A Variable Resolution transform for music analysis

Aliaksandr Paradzinets and Liming Chen

A Research Report, Lab. LIRIS, Ecole Centrale de Lyon

Ecully, June 2009

A Variable Resolution transform for music analysis

Aliaksandr Paradzinets and Liming Chen
Laboratoire d'InfoRmatique en Images et Systemes d'information (LIRIS),
Département MI, Ecole Centrale de Lyon,
University of Lyon
36 avenue Guy de Collongue, 69134 Ecully Cedex, France;
E-mail : {aliaksandr.paradzinets; liming.chen}@ec-lyon.fr

Abstract— This paper presents a novel music representation using a Variable Resolution Transform (VRT) which is particularly well adapted for music audio analysis. The VRT is inspired by continuous wavelet transform and it applies different wavelet function at different scale. This method enables a good flexibility of the transform in order to follow log scale of musical note frequencies and at the same time to maintain good time and frequency resolution. As an example of application of this novel VRT, a multiple f0 detection algorithm is presented and evaluated showing convincing results. Furthermore, a direct comparison with the FFT applied to the same algorithm is also provided.

Index Terms— music representation, music analysis, Variable resolution transform, multiple fundamental frequency estimation

I. INTRODUCTION

As a major product for entertainment, there is a huge amount of digital musical content produced, broadcasted, distributed and exchanged. Consequently there is a rising demand for better ways of cataloging, annotating and accessing these musical data. This in turn has motivated intensive research activities for music analysis, content-based music retrieval, etc.

The primary stage in any kind of audio signal processing is an effective audio signal representation. While there exists some algorithms performing music data analysis in the time domain as for example some beat detection algorithms, the majority of music processing algorithms perform their computation in the frequency domain, or a time-frequency representation, to be exact. So, the performance of all further steps of processing is strictly dependent on the initial data representation.

As compared to a vocal signal, a music signal is likely to be more stationary and owns some very specific properties in terms of musical tones, intervals, chords, instruments, melodic lines and rhythms, etc. [1]. While many effective and high performance music information retrieval (MIR) algorithms have been proposed [2-9], most of

these works unfortunately tend to consider a music signal as a vocal one and make use of MFCC-based features which are primarily designed for speech signal processing. Mel Frequency Cepstrum Coefficients (MFCC) was introduced in the 60's and used since that time for speech signal processing. The MFCC computation averages spectrum in sub-bands and provides the average spectrum characteristics. Whereas they are inclined to capture the global timbre of a music signal and claimed to be of use in music information retrieval [10; 11], they cannot characterize the aforementioned music properties as needed for perceptual understanding by human beings and quickly find their limits [12]. Recent works suggest combining spectral similarity descriptors with high-level analysis in order to overcome existing ceiling [13].

The Fast Fourier Transform and the Short-Time Fourier Transform have been the traditional techniques in audio signal processing. This classical approach is very powerful and widely used owing to its great advantage of rapidity. However, a special feature of musical signals is the exponential law of notes' frequencies. The frequency and time resolution of the FFT is linear and constant across the frequency scale while the human perception of a sound is logarithmic according to Weber-Fechner law (including loudness and pitch perception). Indeed, as it is well known, the frequencies of notes in equally-tempered tuning system in music follow an exponential law (with each semi-tone the frequency is increased by a factor of $2^{1/12}$). If we consider a frequency range for different octaves, this frequency range is growing as the number of octave increases. Thus, to cover a wide range of octaves with a good frequency grid large sized windows are necessary in the case of FFT; this affects the time resolution of the analysis. On the contrary, the use of small windows makes resolving frequencies of neighboring notes in low octaves almost impossible. The ability of catching all octaves in music with the same frequency resolution is essential for music signal analysis, in particular construction of melodic similarity features. In this paper, we propose a new music signal analysis technique by variable-resolution transform (VRT) particularly suitable to music signal.

Our VRT is inspired by Continuous Wavelet Transformation (CWT) [14] and specifically designed to overcome the limited time-frequency localization of the Fourier-Transform for non-stationary signals. Unlike classical FFT, our VRT depicts similar properties as CWT, i.e. having a variable time-frequency resolution grid with a high frequency resolution and a low time resolution in low-frequency area and a high temporal/low frequency resolution on the other frequency side, thus behaving as a human ear which exhibits similar time-frequency resolution characteristics [15].

The remainder of this paper is organized as follows. Section II overviews related music signal representations.

Our variable resolution transform is then introduced in section III. The experiments and the results are discussed in section IV. Finally, we conclude our work in section V.

II. RELATED WORKS

There are plenty of works in the literature dedicated to musical signal analysis. In this section, we propose first to compare the popular FFT with wavelet transform on the basis of desirable properties for music signal analysis and then overviews some other transforms and filter banks so far proposed in the literature.

A. Time-frequency transforms: FFT vs WT

The common approach is the use of FFT (Fast Fourier Transform) which has become a de-facto standard in music information retrieval community. The use of FFT seems straightforward in this field and relevance of its application for music signal analysis is almost never motivated.

There are some works in music information retrieval attempting to make use of wavelet transform as a novel and powerful tool in musical signal analysis. However, this new direction is not very well explored. [8] proposes to rely on discrete wavelet transform for beat detection. Discrete packet wavelet transform is studied in [15] to build time and frequency features in music genre classification. In [16], wavelets are also used for automatic pitch detection.

As it is well known, Fourier transform enables a spectral representation of a periodic signal as a possibly sum of a series of sines and cosines. While Fourier transform gives an insight into the spectral properties of a signal, its major disadvantage is that a decomposition of a signal by Fourier transform has infinite frequency resolution and no time resolution. It means that we are able to determine all frequencies in the signal, but without any knowledge about when they are present. This drawback makes Fourier transform to be perfect for analyzing stationary signals but unsuitable for irregular signals whose characteristics change in time. To overcome this problem several solutions have been proposed in order to represent more or less the signal in time and frequency domains.

One of these techniques is windowed Fourier transform or short-time Fourier transform. The idea behind is to bring time localization into classic Fourier transform by multiplying the signal with an analyzing window. The problem here is that the short-time discrete Fourier transform has a fixed resolution. The width of the windowing function is a tradeoff between a good frequency resolution transform and a good time resolution transform. Shorter window leads to smaller frequency resolution but higher time resolution while larger window leads to greater frequency resolution but lower time resolution. This phenomenon is related to Heisenberg's uncertainty principle which says that

$$\Delta t \sim \frac{1}{\Delta f} \tag{1}$$

where Δt is a time resolution step and Δf is a frequency resolution step.

Remember that in our work the main goal is music analysis. In this respect, we consider a rather music-related example which illustrates specificities of musical signals. As it is known, the frequencies of notes in equally-tempered tuning system in western music follow a logarithmic law, i.e. adding a certain interval (in semitones) corresponds to multiplying a frequency by a given factor. For an equally-tempered tuning system a semitone is defined by a frequency ratio of $2^{1/12}$. So, the interval between two frequencies is

$$n = 12 \cdot \log_2\left(\frac{f_2}{f_1}\right) \tag{2}$$

If we consider a frequency range for different octaves, it is growing as the number of octave is higher. Thus, applying the Fast Fourier Transform we either lose resolution of notes in low octaves (Figure 1) or we are not able to distinguish high-frequency events which are closer in time and have shorter duration.

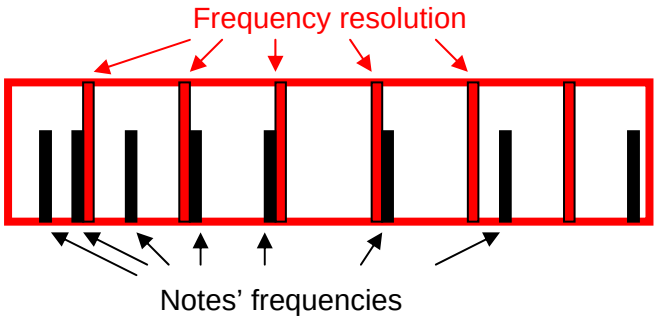

Figure 1. Mismatch of note frequencies and frequency resolution of the FFT.

