable-synthesis piano from C_5 to B_6 (the same as in Figure 5.1.

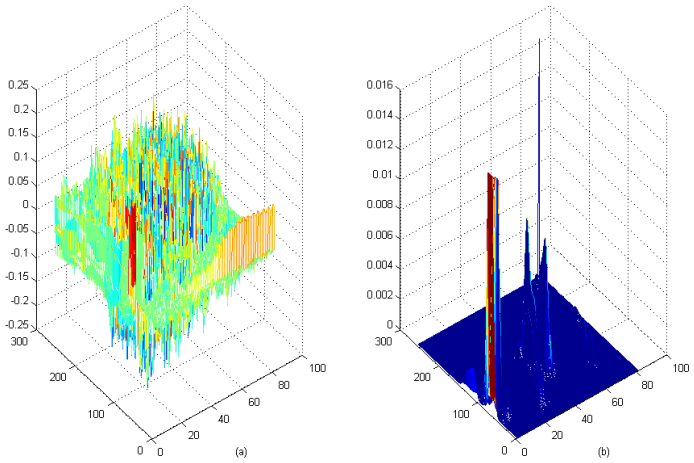


Figure 5.3: 3D representation of the PSWS and the FPSWS of the signal in Figure 5.2 for note C.

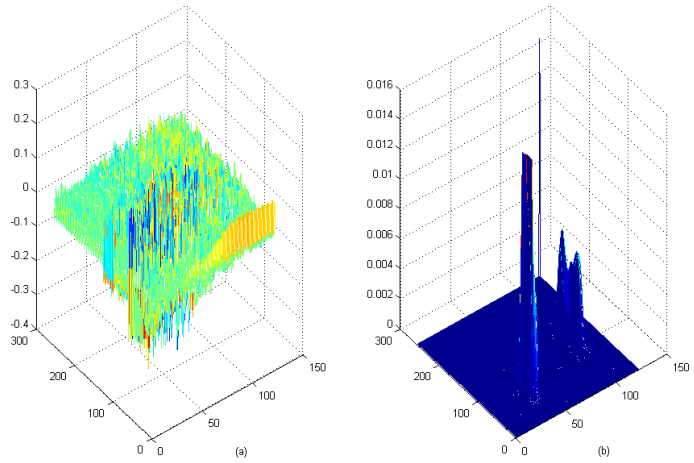


Figure 5.4: 3D representation of the PSWS and the FPSWS of the signal in Figure 5.2 for note G.

through the optimization of the parameters for note detection, and so, not too much care has been taken in the post-processing (filtering, normalization, etc.) of the PSWS. A simple low pass filter has been applied to the columns of the PSWS and the maximum of the 12 spectrograms has been used for normalization. It is expected that more localized or complex filters and normalizations and/or other window lengths would help in a better time-frequency localization and discrimination of notes.

We also show in Figures 5.3 and 5.4 the PSWS and the FPSWS in their 3D representation for notes C and G. The peak value that appear on the top corner of the representation is a reference value that allows for a normalized representation of the different FPSWS's. The DC patterns are not easy to see in the PSWS, but after filtering, they appear clearly in the FPSWS.

We can compare this results with those obtained using the Harmonic Pitch Class Profile (HPCP) representation developed by Gómez in [32]. One important difference between the two types of representation is that HPCP do not care about the octaves of the notes: it represents the harmonic content of a signal by adding the STFT spectral peaks that contribute to each note, no matter the octave. In FPSWS representations we can estimate the octave of the note because coefficients of lower levels than those that correspond to the fundamental frequency of the note detected are ideally zero, as explained in chapter 4.

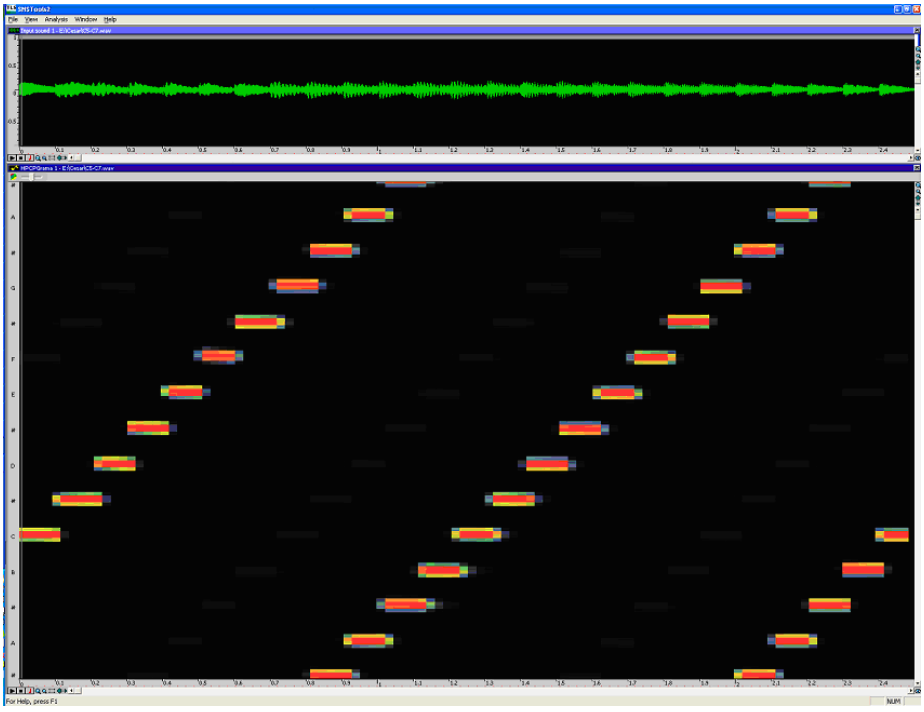


Figure 5.5: HPCP representation of the same signal analyzed in Figure 5.2.

