# The wavelet transform
# Applications in Music Information Retrieval

F. Germain
McGill University, Canada

December 21, 2009

**Abstract**

In this report, we present an overview of existing literature about wavelet-based approaches in music information retrieval (MIR). Indeed, we wants to analyze the possibilities of this novel and popular transform in this particular field. in a first time, we look at general characteristics of the wavelet theory that makes it suitable for such applications compared to state-of-the-art research. Then, we give some examples of applications that were found in the MIR literature.

## Introduction

Compared to the Fourier transform, the theory relative to the wavelet transform is very young and its use is presently expanding across most of the domains where signal processing appears. In many domains, this mathematical tool has been proved to be as efficient or even more efficient than the traditional Fourier analysis. The first wavelet-like transform appears in the thesis of Haar in 1909. It makes a second apparition on the work of Zweig in 1975 in his studies of the auditive perception named as *cochlear transform*. The word *wavelet* (from the French *ondelettes*) is first used by Grossman and Morlet in 1982 in the field of geophysics, when they develop the continuous wavelet transform (CWT). In 1983, Strömberg introduces the discrete wavelet transform (DWT). Then, Daubechies build her family of orthogonal wavelets in 1988, used in many domains (Daubechies 1988; Daubechies 1992). At last, Mallat (Mallat 1989) draws the fast wavelet transform to enable fast computation in digital applications.

The wavelet approach is more and more widely used, and in many domains, it can now be used in cooperation with or instead of traditional approach with very good result improvements. The wavelet properties and the growing conviction of its link to some human perception mechanisms brings new perspectives to research. Its use in the music domain has been relatively slow compared to domains such as image processing or geophysics. The first articles referring to applications in music are "old" (Kronland-Maninet, Morlet, and Grossmann 1987; Kronland-Maninet 1988), but most of the applications presented in this report have been developed in the last decade.

Most of the domains in MIR have been explored with this new approach but, despite good preliminary results, it remains not very popular. In this report, we present an overview of these applications. In the first section, we introduces some basics about the wavelet theory. Next, we presents the general advantages and downsides of this approach. In the third part, we give some

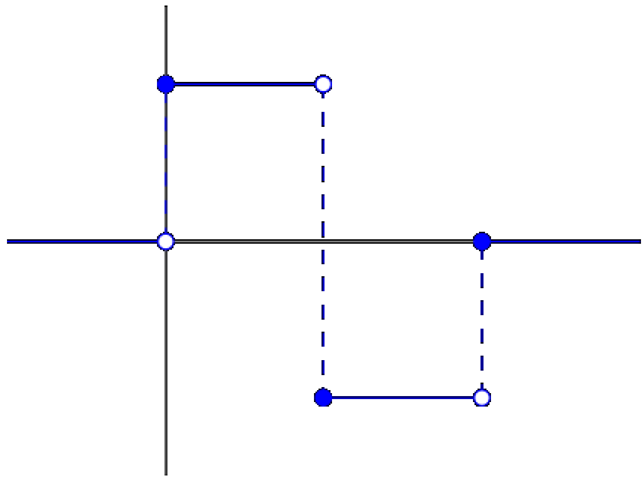

Figure 1: Haar mother wavelet

observations about the different applications reviewed. At last, we provides some examples of promising MIR applications where the wavelet transform was used.

# 1 The wavelet-transform

The wavelet theory is very similar to the Fourier analysis. They use the same approach which is to project linearly the signal on a function base in order to extract meaningful information. However, while the Fourier functions (sines and cosines) are oscillating infinitely, the wavelets are functions with time localization. The two transforms enable perfect signal reconstruction with the continuous and the discrete transform (thanks to the sampling theorem).

## 1.1 Wavelets

The wavelet transform is based on a *mother wavelet* $\psi(t)$. A wavelet is a mathematical function with particular properties such as a finite energy and zero mean. For convenience, the energy is often normalized. Contractions and dilatations of this wavelet are used to tile the time-frequency space. Mallat demonstrated (Mallat 1989) that a wavelet family can also be characterized by a *scaling function* $\phi(t)$, build from the quadratic mirror filter of the mother wavelet. They form a high-pass/low-pass filter couple that is used for the FWT.

**Haar wavelet** They are the first known wavelets. They have some disadvantages such as their non-differentiability but their simplicity have made them popular. The mother wavelet is the step function (Fig. 1), which acts as numerical differentiation, and the scaling function is the constant function on $[0, 1]$, which acts as an averager.