Time-frequency representation, which can overcome resolution issues of the Fourier transform is ***Wavelet transform***. Wavelets (literally "small waves") are a relatively recent instrument in modern mathematics. Introduced about 20 years ago, wavelets have made a revolution in theory and practice of non-stationary signal analysis [14; 17]. Wavelets have been first found in the literature in works of Grossmann and Morlet [18]. Some ideas of wavelets partly existed long time ago. In 1910 Haar published a work about a system of locally-defined basis functions. Now these functions are called Haar wavelets. Nowadays wavelets are widely used in various signal analysis, ranging from image processing, analysis and synthesis of speech, medical data and music [16; 19].

Continuous wavelets transform of a function $f(t) \in L^2(R)$ is defined as follows:

$$W(a,b)=\frac{1}{\sqrt{|a|}}\int_{-\infty}^{\infty} f(t)\psi^{*}\left(\frac{t-b}{a}\right)dt \qquad (3)$$

where $a, b \in R, a \neq 0$.

In the equation (3) $\psi(t)$ is called basic wavelet or mother wavelet function (* stands for complex conjugate). Parameter *a* is called wavelet scale. It can be considered as analogous to frequency in the Fourier transform. Parameter *b* is localization or shift. It has no correspondence in the Fourier transform.

One important thing is that the wavelet transform does not have a single set of basis functions like the Fourier transform. Instead, the wavelet transform utilizes an infinite set of possible basis functions. Thus, it has an access to a wide range of information including the information which can be obtained by other time-frequency methods such as Fourier transform.

As explained in brief introduction on music signal, a music excerpt can be considered as a sequence of note (pitches) events lasting certain time (durations). Beside beat events, singing voice and vibrating or sweeping instruments, the signal between two note events can be assumed to be quasi-stationary. The duration of a note varies according to the main tempo of the play, type of music and type of melodic component the note is representing. Fast or short notes usually found in melodic lines in high frequency area while slow or long notes are usually found in bass lines with rare exceptions. Let's consider the following example in order to see the difference between the Fourier transform and wavelet one. We construct a test signal as containing two notes E1 and A1 playing simultaneously during the whole period of time (1 second). These two notes can represent a bass line, which, as it is well known, does not change quickly in time. At the same time, we add 4 successive notes B5 with small intervals between them (around 1/16 sec). These notes can theoretically be notes of the main melody line. Let's see now the Fourier spectrogram of the test signal with a small analyzing window.

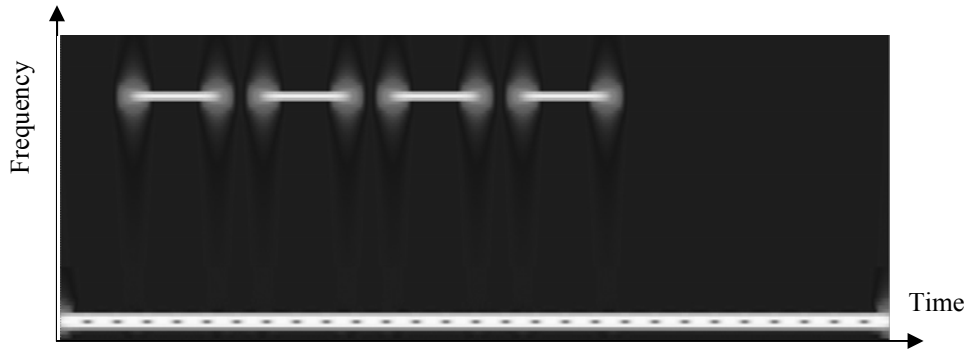

Figure 2. Small-windowed Fourier transform (512 samples) of the test signal containing notes E1 and A1 at the bottom and 4 repeating B5 notes at the top.

As we can see from Figure 2, while high-octave notes can be resolved in time, two bass notes are irresolvable in frequency domain. Now we increase the size of the window in the Fourier transform. Figure 3 illustrates the resulting spectrogram.

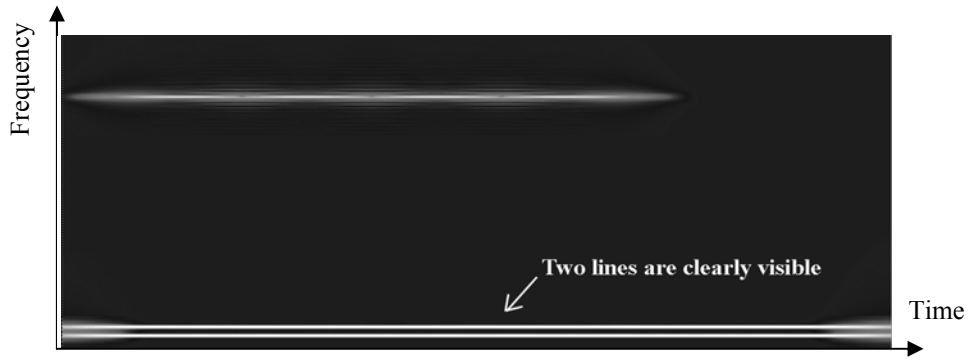

Figure 3. Large-windowed Fourier transform (≥1024 samples) of the test signal containing notes E1 and A1 at the bottom and 4 repeating B5 notes at the top.

As we can see, two lines at the bottom of the spectrogram are now clearly distinguishable while the time resolution of high-octave notes has been lost.

Finally we apply wavelet transform to the test signal. Figure 4 shows such Morlet-based wavelet spectrogram of our test signal.

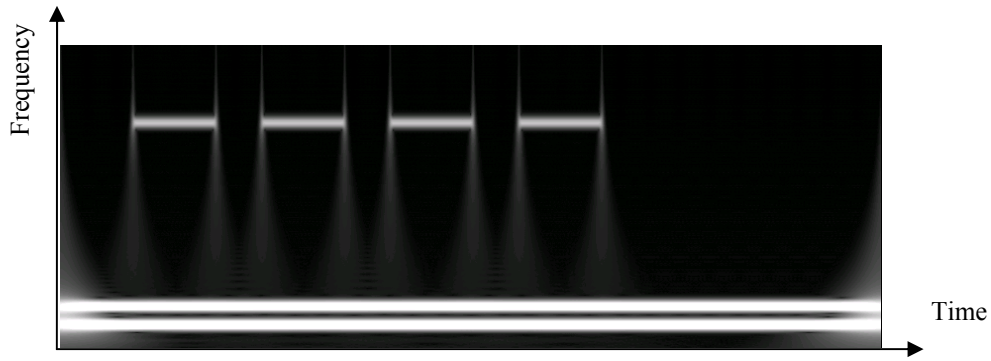

Figure 4. Wavelet transform (Morlet) of the test signal containing notes E1 and A1 at the bottom and 4 repeating B5 notes at the top.

Of course, the given example is quite artificial; however it explains well our motivation for a wavelet like time-frequency representation of a signal. It is also known, that human ear exhibits time-frequency characteristic closer to that from wavelet transform [20].

B. Other transforms and filter banks

The idea to adapt the time/frequency scale of a Fourier-related transform to musical applications is not completely novel. A technique called **Constant Q Transform** [21] is related to the Fourier transform and it is used to transform a data series to the frequency domain. Similar to the Fourier transform a constant Q transform is a bank of filters, but contrary to the Fourier transform it has geometrically spaced center frequencies $f_k = f_0 \cdot 2^{\frac{k}{b}}$ (k = 0;…), where b is the number of filters per octave. In addition it has a constant frequency resolutions ratio $R_{f/\Delta} = \left(2^{\frac{1}{b}} - 1\right)^{-1}$. Choosing appropriately *k* and f_0 makes central frequencies to correspond to the frequencies of notes.