Another difference has to do with the normalization of the representation. The spectral energy of the notes decay with time (we say that the note has a "tail"). In the case of the FPSWS representation, this tails are very short because we normalize the view by raising the coefficients to the power of 2 to distinguish between high values (that are likely to represent notes) and low values (that are usually spurious). By doing this, we remove the tails, which is an undesired effect, because we loose information about the duration of the notes. On the other hand, the HPCP representation is normalized in such a way that the decaying of the notes is not represented, and it just seems to distinguish between whether there is a note or not, regardless of its temporal envelope.

5.2 Polyphonic note detection

5.2.1 Case 1: wavetable synthesized piano

The approach of the note detector proposed in this chapter can be also applied to polyphonic note detection. At least for simple harmonic signals in the absence of strong perturbations.. The procedure for the generation of the FPSWS is exactly the same as in monophonic note detection, and it works based on the same principle: notes with the corresponding frequency will create DC patterns at the corresponding PSWS, and other notes or noise will create AC patterns in the corresponding PSWS. In the case of notes that share some of the partials (fifths, thirds, etc) the *reinforcement effect* studied in chapter 4 is important to highlight the fundamental frequency of the "true" note against its fifth, for example.

We have created a test signal using a wavetable-synthesis piano that consists of a sequence of 4 chords: C - Cm7 in the 5th octave, and then the same 2 chords in the 6th octave. The chords are played in whole notes in a 4/4 time signature. This means that there are 4 bars with one different chord in each. The spectrogram of this signal is shown in Figure 5.6.

In Figure 5.7 we can see the FPSWS of this signal for the 12 semitones of the scale. In this case, we did not raise the coefficients to the power of two for the representation, and we just took their absolute value. This way we have a better idea of the tails of the notes, but there is more "background noise". We see that all the notes that make up this chords are correctly detected, but there are some false positives: the FPSWS of the C# or the F notes should be flat, for example. False positives could be removed by raising the coefficients to the power of 2, and/or making a proper adjustment of the threshold, but then we would loose the "tails" of the notes.

The FPSWS shows the coefficients that represent the fundamental frequency of the notes, reinforced by the harmonics of the notes. This means that these coefficients have much more energy than the others, which is consistent with what we could expect taking into account the *reinforcement effect* and the fact that usually the fundamental is the most energetic partial.

The representation of the detected notes is not "continuous" in time due

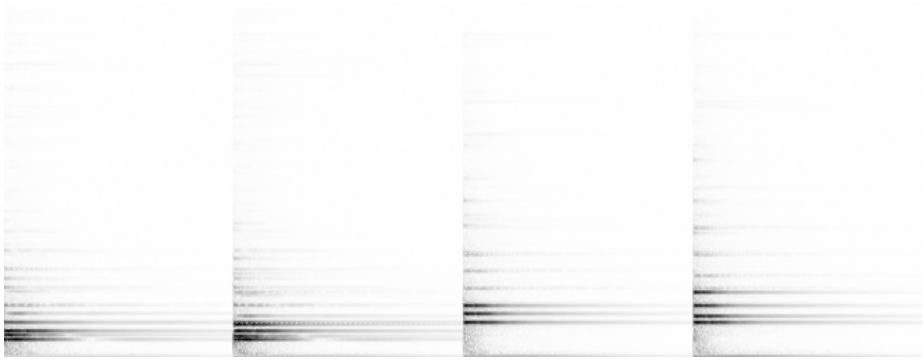


Figure 5.6: Spectrogram of a sequence of the chords C - Cm7 in the 5th octave, and C - Cm7 in the 6th octave played in in a wavetable-synthesis piano. FFT widow size: 512 samples; hop size: 256 samples.

to the *orthogonality effect*: the coefficients are approximately constant within the same level in the same frame, but the phase varies slowly from frame to frame, and so the coefficients can be positive, cross the zero level, and then negative, in an approximately periodic fashion, until the note disappears. Note also that we can guess the octave of the notes by looking at the coefficients at the corresponding levels.

In Figure 5.8 we see the corresponding HPCP representation. We can see that it correctly detects the notes at the moment they are produced, without any false positives, but the duration of the notes (in theory, the same for all) established by the representation varies substantially. As said before, no information about the octave of the notes is provided.