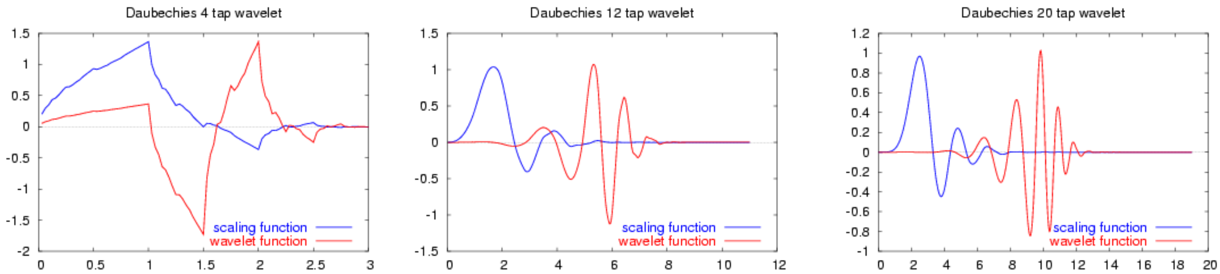


Figure 2: Examples of Daubechies wavelet (mother wavelet and scaling function) - db4, db12 and db20

**Daubechies wavelet** The discrete wavelet family drawn by Daubechies (Daubechies 1988) is widely used in wavelet applications. They have been proven to be very efficient to catch useful information in signals. It is a family of orthogonal wavelets (what means that the contraction of the mother wavelet are an orthogonal base of their subspace) with the highest possible number of vanishing moments, and among the possible solutions, the one whose scaling function has the extremal phase (Fig. 2). They are build recursively from the *Daubechies db2* wavelet, which is equal to the Haar wavelet. They are numbered with even numbers, the db2$n$ wavelet (e.g., db2, db4, db6,..., db20) has $n$ vanishing moments.

**Choice of wavelets** As we will see further in this report, most of the applications use one of the Daubechies wavelet which are widely used for image processing as well. Some applications prefer the Haar one due to its simplicity and its low computation time. Many types of wavelet have been explored in research, and the choice of the wavelet is an important criterion of efficiency in applications. One can cite Gabor-complex (optimally located in time and frequency), the Simlet, the Coifman, the Morlet or the cubic spline wavelets. In some applications, the choice of the wavelets have been proven to have little or no effect (Didiot et al. 2010).

## 1.2 Wavelet transforms

**Continuous wavelet transform** The continuous wavelet transform is similar to the Fourier transform (FT). However, since the wavelets are localized in time and frequency, while the sines and cosines have infinite lengths, the wavelets have to be shifted in time to transform the whole space. Thus, the coefficients are defined on a time/frequency space (while the FT coefficients are only on the frequency axis) by the projection of the signal $y(t)$ on the contracted/dilated shifted wavelets:

$$C(b, a) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} \psi\left(\frac{t-b}{a}\right) y(t)dt \tag{1}$$

The signal can be rebuilt perfectly from these coefficients with a double integration on time and frequency axes. Thus, the coefficients are highly redundant.

**Discrete wavelet transform** The discrete wavelet transform (DWT) is directly derived from the CWT. It consists in using only dyadic wavelets (contraction and dilatation of the mother wavelet