In general, the transform is well suited to musical data (see e.g. [22], in [23] it was successfully used for recognizing instruments), and this can be seen in some of its advantages compared to the Fast Fourier Transform. As the output of the transform is effectively amplitude/phase against log frequency, fewer spectral bins are required to cover a given range effectively, and this proves useful when frequencies span several octaves. The downside of this is a reduction in frequency resolution with higher frequency bins.

Besides constant Q transform there are bounded version of it (BQT) which use quasi-linear frequency sampling when frequency sampling remains linear within separate octaves. This kind of modification allows construction of medium complexity computation schemes in comparison to standard CQT. However, making the frequency sampling quasi-linear (within separate octaves) renders the finding of harmonic structure much more complex task.

Fast Filter Banks are designed to deliver higher frequency selectivity maintaining low computational complexity. This kind of filter banks inherits all disadvantages of FFT in music analysis applications.

More advanced techniques, described for example in [24] are medium-complexity methods which aim to overcome disadvantages of FFT and try to follow note system frequency sampling. However, octave-linear frequency sampling keeps the same disadvantage as in the case of bounded Q transforms.

III. VARIABLE RESOLUTION TRANSFORM

Our Variable Resolutions Transform (VRT) is first derived from the classic definition of Continuous Wavelet Transform (CWT) in order to enable a variable time-frequency coverage which should fit to music signal analysis better. The consideration of specific properties of music signal finally leads us to change the mother function as

well and thus our VRT is not a true CWT but a filter bank.

We start the construction of our VR Transform from Continuous Wavelet Transform defined by (3). Thus, we define our mother function as follows

$$\psi(t) = H(t,l)e^{j\cdot 2\pi\cdot t} \tag{4}$$

where *H(t,l)* is the Hann window function of a length *l* with $l \in Z$ as defined by (5). In our case *l* will lie in a range between 30-300 ms. Notice that using different length values *l* amounts to change the mother wavelet function Ψ.

$$H(t,l) = \frac{1}{2} + \frac{1}{2}\cos\frac{2\pi t}{l} \tag{5}$$

Once the length *l* is fixed, function (4) becomes much more similar to a Morlet wavelet. It is an oscillating function, a flat wave modulated by a Hann window. The parameter *l* defines the number of periods to be present in the wave. Figure 5 illustrates such a function with *l=20 waves*.

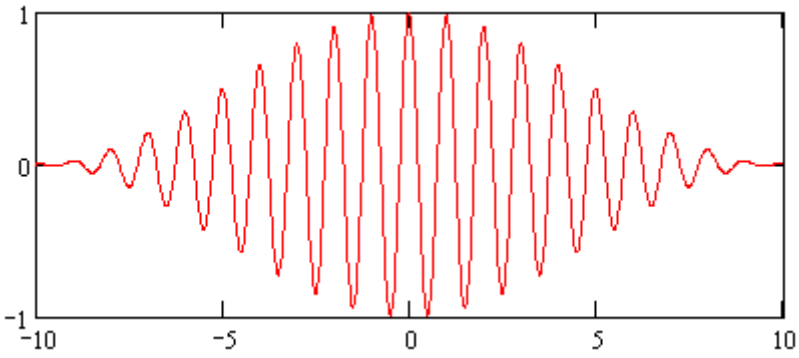

Figure 5. Our mother wavelet function. A flat wave modulated by a Hann window with *l=20*.

We can write according to the definition of the function (since $l < \infty$):

$$\int_{-\infty}^{\infty} |\psi(t)| dt < \infty \text{ and } \int_{-\infty}^{\infty} |\psi(t)|^2 dt < \infty \tag{6}$$

The function is oscillating symmetrically around its 0 value, hence

$$\int_{-\infty}^{\infty} \psi(t) dt \to 0 \tag{7}$$

Using (3) we write a discrete version of the transform for a sampled signal between the instants of time form *t–l/2* to *t+l/2*. Applying the wavelet transform to the signal, we are interested in spectrum magnitude

$$W(a,b)=\frac{1}{\sqrt{a}}\sqrt{\left(\sum_{t=-l/2}^{l/2} s[t+b]\cdot H\left[\frac{t}{a},l\right]\cdot\cos\left(2\pi\frac{t}{a}\right)\right)^2+\left(\sum_{t=-l/2}^{l/2} s[t+b]\cdot H\left[\frac{t}{a},l\right]\cdot\sin\left(2\pi\frac{t}{a}\right)\right)^2} \quad (8)$$

Here *W(a,b)* is the magnitude of the spectral component for the signal *s[t]* at time instant *b* and wavelet scale *a*.

The value of *W(a,b)* can be obtained for any *a* and *b* provided that *b* does not exceed the length of the signal. The equation (8) thus defines a Continuous Wavelet Transform for a discrete signal (time sampling).

The scale of wavelet *a* can be expressed in terms of central frequency corresponding to it since our mother function is a unit oscillation:

$$a=\frac{f_S}{f} \quad (9)$$

where f_S is the sampling frequency of the signal.

A higher value of *a* stands for a lower central frequency.

A. Logarithmic frequency sampling

First of all, the sampling of the scale axis is chosen to be logarithmic in the meaning of frequency. It means that each musical octave or each note will have an equal number of spectral samples. Such a choice is explained by the properties of a music signal, which is known to have frequencies of notes to follow a logarithmic law (following the human perception). Logarithmic frequency sampling also simplifies harmonic structure analysis and economizes the amount of data necessary to cover the musical tuning system effectively.

A voiced signal with single pitch is in the general case represented by its *fundamental frequency* and the fundamental frequency's *partials (harmonics)* with the frequencies equal to the fundamental frequency multiplied by the number of a partial. Hence the distances between partials (harmonic components) and f_0 (basic frequency) in logarithmic frequency scale are constant independently from f_0. Such harmonic structure looks like a "fence", depicted on Figure 6.

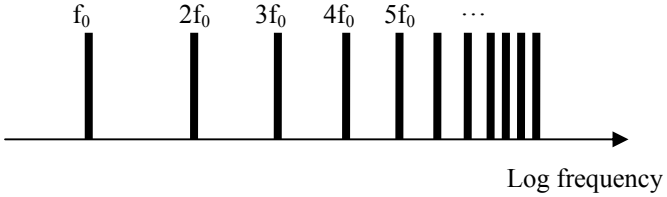

Figure 6. Harmonic structure in logarithmic frequency scale.

In order to cover the frequency axis form f_{min} to f_{max} with N frequency samples with a logarithmic law we define a discrete function $a(n)$, which denotes the scale of wavelet and where n stands for a wavelet bin number ranging in the interval 0..N-1.

$$a(n) = \frac{f_S}{f_{\min} e^{\frac{n}{N} \ln\left(\frac{f_{\max}}{f_{\min}}\right)}} \tag{10}$$

Now the transform (8) sampled in both directions gives

$$W(n,b) = \frac{1}{\sqrt{\frac{f_S}{f_{\min} e^{n \cdot C}}}} \left| \sum_{-l/2}^{l/2} s[t+b] \cdot H\left[\frac{t \cdot f_{\min} \cdot e^{n \cdot C}}{f_S}, l \right] \cdot e^{-i \frac{t f_{\min} \cdot e^{n \cdot C}}{f_S}} \right| \tag{11}$$

where the constant $C = \frac{1}{N} \ln\left(\frac{f_{\max}}{f_{\min}}\right)$.

Expression (11) is the basic expression to obtain an N-bin spectrogram of the signal at time instant b. Thus, for a discrete signal of length S, expression (11) provides $S \times N$ values for each instant of time, N being the number of frequency samples. The expression (11) is still a sampled version of the Continuous Wavelet Transform where the sampling of the scale axis has been chosen logarithmic for N samples.