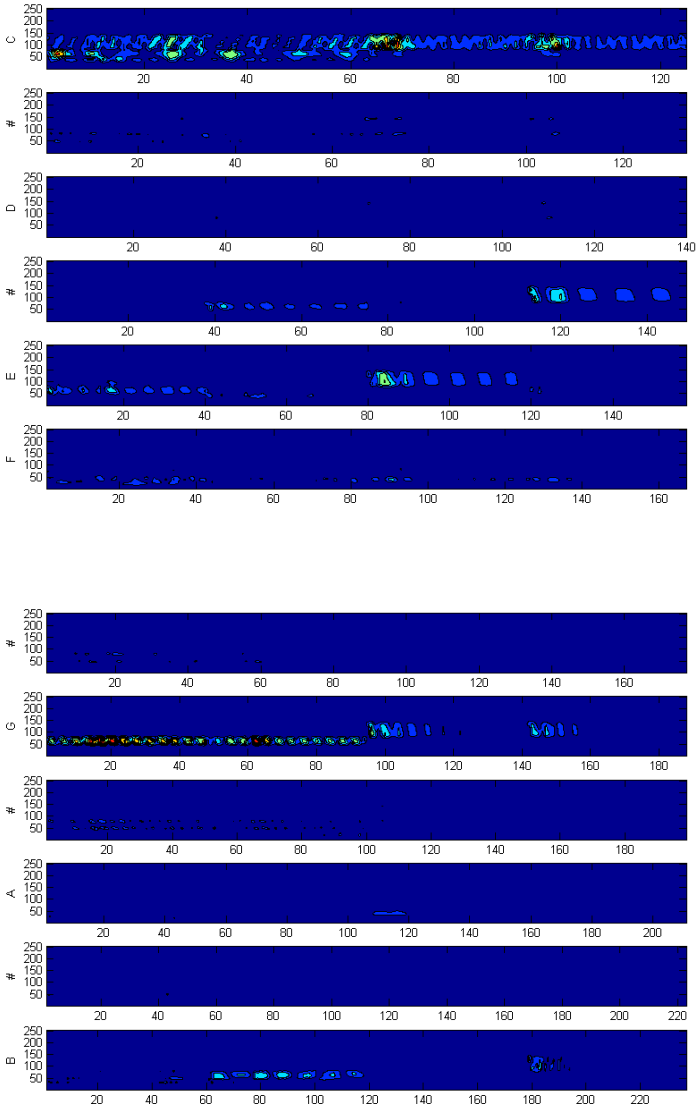


Figure 5.7: FPSWS's of the signal on Figure 5.6.

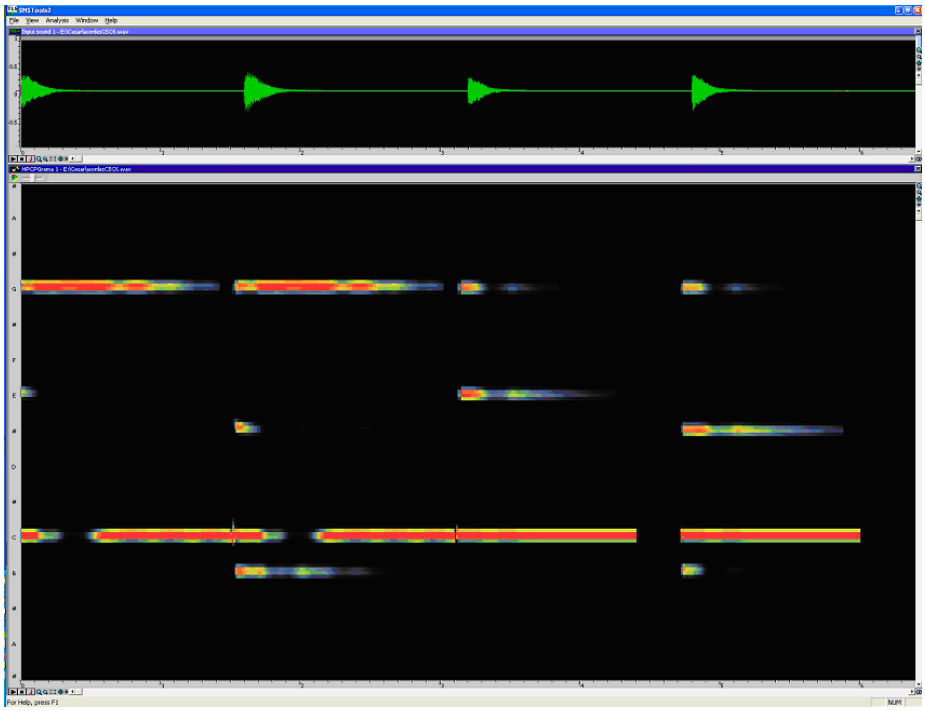


Figure 5.8: HPCP representation of the signal on Figure 5.6.

5.2.2 Case 2: wavetable synthesized piano + drums

Let us now analyze the same signal in Figure 5.6 in a not-so-ideal situation. We have added a charles at quarter notes from the first bar, a snare drum at half notes in the second and fourth bar, and a bass drum at half notes in the third and the fourth bar. The spectrogram of this signal is shown in Figure 5.9

Figure 5.9: Spectrogram of a sequence of the chords in Figure 5.6 plus a drum pattern in which the notes coincide exactly first with the charles (first bar), then with the charles and the snare drum (second bar), then only with the charles and the snare drum (third bar), and finally with the charles, the bass drum and and snare drum (fourth bar). FFT widow size: 512 samples; hop size: 256 samples.