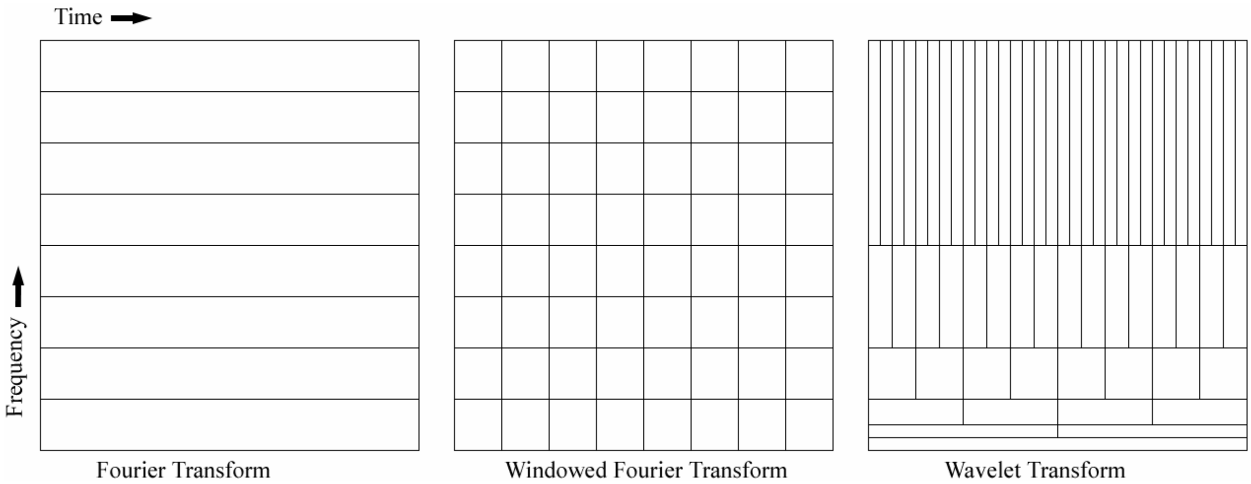


Figure 3: Time/frequency tiling with FFT, STFT and DWT (Hughes 2006)

by powers of 2) with a sampling of the coefficients:

$$C(j,k) = \sum_j \sum_k \psi\left(2^{-j}n - k\right) y(k) \times 2^{-j/2} \tag{2}$$

It removes the redundancy of the CWT coefficients, and have an optimal compactness (if the sampling theorem condition is verified, the reconstruction of the analog signal is possible). In its expression, DWT is very similar to the short-time Fourier transform. The difference is that, rather tiling uniformly the frequency and time axes, the wavelet transform uses dilatation of the function so that we have low time resolution and high frequency resolution at low frequencies and the opposite at low frequencies (Fig. 3). The calculation is based on the projection of the signal on the base composed by the contracted wavelet and the scaling functions. An efficient implementation was presented by Mallat (Mallat 1989) using a pyramidal algorithm and quadratic mirror filters. It uses the filters associated with the mother wavelet and the scaling function and downsampling to get the coefficients (Fig. 4). The wavelet output contains as many coefficients as the input signal. The computation has a linear complexity.

**Wavelet packet transform** A slightly different algorithm is sometimes used in some music applications. To increase the information available in the highest part of the frequency axis (where the frequency resolution is the lowest), this subbands are also processed by a wavelet transform. The result is often sorted in a binary tree. Each level of the tree contains as many coefficients as the original signal.

**Alternative wavelet transforms** Noticing some recurrent weaknesses of the basic wavelet transforms in music applications, Evangelista has worked during many years on alternative wavelet transforms in order to improve some particular properties. The examples of such transforms are the comb-filter wavelet transform, designed for onset detection, the multiplexed wavelet transform, designed for pitch tracking (Evangelista 1994), the pitch-synchronous wavelet transform, designed for audio compression (Evangelista 1993), and frequency-warped wavelet transform, designed to

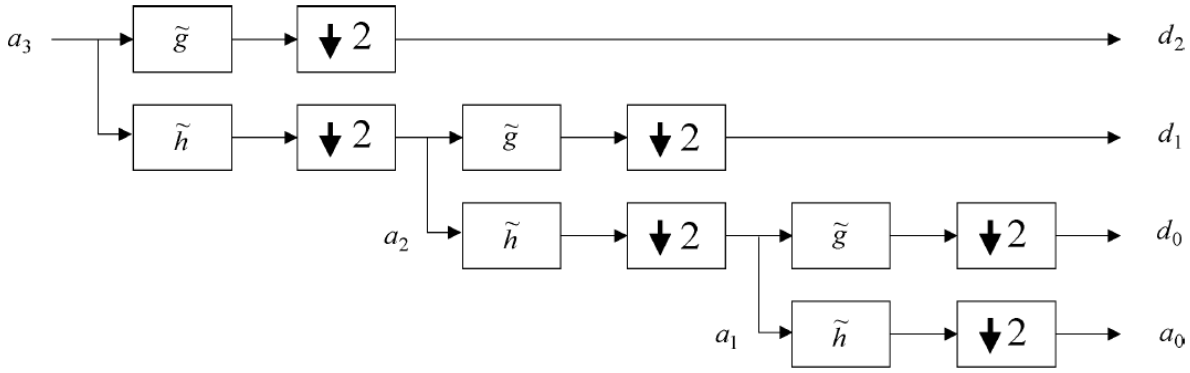


Figure 4: Three-level fast wavelet transform - $\tilde{g}$ is the scaling function filter and $\tilde{h}$ is the mother wavelet filter (Lin et al. 2005)