Frequency dependency on the bin number has the following form (with $f_{min}=50$, $f_{max}=8000$, $N=1000$).

$$f(n) = f_{\min} e^{\frac{n}{N} \ln\left(\frac{f_{\max}}{f_{\min}}\right)} = f_{\min} e^{nC} \tag{12}$$

In order to depict the time/frequency properties of music signals by this discretized wavelet transform with a fixed length value (l=20), let's consider wavelet spectrograms of several test signals. Figure 7 shows the wavelet spectrogram $W(n,b)$ of a piano recording. One can observe single notes on the left and chords on the right. Fundamental frequency (f_0) and its harmonics can be observed in the spectrum of each note. As we can see from the Figure 7, up to 5 harmonics are resolvable. Higher harmonics after the 5th one become indistinguishable especially in the case of chords where the number of simultaneously present frequency components is higher.

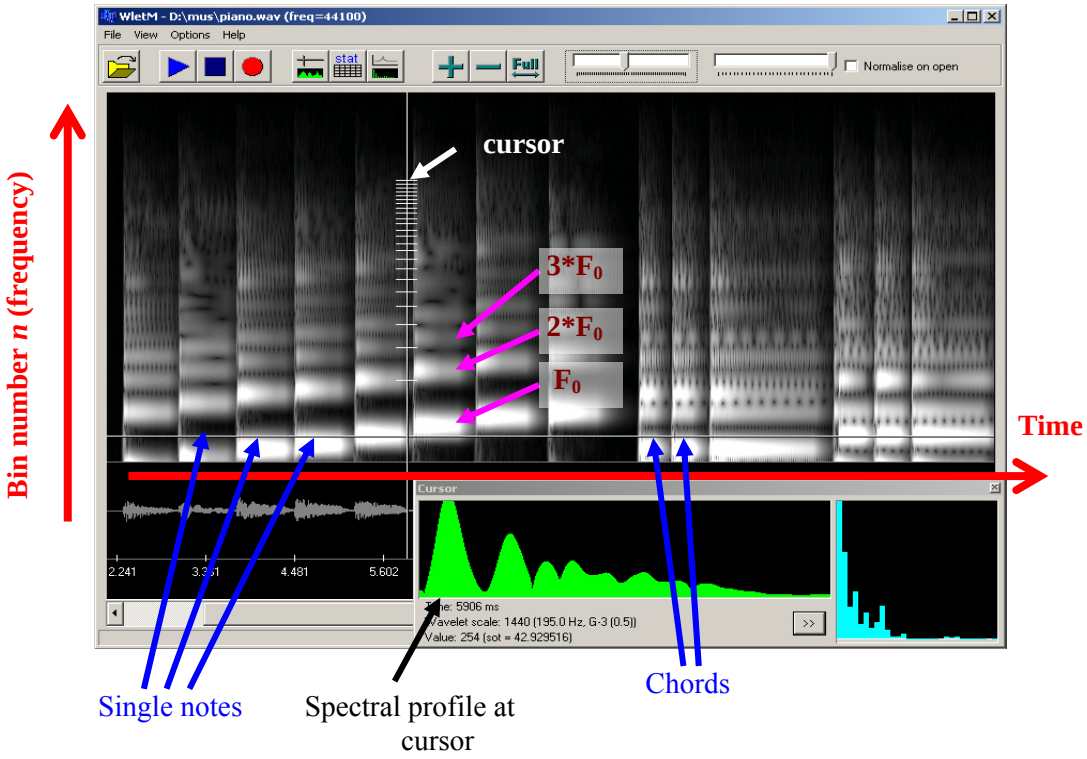

Figure 7. Wavelet spectrogram of a piano recording (wavelet (4)). Single notes on the left and chords on the right. Up to 5 harmonics are resolvable. Higher harmonics after the 5th one become indistinguishable especially in the case of chords where the number of simultaneous frequency components is higher.

Good time resolution is important in such tasks as beat or onset detection for music signal analysis. The next example serves to illustrate the time resolution properties of the Variable Resolution Transform we are developing. In this example we examine a signal with a series of delta-pulses (Dirac) as illustrated in Figure 8 which is a wavelet spectrogram of 5 delta-pulses (1 on the left, 2 in the middle and 2 on the right). As we can see from this figure, Delta-pulses on the picture are still distinguishable even if the distance between them is only 8 ms (right case). In the case of FFT one need 64-sample window size in order to obtain such time resolution.

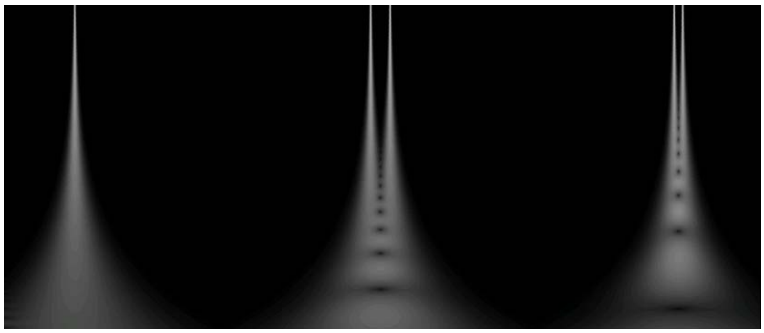

Figure 8. Wavelet transform of a signal containing 5 delta-pulses. The distance between two pulses on the right is only 8 ms.

A quite straightforward listening experiment that we have carried out reveals that the human auditory system is capable to distinguish delta-pulses when a distance between them is around 10 ms. On the other hand, the human auditory system is also able to distinguish very close frequencies - 4Hz in average[1], and down to 0.1Hz

B. *Varying the mother function*

However, music analysis requires good frequency resolution as well. As we can see from the spectrogram in Figure 7, neither high-order partials nor close notes are resolvable, because the spectral localization of the used wavelet is too wide. Increasing the length parameter *l* in (4) or (11) of the Hann window would render our wavelet transform unusable in low-frequency area since the time resolution in low-frequency area would rise exponentially. Thus, we propose in this work to make dynamic parameter *l* **with** a possibility to adjust its behavior across the scale axis**.** For such a purpose we propose to use the following law for parameter *l* in (11) instead of applying scale *a(n)* to parameter *t* in *H(t,l)*:

$$l(n) = L \cdot \left(1 - k_1 \frac{n}{N}\right) \cdot e^{-k_2 \frac{n}{N}} \tag{13}$$

where L is the initial window size, k_1 and k_2 – adjustable parameters

The transform (11) becomes:

$$W(n,b) = \frac{1}{\sqrt{\frac{f_S}{f_{\min} e^{n \cdot C}}}} \left| \sum_{-l/2}^{l/2} s[t+b] \cdot H\left[t, L \cdot \left(1 - k_1 \frac{n}{N}\right) \cdot e^{-k_2 \frac{n}{N}}\right] \cdot e^{-i \frac{t f_{\min} \cdot e^{n \cdot C}}{f_S}} \right| \tag{14}$$

The expression (13) allows the effective"wavelet" width to vary in different ways: from linear to completely exponential to follow the original transform definition. When $L = \frac{f_S}{f_{\min}}$, $k_1=0$ and $k_2=C \cdot N$, (14) is equivalent to (11).

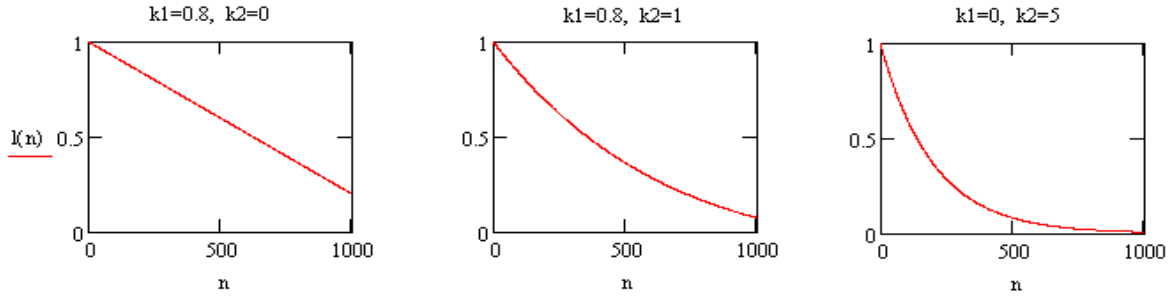

Figure 9. Various l(n), depending on parameters. From linear (left) to exponential (right).