In Figure 5.10 we see that the bass drum and specially the snare drum affect quite a lot the detection of the notes, creating false positives. The charles does not seem to affect too much the detection. Nevertheless, it is possible to some extent to distinguish the notes from the drum strikes by taking into account that these last are shorter in duration and appear in every FPSWS with a similar pattern.

In Figure 5.11 we can see the HPCP representation of the same signal. The representation is also quite leakaged by the drum strikes, and again the only hint to distinguish between notes and drum strikes seems to be the sustain in time of the notes. But is interesting to note that in the HPCP representation, there are false positives that appear also as clear sustained notes along time. It is the case of a clear "ghost" F note that seems to be present most of the time, and another "ghost" note between D and $D\#$, for example.

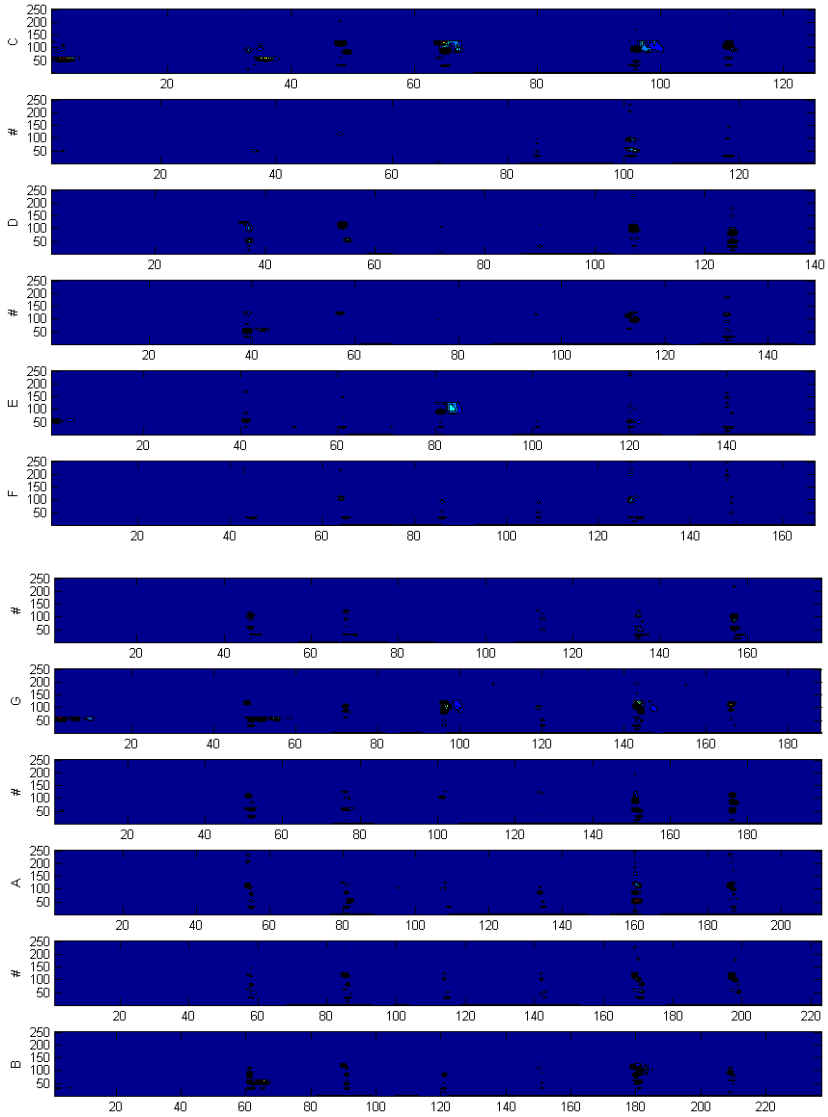


Figure 5.10: FPSWS's of the signal on Figure 5.9.