[1] http://tonometric.com/adaptivepitch/

Doing so, we are now able to control the time resolution behavior of our transform. In fact, such transform is not anymore a wavelet transform since the mother-function changes across the scale axis. For this reason we call the resulted transform as *variable resolution transform* (VRT). It can be also referred as a custom filter bank.

As the effective mother-function width (number of wave periods) grows in high-frequency relatively to the original mother-function, the spectral line width becomes more narrow, and hence the transform allows to resolve harmonic components (partials) of the signal. An example of the spectrogram with new variable resolution transform is depicted in Figure 10.

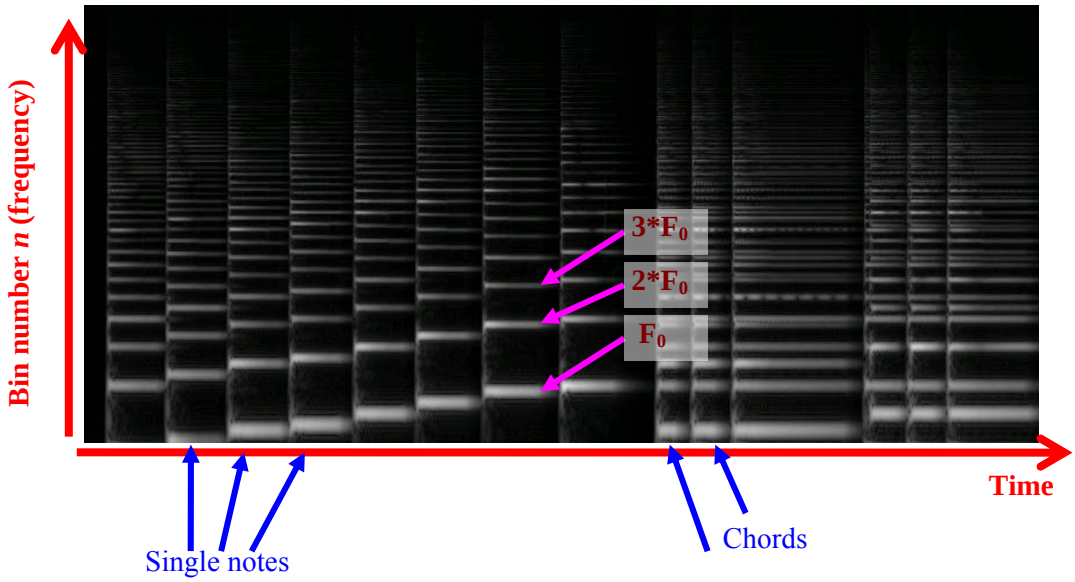

Figure 10. VRT spectrogram of the piano recording used in the previous experiment. Fundamental frequencies and partials are distinguishable (k1=0.8, k2=2.1).

C. *Properties of the VR transform*

A music signal between 50 and 8000 Hz contains approximately 8 octaves. Each octave consists of 12 notes, leading to a total number of notes around 100. A filterbank with 100 filters would be enough to cover such octave range. In reality, frequencies of notes may differ from the theoretical note frequencies of equal-tempered tune because of recording and other conditions. Therefore for music signal analysis considered here, we are working with spectrogram size of 1024 bins – 10 times the amount necessary which covers the note scale by 10 bins per note. Timbre is a one of major properties of music signal along with melody and rhythm. Let's consider now a structure of partials of a harmonic signal (harmonic structure). In Figure 6 we have depicted an approximate view of such structure in logarithmic frequency scale. According to the definition of the function *f(n)* (12), the distance

between partial *i* and partial *j* in terms of number of bins is independent of the absolute fundamental frequency value.

Indeed, according to (12) $n(f)=\frac{1}{C}\ln\frac{f}{f_{\min}}$ and taking into account $f_i=i*f_0$ and $f_j=j*f_0$ we obtain:

$$n(f_j)-n(f_i)=\frac{1}{C}(\ln(f_0\cdot j)-\ln f_{\min})-\frac{1}{C}(\ln(f_0\cdot i)-\ln f_{\min})=\frac{1}{C}(\ln(f_0\cdot j)-\ln(f_0\cdot i))=\frac{1}{C}\ln\frac{j}{i}$$

An accurate harmonic analysis of music signal implies that frequency resolution in terms of spectrogram bin number, expressed by the spectral dispersion, should be always below the distance between neighboring components under consideration.

Having the total width of 20-partial harmonic structure to be a constant around 600 points in terms of number of bins ($n(f_{20})$ - $n(f_0)$), we can establish that the frequency resolution of the obtained transform is large enough to resolve high-order partials we are interested in at all positions of the VRT spectrogram, especially for low octave notes. It means that a 20-partial harmonic structure starting from the beginning of the spectrogram will always lie *above* the dispersion curve. If we consider now the time resolution of the transform, we must recall Figure 9, where various dependencies on the effective width of filter were given. If we define the maximum effective window size to be 180ms (recall our musical signal properties) we obtain the following time resolution grid as illustrated in Figure 11.

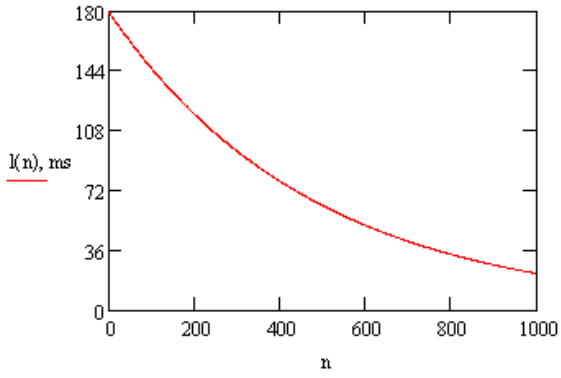

Figure 11. Time resolution dependency of VR transform with $k_2=0.8$, $k_2=2.1$.

D. Discussion

As we can see, our Variable Resolution Transform is derived from the classic definition of Continuous Wavelet Transform [25; 26]. However, our VRT is not a CWT even though they have many similarities. The main difference between VRT and CWT resides in the frequency axis sampling, as well as in the mother wavelet

function which is changing its form across the scale (or frequency) axis in the case of VRT in order to have enough resolution details for high order frequency partials. This last property is not a wavelet transform, because in the true wavelet transform the mother function is only scaled and shifted making a discrete tiling of the time-frequency space in the case of DWT or infinite coverage in the case of CWT. Our VRT can be also referred to as a specially crafted filter bank. Major differences between our VRT and a wavelet transform are:

- no 100% space tiling
- no 100% signal reconstruction (depending on parameters)
- mother function changes

Major similarities between our VRT and a wavelet transform are the following:

- They are based on specially sampled version of CWT
- with certain parameters they can provide 100% signal reconstruction
- low time resolution and high frequency resolution in low frequency area and high time with low frequency resolution in high frequency area

IV. APPLICATIONS: MULTIPLE F0 ESTIMATION

A music signal generally is a composite signal blended of signals from several instruments and/or voices thus having multiple fundamental frequencies. Accurate estimation of these multiple F0s can greatly contribute to further music signal processing and it is an important scientific issue in the field. As the estimation of multiple F0s mostly requires the signal processing in the frequency domain, this problem is a very good illustration highlighting the properties of our VRT.

Early works on automatic pitch detection were developed for speech signal. (see e.g. [27; 28]). Much literature nowadays treats the ***monophonic*** case (only one f_0 present and detected) of fundamental frequency estimation. There are also works studying the polyphonic case of music signal. However, in most of these works the polyphonic music signal is usually considered with a number of restrictions such as the number of notes played simultaneously or some hypothesis about the instruments involved.