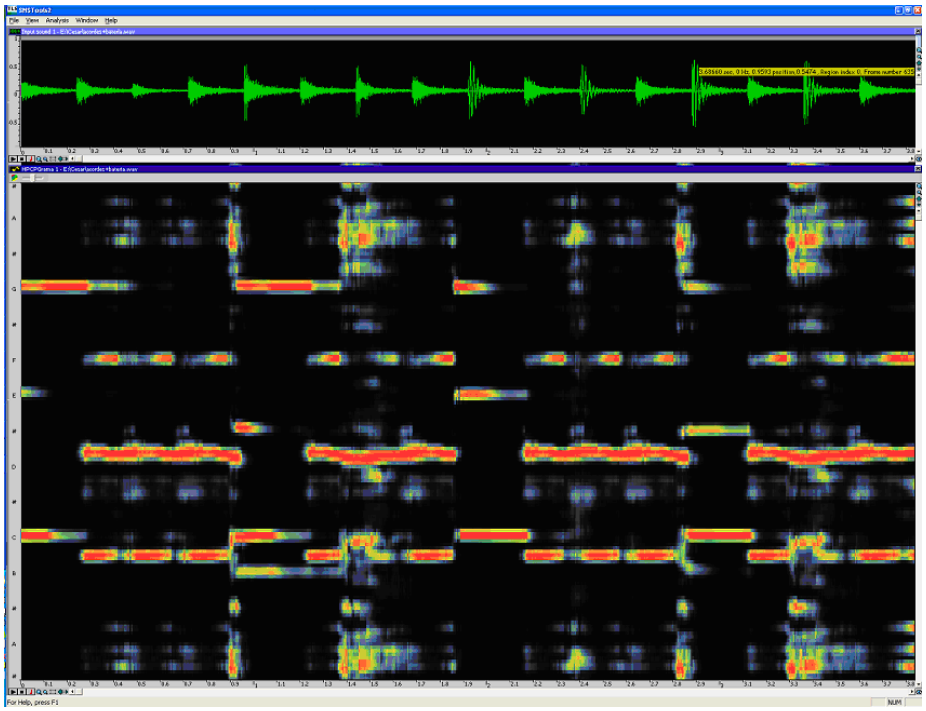


Figure 5.11: HPCP representation of the signal on Figure 5.9.

5.2.3 Case 3: real piano

So far, we have used only a wavetable-synthesis piano for our analyses. In the next example, an excerpt taken from a recording of the beginning of Beethoven's "For Elisa" played in a real piano is used. The spectrogram is shown in Figure 5.12.

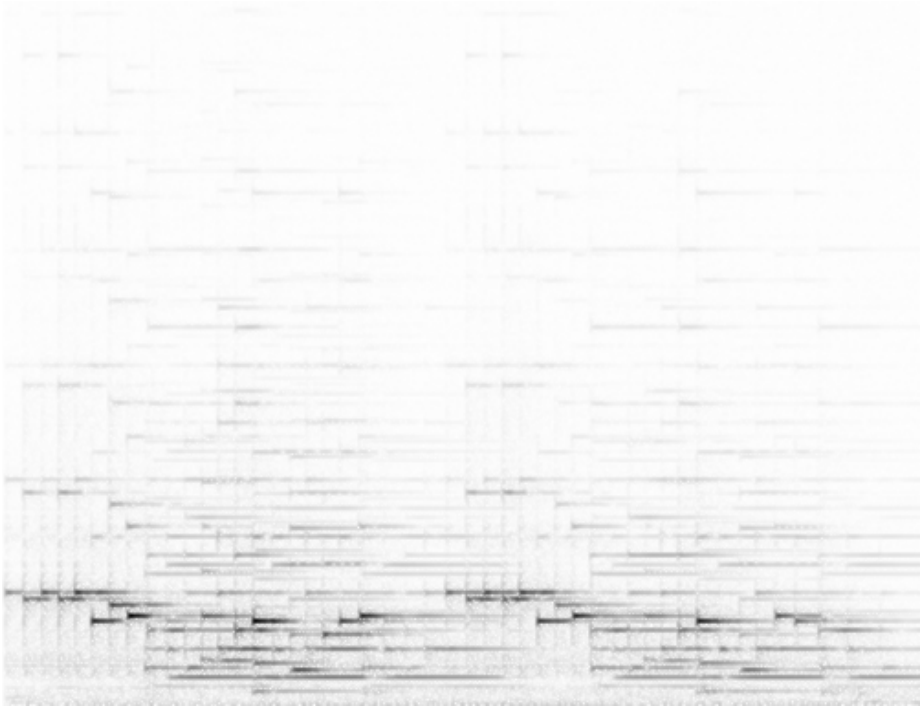


Figure 5.12: Spectrogram of the beginning of Beethoven's "For Elisa" played in a real piano. FFT widow size: 2048 samples; hop size: 512 samples. Only low frequencies are represented in the picture.

The excerpt was transposed 2 octaves in order to center the signal in frequency in the FPSWS. The results are shown in Figure 5.13. There are some inaccuracies, specially in the A and E bass notes, but the overall performance is not too bad: in general, the notes that are not present in the signal do not appear in the representation, and most of the ones that are present appear somehow in the representation.

The HPCP representation is shown in Figure 5.14. We see it is much more "clean" and precise than the FPSWS. It is expected that the use of more accurate filters and normalization of the signal for representation, and the use of other wavelet transforms that could solve the *orthogonality effect* could help in achieving a better detection and representation of the notes in the FPSWS.

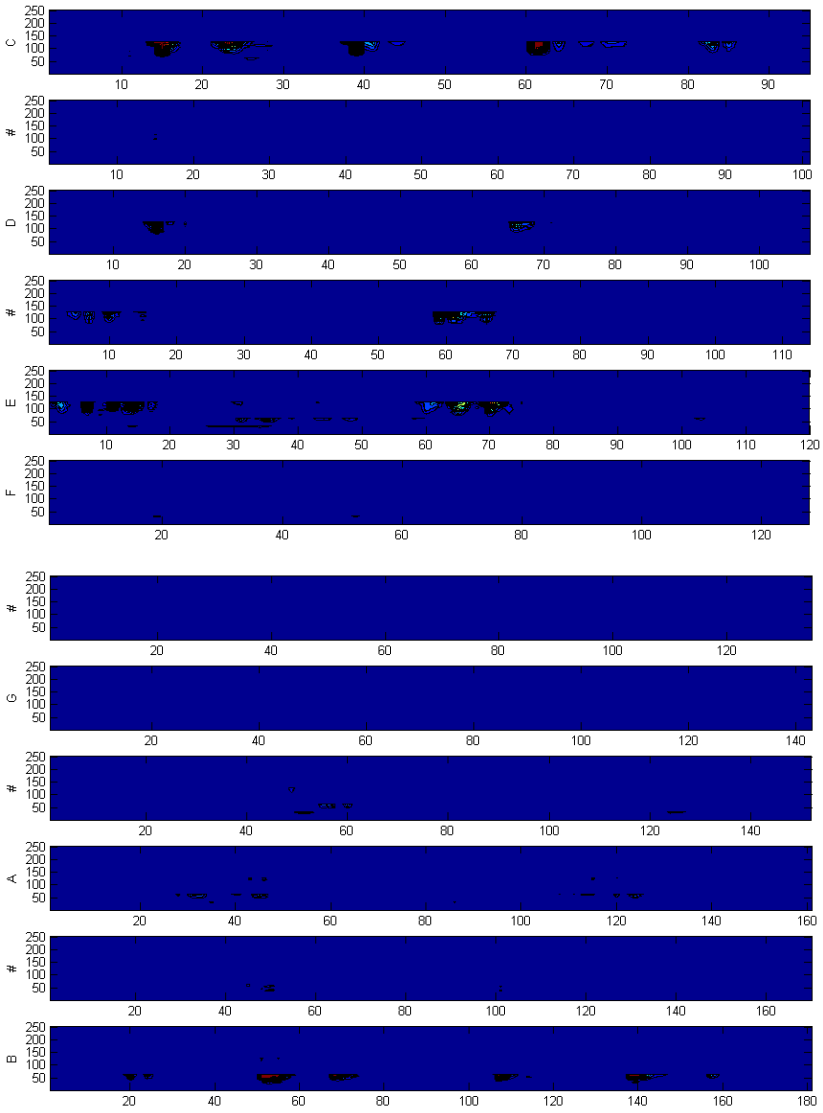


Figure 5.13: FPSWS's of the signal on Figure 5.12. The excerpt was transposed 2 octaves in order to center the signal in frequency in the FPSWS representation.

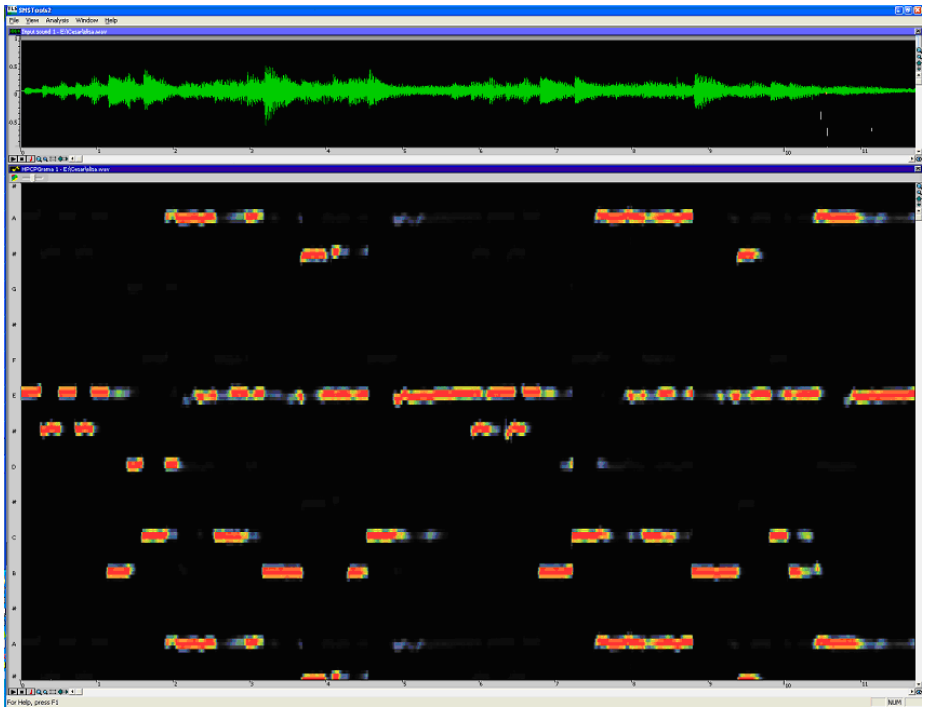


Figure 5.14: HPCP representation of the signal on Figure 5.12.

5.2.4 Other signals

We have used the system also with "real-world" guitar signals and the results are similar to those provided with the real piano: the inaccuracies grow as polyphony grows too. For instruments which notes do not present a precise location in frequency, like the human voice, the results are not good: because of small variations in the pitch (vibrato, for example) nonzero coefficients appear not only in the FPSWS of the corresponding notes, but also in the neighboring FPSWS's. In any case, no extensive experiments have been conducted yet for different kinds of music signals, and the purpose of the examples of this chapter is only to illustrate the performance of the method in nearly ideal situations.

It is clear that the reliability of the method lies heavily on the precise stability of the pitch in the corresponding theoretical frequencies. When small deviations from these positions occur, there is a tolerance margin in which the corresponding wavelet coefficients still draw nearly continuous patterns. But outside this margin, the coefficients either draw pseudo-random AC patterns and so they are removed by the filter, or they create DC patterns in FPSWS's corresponding to neighboring notes, thus creating false positives.