The work [29] presents a pitch detection technique using separate time-frequency windows. Both monophonic and two-voice polyphonic cases are studied. Multiple-pitch estimation in the polyphonic single-instrument case is described in [30] where authors propose to apply a comb-filter mapping linear frequency scale of FFT into logarithmic scale of notes frequencies. As the method is FFT-based, the technique inherits drawbacks of FFT for

music signal analysis as we highlighted in Chapter 3, namely requiring large FFT analysis windows thus leading to low time resolution.

An advanced f_0 detection algorithm is presented in [31] which is based on finding frequencies which maximize a f_0 probability density function. The algorithm is claimed to work in the general case and have been tested on CD recordings.

We can also mention many other recent works on multiple fundamental frequency estimation, for instance the ones in [32; 33]. Both these works are probabilistic methods. The first one uses a probabilistic HMM-based approach taking into account some a priori musical knowledge such as tonality. Variable results from 50% to 92% of recognition rates for different instruments in MIDI synthesized sequences are reported. The second algorithm is evaluated on synthetic samples where each file contains only one combination of notes (1 note or 1 chord).

It is not evident how to compare these different multiple f_0 estimation algorithms as assumptions or models on the polyphonic music signal are often not explicitly stated. On the other hand, there is no single evident way of multiple f_0 detection. Some algorithms are strong in noisy environment; some algorithms require a priori training; others are able to detect inharmonic tones etc. The most popular approach to f_0 estimation is harmonic pattern matching in frequency domain. Our multiple-f_0 estimation algorithm makes use of this basic idea. It is illustrated in this paper as an example which relies on our VRT specifically designed for music signal analysis.

A. VRT-based multiple f0 estimation

The basic principle of the f_0 estimation algorithm consists of modeling of our VRT spectrum with harmonic models. Real musical instruments are known to have inharmonic components in their spectrum [34]. It means that the frequency of the n^{th} partial can be not strictly equal to f_0*n. The algorithm we describe does not take such inharmonic components into account, but it tolerates some displacement of partials in a natural way.

A typical "flat" harmonic structure used to model the spectrum is depicted in the Figure 12.

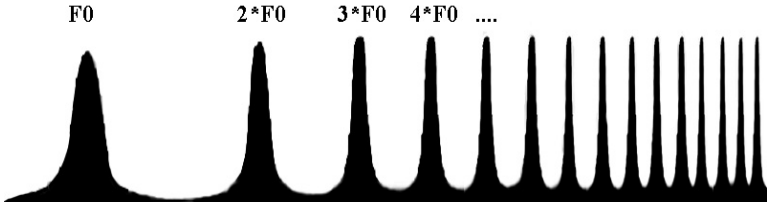

Figure 12. Harmonic structure.

This fence is a vertical cut of VRT spectrogram calculated from a synthetic signal representing an ideal harmonic instrument. The width of peaks and space between them is variable because the VR transform has a

logarithmic frequency scale.

In the next step, these models are used to approximate the spectrum of the signal being analyzed in order to obtain a list of f_0 candidates.

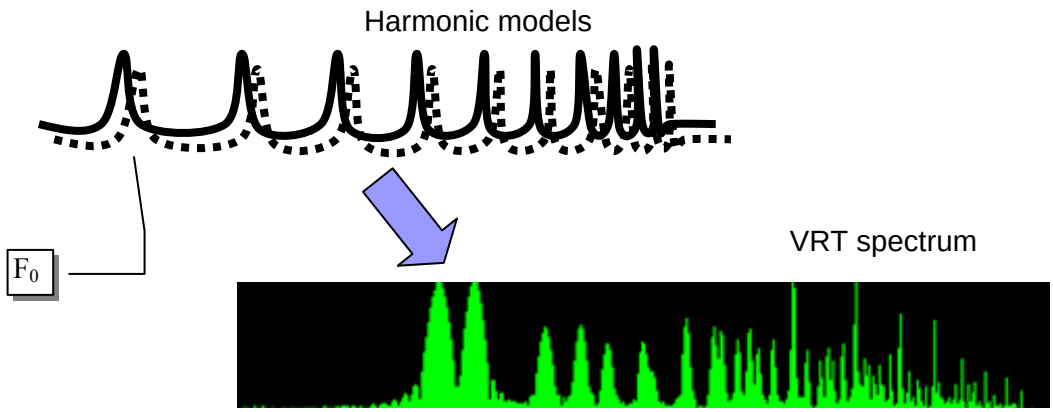

Figure 13. Matching of harmonic models to spectrum.

During every iteration of the algorithm, such harmonic fence is shifted along the frequency axis of the spectrogram and matched with it at each starting point.

The matching of the harmonic model is done as follows. At every harmonic their amplitudes a_i are taken from the values of the spectrogram for the frequencies of i^{th} harmonics. As frequencies of harmonics do not necessarily have integer ratios to the fundamental frequency, we take the maximum amplitude in a close neighborhood, as it is explained in Figure 14.

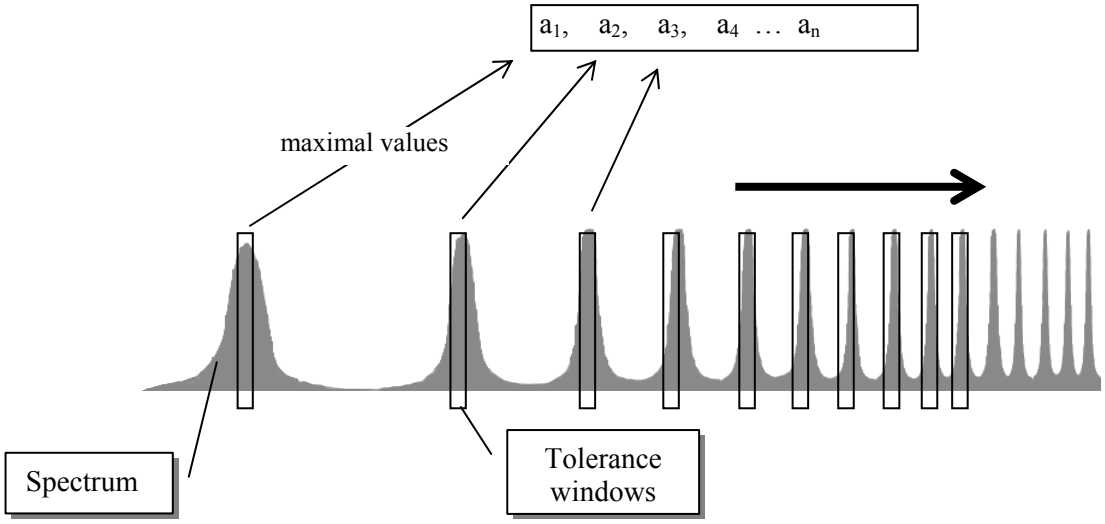

Figure 14. Procedure of extraction of harmonic amplitude vector.

This procedure forms a function $A(f)$ which is a norm of the vector $\boldsymbol{a}$ for the frequency f. The value of frequency for which the function A takes its maximum value is considered as an f_0 candidate.

Further, the obtained f_0 candidate and the corresponding vector $\boldsymbol{a}$ of harmonics amplitudes is transformed into a spectrum slice like in Figure 12. The shape of peaks is taken from the shape of VRT spectrum of a signal with a sine wave with corresponding frequency. This slice is then subtracted from the spectrum under study. The iterative process is repeated either until the current value of harmonic structure *A(f)* becomes inferior compared to a certain threshold or until the maximum number iterations has been reached. We limit the maximum number of iterations to 4, and therefore the maximum number of notes that can be simultaneously detected is 4. As it was observed in preliminary experiments, increasing the number of simultaneously detected notes doesn't improve the f_0 detection performance significantly for high-polyphonic music, because after 3rd or 4th iteration the residue of spectrum is already quite noisy as almost all harmonic components have been already subtracted from it due to harmonic overlaps.