One solution to this problem would be the use of, say, 36 PSWS's instead of only 12 to achieve a resolution of 1/3 of semitone, for example. This way, the system would be able to detect small excursions of the pitch from its ideal locations. This idea is implemented in the HPCP calculation that has been used for comparison throughout this chapter [32]. In particular, a resolution of 1/10 of semitone is considered decent enough to distinguish frequency details like vibrato, glissando, etc. and this is the resolution used in the HPCP representations presented in this chapter. The disadvantage of this solution is that the more resolution we use, the more computationally intensive will be the system.

The design and optimization of a fully working note detection system that takes advantage of the ideas presented in this thesis is left as future work, and it will have to take into account the *orthogonality effect* described in chapter 4 and the development of an optimal way to detect and localize in time and frequency the DC patterns in the PSWS. It is also interesting to consider the generation of a larger number of PSWS's to achieve more frequency resolution for the cases in which small deviations from the central frequency occur, like in [32], in order for the system to be useful for non-stable pitch musical instruments.

Chapter 6

Conclusions and future work

The motivation for the development of this thesis, was somehow atypical: it did not come out from the need to solve a concrete problem; on the contrary, it came out from a couple of intuitive ideas that related the dyadic wavelet decomposition and the theoretical frequency location of the fundamental frequencies of musical notes, as it was explained in the introduction of the thesis. How to exploit these ideas in a useful way to solve some problem only came later.

In this sense, the objective of this work was not to make an exhaustive state of the art about a given problem, propose a solution, and compare it to existing methods. The objective was more to achieve a better understanding of wavelets and multiresolution analysis, and study their possible applications in relation to some well-known problem in music processing. The time-frequency tradeoff of the STFT was described in detail and some general guidelines about how wavelets could work better than the Fourier transform in some concrete situations were suggested.

The idea of creating a subspace in which musical notes are somehow orthogonal has been also proposed in this thesis. It is clear that the Fourier representation of harmonic signals is very useful to describe them: the location of the partials and the spectral envelope give very useful information about the pitch and the timbre of musical sounds. But as long as different notes present partials at the same frequency locations, the Fourier representation is limited in what source separation or note transcription is concerned. We have presented here a novel approach to characterize the frequency content of a musical signal taking into account some prior knowledge about the theoretical location of the fundamental frequency of musical notes and the harmonic structure of the notes themselves to build a set of wavelets adapted to the musically most important frequencies. It has been shown that we can generate a wavelet-based representation, the FPSWS, in such a way that musical notes are approximately orthogonal. This way, they can be detected even in a polyphonic mix by pro-

jecting them on the proper subspace. This note detection method seems to be reasonably robust against percussive sounds distortion and the coinciding partials problem, and it appears as a promising technique, even though there is still a lot of work to do in the optimization of the method.

To be fully useful, the note transcription system proposed should take into account the *orthogonality* effect described in section 4.3.2 and consider the use of more refined techniques to find and represent the DC components inside a PSWS. But undoubtedly, the most important challenge is to solve the problem of musical instruments that generate notes with unstable pitches. One solution proposed in chapter 5 is to increase the number of PSWS to achieve a resolution higher than just one semitone. Another solution could be the use of some kind of pitch detection scheme to adapt the length of the wavelets dynamically to small excursions of the frequencies of the notes around their theoretical positions in frequency. The same frame-to-frame temporal continuity of wavelet coefficients that draws the typical DC patterns could be exploited also to design a phase-locked system that follows these small variations of pitch.

This idea of a phase-locked digital analysis system could be very interesting also for peak tracking purposes in the extraction of sinusoidal components, for example. Phase-locked loops are widely used in the analog world, but it seems not to have been so used in the digital world, possibly because of the difficulties of an accurate tracking of the phase changes across frames described in chapter 1, specially for high frequencies. As it has been stated throughout the thesis, multiresolution analysis can give much more accurate information about the phase changes between coefficients than the STFT, thus making it feasible the development of these phase-locked pitch representations. Robustness against percussive sounds and noise could also benefit a lot from a phase-locked approach, since non-harmonic components could be naturally rejected.

Finally, it is expected that well-known bottom-up and top-down approaches for polyphonic music transcription or pitch class profile classification could benefit from a better detection of notes at the very low level. In this sense, the natural continuation of this work would be the optimization of the note detection capability using the PSWS following the ideas just suggested, and also to apply higher level A.I. techniques to refine the results and compare them to other state-of-the-art schemes.

References

[1] S. Abdallah and M. Plumbley, "Unsupervised Analysis of Polyphonic Music by Sparse Coding," *Neural Networks, IEEE Transactions on*, vol. 17, no. 1, pp. 179–196, 2006.

[2] T. Björkvald and E. Svensson, "Semi-automatic Music Creation using the Continuous Wavelet Transform and Markov Chains," May 2004.

[3] F. Charpentier and M. Stella, "Diphone synthesis using an overlap-add technique for speech waveforms concatenation," *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'86.*, vol. 11, 1986.