The procedure of note extraction is applied each 25 ms to the input signal sampled at 16 kHz 16 bits. Hence, for the shortest notes with duration around 50-70 ms we obtain note candidates at least twice in order to be able to apply filtering techniques. Every slice produces a certain number of f_0 candidates; then, f_0 candidates are filtered in time in order to remove noise and unreliable notes. The time filtering method used is the nearest neighbor interframe filtering. 3 successive frames are taken and f_0 candidates in the middle frame are changed according to the f_0 candidates in the side neighbors. This filter removes noisy (false detected) f_0 candidates as well as holes in notes issued by misdetection.

B. Experimental evaluation

The easiest way to make basic evaluation experiments in automated music transcription is to use MIDI files (plenty of them can be freely found on the Internet) rendered into waves as input data. The MIDI events themselves serve as the ground truth. However, the real life results must be obtained from recorded music with true instruments and then transcribed by educated music specialists.

In our work we used wave files synthesized from MIDI using hardware wavetable synthesis of Creative SB Audigy2 soundcard with a high quality 140Mb SoundFont bank "Fluid_R3" freely available on the Internet. In such wavetable synthesis banks all instruments are sampled with good sampling rates from real ones: the majority of pitches producible by an instrument are recorded as sampled (wave) block and stored in the soundfont. In the soundfont we used, acoustic grand piano, for example, is sampled every four notes from a real acoustic grand piano. Waves for notes which are in between these reference notes are taken as resampled waves of closest reference notes. Therefore, signal generated using such wavetable synthesis can be considered as a real instrument signal

recorded under ideal conditions. And a polyphonic piece is an ideal linear mixture of true instruments. To make the recording conditions closer to reality in some tests we passed the signal over speakers and record it with a microphone.

Recall and Precision measures are used to measure the performance of the note detection. Recall measure is defined as:

$$\text{Recall} = \frac{\textit{the number correct notes detected}}{\textit{the actual number of notes}} \tag{15}$$

Precision is defined as follows:

$$\text{Precision} = \frac{\textit{the number correct notes detected}}{\textit{the number of all notes deteced}} \tag{16}$$

For the overall measure of the transcription performance, the following *F1* measure is used

$$F1 = \frac{2 \cdot Recall \cdot Precision}{Recall + Precision} \tag{17}$$

All falsely detected notes also include those with octave errors. For some tasks of music indexing as for instance tonality determination, what is important is the note basis and not the octave number. For this reason, the performance of note detection without taking into account octave errors is estimated as well.

Our test dataset consists of 10 MIDI files of classical and pop compositions containing 200 to 3000 notes. Some other test sequences were directly played using the keyboard. The following tables (Table 1 - Table 4) display precision results of our multiple pitch detection. Perf.Oct column stands for performance of note detection not taking into account notes' octaves (just the basic note is important). The *polyphony* column indicates the maximum and the average number of simultaneously sounding notes found in the play.

Table 1**.** Note detection performance in monophonic case. Sequences are played manually using the keyboard.

Name	№ of notes	Polyphony max / avg	Performance Recall	Prec	F1	Perf. Oct F1
Piano Manual	*150*	*1 / 1*	*100*	*100*	*100*	*100*
Violin Manual	*160*	*1 / 1*	*100*	*97*	*98.5*	*100*

Table 2. Note detection performance in polyphonic case. Sequences of chords are played manually using the keyboard.

Name	№ of notes	Polyphony max / avg	Performance Recall	Prec	F1	Perf. Oct F1
Piano Manual	*330*	*2 / 1.8*	*98.5*	*100*	*99.5*	*99.7*
Piano Manual	*214*	*5 / 2.2*	*95.8*	*100*	*97.8*	*99.1*
Flute Manual	*174*	*4 / 2*	*97.7*	*97.7*	*97.7*	*99.7*

Table 3**.** Note detection performance in polyphonic case. Classical music titles (single- and multi-instrument, no percussion).

Name	№ of notes	Polyphony max / avg	Performance Recall	Prec	F1	Perf. Oct F1
Fur_Elize	*924*	*6 / 1.6*	*91.1*	*88.7*	*88.9*	*95.6*
Fur_Elize w/ microphone	*924*	*6 / 1.6*	*88.1*	*86.9*	*87.5*	*95.4*
Tchaikovski 01	*177*	*4 / 3.5*	*84.7*	*95.5*	*89.8*	*95.4*
Tchaikovski 16	*186*	*4 / 2.6*	*86.5*	*100*	*92.8*	*97.2*
Bach 01	*687*	*5 / 1.7*	*91.1*	*88.7*	*89.9*	*98.2*
Bach 03	*549*	*5 / 2.1*	*98.9*	*91.9*	*95.2*	*96.8*
Bach Fugue	*252*	*5 / 2.4*	*83.7*	*76.1*	*79.8*	*93.2*
Vivaldi Mandolin Concerto	*1415*	*6 / 2.9*	*70.1*	*74.8*	*72.4*	*91.5*

Table 4**.** Note detection performance in polyphonic case. Popular and other music (multi-instrument with percussion).

Name	№ of notes	Polyphony max / avg	Performance Recall	Prec	F1	Perf. Oct F1
K. Minogue	*2545*	*10 / 4.7*	*40.6*	*37.1*	*38.8*	*64.3*
Madonna	*2862*	*8 / 3.9*	*43.9*	*56.9*	*49.5*	*66.4*
Soundtrack f/ Godfather	*513*	*9 / 4.1*	*88.7*	*67.2*	*76.5*	*90.4*

As we can see from these tables, our algorithm performs quite well in the monophonic case. Good results are also obtained in polyphonic case with classical music having a low average level of polyphony (number of notes simultaneously played). More complex musical compositions which include percussion instrument and have high polyphony rate have produced lower recognition rates. In our note detection algorithm, we have limited the maximal detectable polyphony to 4 while the maximal and average polyphony in the case of popular and other music is 10 and 4.7 correspondingly. The octave precision, however, stays high (perf. Oct F1 field).

For comparison purpose, we also implemented our note detection algorithm based on FFT with different window size instead of our VRT. We carried out an experiment with a set of polyphonic classical compositions (~1000 notes) using this FFT-based note detection algorithm. Table 5 and Figure 15 summarize the experimental results.

Table 5**.** Comparison of transcription performance based on different time-frequency transforms (the FFT with various window sizes versus VRT).

Transform	FFT	FFT	FFT	VRT
FFT size or number of VRT frequency samples	*1024*	*2048*	*4096*	*1024*
Result (F1)	*66.2*	*77.6*	*80.5*	***91.3***

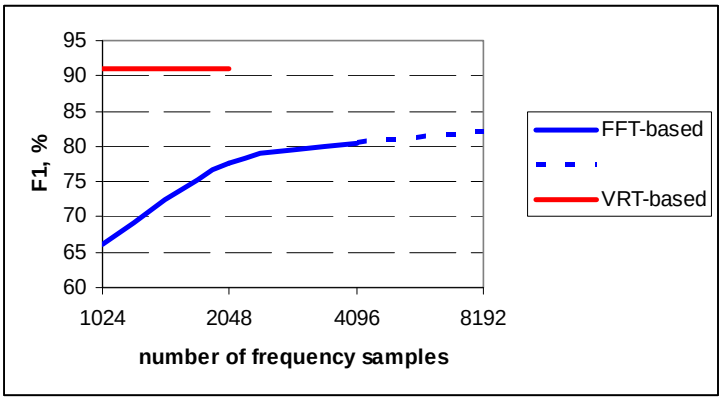

Figure 15. Note detection algorithm performance according to underlying spectral analysis approach.

Further increase of the FFT window size lowers the time resolution down to 0.5-1 seconds so that note changes quicker that 0.5 seconds cannot be resolved anymore.

These experimental results show the advantage of our VRT and its simple use performs multiple note detection quite well in the case of low average polyphony rate.