[4] F. Auger, P. Flandrin, and S. GE44-LRTI, "Improving the readability of time-frequency and time-scalerepresentations by the reassignment method," *Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on]*, vol. 43, no. 5, pp. 1068–1089, 1995.

[5] G. Peeters and X. Rodet, "SINOLA: A New Analysis/Synthesis Method using Spectrum Peak Shape Distortion, Phase and Reassigned Spectrum," *Proc. ICMC*, vol. 99, 1999.

[6] R. A. García and K. M. Short, "Signal Analysis Using the Complex Spectral Phase Evolution (CSPE) Method," *120th AES Convention*, 2006.

[7] R. Kronland-Martinet, "The wavelet transform for analysis, synthesis, and processing of speech and music sounds." *COMP. MUSIC J.*, vol. 12, no. 4, pp. 11–20, 1988.

[8] S. Levine, T. Verma, and J. Smith III, "Multiresolution sinusoidal modeling for wideband audio with modifications," *Proc. Int. Conf. Acoustics, Speech, and Signal Processing, Seattle*, 1998.

[9] P. Vaidyanathan, "Quadrature mirror filter banks, M-band extensions and perfect-reconstruction techniques," *ASSP Magazine, IEEE [see also IEEE Signal Processing Magazine]*, vol. 4, no. 3, pp. 4–20, 1987.

[10] J. Brown, *Calculation of a Constant Q Spectral Transform*. Vision and Modeling Group, Media Laboratory, Massachusetts Institute of Technology, 1990.

[11] S. Mallat, *A Wavelet Tour of Signal Processing*. Academic Press, 1999.

[12] A. Klapuri, "Signal Processing Methods for the Automatic Transcription of Music," Ph.D. dissertation, 2004.

[13] (2007) Amara's wavelet page. [Online]. Available: http://www.amara.com/current/wavelet.html

[14] I. Daubechies, *Ten Lectures on Wavelets*. Society for Industrial & Applied Mathematics, 1992.

[15] A. Klapuri and M. Davy, *Signal Processing Methods for Music Transcription*. Springer, 2006.

[16] S. Hainsworth, "Techniques for the Automated Analysis of Musical Audio," Ph.D. dissertation, 2004.

[17] V. Lesser, S. Nawab, and F. Klassner, "IPUS: An architecture for the integrated processing and understanding of signals," *Artificial Intelligence*, vol. 77, no. 1, pp. 129–171, 1995.

[18] A. Sterian, M. Simoni, and G. Wakefield, "Model-Based Musical Transcription," *Proc. International Computer Music Conference*, 1999.

[19] S. Levine, T. Verma, and J. Smith III, "Alias-free, multiresolution sinusoidal modeling for polyphonic, wideband audio," *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz, NY*, 1997.

[20] P. Polotti and G. Evangelista, "Multiresolution Sinusoidal/Stochastic Model for Voiced-Sounds," *Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFX-01). Limerick, Ireland*, 2001.

[21] A. Kobzantsev, D. Chazan, and Y. Zeevi, "Automatic Transcription of Piano Polyphonic Music," *Image and Signal Processing and Analysis, 2005. ISPA 2005. Proceedings of the 4th International Symposium on*, pp. 414–418, 2005.

[22] H. Jang and J. Park, "Multiresolution sinusoidal model with dynamic segmentation for timescale modification of polyphonic audio signals," *Speech and Audio Processing, IEEE Transactions on*, vol. 13, no. 2, pp. 254–262, 2005.

[23] D. Newland, "Harmonic Wavelet Analysis," *Proceedings: Mathematical and Physical Sciences*, vol. 443, no. 1917, pp. 203–225, 1993.

[24] D. Newland", "Harmonic and Musical Wavelets," *Proceedings: Mathematical and Physical Sciences*, vol. 444, no. 1922, pp. 605–620, 1994.

[25] D. Newland, "Harmonic wavelets in vibrations and acoustics," *Philosophical Transactions: Mathematical, Physical and Engineering Sciences*, vol. 357, no. 1760, pp. 2607–2625, 1999.

[26] G. Evangelista and S. Cavaliere, "Discrete frequency warped wavelets: theory and applications," *Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on]*, vol. 46, no. 4, pp. 874–885, 1998.

[27] G. Evangelista and S. Cavaliere, "Auditory Modeling via Frequency Warped Wavelet Transform," *inProc. EUSIPCO98*, pp. 117–120, 1998.

[28] P. Polotti and G. Evangelista, "Fractal Additive Synthesis via Harmonic-Band Wavelets," *Computer Music Journal*, vol. 25, no. 3, pp. 22–37, 2001.

[29] P. Polotti and G. Evangelista, "Analysis and Synthesis of Pseudo-Periodic 1/f-Like Noise by Means of Wavelets with Applications to Digital Audio," *EURASIP Journal on Applied Signal Processing*, vol. 1, pp. 1–14, 2001.

[30] S. Levine, "Audio Representations for Data Compression and Compressed Domain Processing," Ph.D. dissertation, Stanford University, December 1998.

[31] J. Bello, "Towards the automated analysis of simple polyphonic music: A knowledge-based approach," Ph.D. dissertation, Univ. of London, 2003.

[32] E. Gomez, "Tonal Description of Music Audio Signals," Ph.D. dissertation, Universitat Pompeu Fabra, 2006.