V. Conclusion

In this paper we have introduced a Variable Resolution Transform as a novel signal processing technique specifically designed for music signal analysis. A music signal is characterized by four major properties: melody, harmony, rhythm and timbre. The classic Fast Fourier transform, a de-facto standard in music signal analysis in the current literature, has its main drawback of having a uniform time-frequency scale which makes it impossible to perform efficient spectrum analysis together with good time resolution. The wavelet transform overcomes this limit by varying the scale of mother-wavelet function and, hence, the effective window size. This kind of transform keeps frequency details in low-frequency area of the spectrum as well as time localization information about quickly changing high-frequency components. However, the dramatic decrease of frequency resolution of the basic wavelet transform in high-frequency area leads to confusion in high order harmonic components where a sufficient resolution is necessary for the analysis of harmonic properties of a music signal. We have thus introduced our Variable Resolution Transform in varying mother-function. The law of variation is controlled by two parameters, linearity and “exponentiality”, which can be carefully chosen in order to adjust the frequency-time resolution grid of the VRT. Hence, our VRT takes advantage of the classic continuous wavelet transform and the windowed or short-time.

As an example of direct VRT application we have presented a VRT-based multiple-f_0 estimation algorithm characterized by its simplicity, rapidity and high temporal resolution as opposed to the FFT-based methods. It performs pretty well in the detection of multiple pitches with non-integer rates. However, as other similar

algorithms, our VRT-based multiple f_0 estimation algorithm does not solve the following problem: two notes with a distance of an octave can hardly be separated, because the second note does not bring any new harmonics into the spectrum, but rather changes the amplitude of existing harmonics of the lower note, so some knowledge of the instruments involved in the play or instrument recognition techniques and multi-channel source separation is necessary to resolve the problem.

Our note detection mechanism was evaluated in its direct application – musical transcription from the signal. In this evaluation ground truth data was taken as note score files – MIDI. These files from various genres (mostly classical) were rendered into waves using high-quality wavetable synthesis. The resulting wave files were passed as input for the transcriptions algorithm. The results of the transcription and the ground-truth data were compared and a performance measure was calculated. Compared to the FFT, the VRT being used in described f_0 estimation algorithms gives much higher results together with excellent time resolution. As a major drawback of the VRT an important complexity could be mentioned. Nevertheless, it does not hamper a real-time audio processing every 25ms.

Actually we also applied the VRT to the extraction of other music features including timber, tempo estimation or music similarity-based retrieval [25; 26]. In all these problems, the VRT has depicted interesting properties for music signal analysis [thesis].

VI. REFERENCES

[1] Tanguiane A.S.. Artificial perception and music recognition (lecture notes in computer science). . Springer, October 1993.

[2] Casagrande N., Eck D., Kegl B.. Frame-level audio feature extraction using adaboost. *Proceedings of the ISMIR International Conference on Music Information Retrieval (London)* (2005) : pp. 345-350.

[3] Logan B. SA. A music similarity function based on signal analysis.. *In Proceedings of IEEE International Conference on Multimedia and Expo ICME 01* (2001)

[4] Mandel M. ED. Song-level features and support vector machines for music classification. *Proceedings of the ISMIR International Conference on Music Information Retrieval (London)* (2005) : pp. 594-599.

[5] McKinney M.F. BJ. Features for audio and music classification. *Proceedings of the ISMIR International Conference on Music* (2003) : pp. 151-158.

[6] Meng A., Shawe-Taylor J.,. An investigation of feature models for music genre classification using the

support vector classifier. *Proceedings of the ISMIR International Conference on Music Information Retrieval (London)* (2005) : pp. 604-609.

[7] Scaringella N. ZG. On the modeling of time information for automatic genre recognition systems in audio signals. *Proceedings of the ISMIR International Conference on Music Information Retrieval (London)* (2005) : pp. 666-671.

[8] Tzanetakis G. CP. Automatic musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing 10* (2002) : p. no 5, 293-302.

[9] West K. CS. Features and classifiers for the automatic classification of musical audio signals. *Proceedings of the ISMIR International Conference on Music Information Retrieval (Barcelona, Spain)* (2004) : pp. 531-536.

[10] Foote Jonathan T.. Content-based retrieval of music and audio. *Proceedings of SPIE Multimedia Storage and Archiving Systems II (Bellingham,WA) vol. 3229, SPIE* (1997) : pp. 135-147.

[11] Logan B.. Mel frequency cepstral coefficients for music modeling. *Proceedings of the ISMIR International Symposium on Music Information Retrieval (Plymouth, MA)* (2000)

[12] Aucouturier J.J. PF. Timbre similarity: how high is the sky?. *In JNRSAS* (2004)

[13] Pampalk E.. Computational models of music similarity and their application in music information retrieval. PhD thesis at Technishen Universitaet Wien, Fakultaet fuer Informatik.2006.

[14] Kronland-Martinet R. MJAGA. Analysis of sound patterns through wavelet transform. *International Journal of Pattern Recognition and Artificial Intelligence,Vol. 1(2)* (1987) : pp. 237-301.

[15] Grimaldi M., Kokaram A., Cunningham P.. Classifying music by genre using the wavelet packet transform and a round-robin ensemble. (2002)

[16] Kadambe S. FBG. Application of the wavelet transform for pitch detection of speech signals. *IEEE Transactions on Information Theory* (1992) 38, no 2: pp. 917-924.

[17] Mallat S.G.. A wavelet tour of signal processing. . Academic Press, 1999.

[18] Grossman A. MJ. Decomposition of hardy into square integrable wavelets of constant shape. *SIAM J. Math. Anal.* (1984) 15: pp. 723-736.

[19] Lang W.C. FK. Time-frequency analysis with the continuous wavelet transform. *Am. J. Phys.* (1998) 66(9): pp. 794-797.

[20] Tzanetakis G., Essl G., Cook P.. Audio analysis using the discrete wavelet transform. . *WSES Int. Conf. Acoustics and Music: Theory 2001 and Applications (AMTA), Skiathos, Greece* (2001)

[21] Brown J. C.. Calculation of a constant q spectral transform. *J. Acoust. Soc. Am.* (1991) 89(1): pp. 425-434.

[22] Nawab S.H., Ayyash S.H., Wotiz R.. Identification of musical chords using constant-q spectra.. *In Proc. ICASSP* (2001)

[23] Essid S.. Classisfication automatique des signaux audio-fréquences : reconnaissance des instruments de musique. PhD thesis Informatique, Telecommunications et Electronique, ENST.2005.

[24] Diniz F.C.C.B, Kothe I, Netto S.L., Biscainho L.P.. High-selectivity filter banks for spectral analysis of music signals. *EURASIP Journal on Advances in Signal Processing* (2007)

[25] Paradzinets A., Harb H., Chen L.,. Use of continuous wavelet-like transform in automated music transcription. *Proceedings of EUSIPCO* (2006)

[26] Paradzinets A., Kotov O., Harb H., Chen L.,. Continuous wavelet-like transform based music similarity features for intelligent music navigtion. *In proceedings of CBMI* (2007)

[27] Abe T. et al.. Robust pitch estimation with harmonics enhancement in noisy environments based on instantaneous frequency. *In proceedings of ICSLP'96* (1996) : pp. 1277-1280.

[28] Hu J., Sheng Xu., Chen J.. A modified pitch detection algorithm. *IEEE COMMUNICATIONS LETTERS* (2001) Vol. 5, No 2

[29] Klapuri A.. Pitch estimation using multiple independent time-frequency windows. *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics* (1999)

[30] Lao W., Tan E.T., Kam A.H.. Computationally inexpensive and effective scheme for automatic transcription of polyphonic music. *Proceedings of ECME* (2004)

[31] Goto M.. A predominant-f0 estimation method for cd recordings: map estimation using em algorithm for adaptive tone models. *In proceedings of ICASSP* (2001)

[32] Li Y. WD. Pitch detection in polyphonic music using instrument tone models. *In proceedings of ICASSP* (2007)

[33] Yeh C., Roebel A., Rodet X.. Multiple fundamental frequency estimation of polyphonic music signals. *in Proc. IEEE, ICASSP* (2005)

[34] Klapuri A.. Signal processing methods for the automatic transcription of music. PhD thesis at Tampere University of Technology.2004.