enable user control on the time/frequency space tiling (Evangelista and Cavaliere 1998a; Evangelista and Cavaliere 1998b; Evangelista 2001). These transforms can be considered with the idea that Lambrou has demonstrated that a classification efficiency can be improved by using adaptative frequency subband definition (Lambrou et al. 1998). However, due to the novelty of the wavelet transform in the music domain, basic transforms are preferred in current research to assess the potential of this approach.

## 2 Interests of the wavelet transform

The wavelet appears to have some particular properties that make it useful to be used in cooperation with or instead of usual techniques such as the Fourier analysis for many applications in many domains such as MIR.

### 2.1 Advantages

**Compactness** With as many coefficients as the initial signal, the wavelet transform conveys a wide range of information due to the multi-scale analysis. It is so interesting for applications where the number of stored information is a concern.

**Robustness** On the contrary to the Fourier analysis, the wavelet transform is more robust to non-stationarity or distortions present in the signal. This difference is due to the fact that the Fourier analysis is based on the hypothesis of quasi-stationary signals (locally in the case of STFT). The wavelet-transform also appears to be very robust to noise and bitrate difference (Kobayakawa, Hoshi, and Onishi 2005). The robustness to noise is particularly important due to the vanishing moment property of the wavelet that focuses on the important information and discard noisy signal.

**Logarithmic frequency scale** The interest of having a logarithmic scale are obvious in music applications. First, the music notes are also define on a logarithmic scale (Azizi, Faez, and Delui 2009). That means that wavelets can isolate the different octaves in independent subbands. Furthermore, the human perception mechanism is also based on logarithmic scales. For an example, the subbands of the auditive mechanism are roughly organized with this scheme. Other perceptive systems such as vision are considered as performing wavelet-like processing to detect edges on images.

**Computation time** The wavelet transform appears to be faster to compute than the Fourier one. Indeed, for a signal of $n$ samples, the FFT requires $O(n \log n)$ operations while the wavelet transform requires only $O(n)$. This is no longer the case for wavelet packet transform, which has the same complexity than the FFT. In some cases, the implementation of wavelets such as the simple Haar one can enable real-time implementation (Fitch and Shabana 1999).

**Event detection** The multi-scale aspect of the wavelet is an obvious advantages of the wavelet compared to the STFT. Indeed, while the increasing of the time resolution in STFT makes impossible to keep an acceptable frequency resolution, the wavelet can combine the two precisions. However, the resolution of a wavelet could still be too constraining to obtain satisfactory precision on both axes. That's why for some applications (Grimaldi, Cunningham, and Kokaram 2002; Grimaldi, Cunningham, and Kokaram 2003) require to use two different scales.

**Knowledge-free transform** In numerous applications, the wavelet transform seems to outperform the existing features on knowledge-free applications. That means that for all the application non based on models, what is quite common in MIR where few knowledge about the audio pieces is known, the wavelet transform should appear as efficient.

**Decorrelated coefficients** In the case of orthogonal DWT, the different output coefficients are statistically independent from each other. That's means that there is no overlapping in the information contained in each coefficients. It is also the proof that the temporal correlation of the original signal is reduced in its wavelet representation.

## 2.2 Drawbacks

**Intuitiveness** The wavelet transform, on the contrary to the FFT, lacks of intuitiveness for the user. Furthermore, the wide range of available wavelets can make the research of the optimal one quite difficult. There is no notion equivalent to the spectrum to represent the wavelet results. The derivation of usual features such as the centroid is also less intuitive, and the definition of high-level features equivalent to brightness is more complex.

**Feature dimension** The wavelet transform coefficients are often unsuitable to be used as features due to their number and their low-level meaning. In most of the cases, high-level features were derived from these coefficients, often discriminating the different subbands of the transform, or common dimension reduction techniques were used (Lukasik 2005) to keep only meaningful wavelet coefficients. This last approach has very low intuitiveness due to the non obvious meanings of the different wavelet-transform coefficients.

**Periodicity attenuation** As we said before, the wavelet transform doesn't assess the pseudo-periodicity of the signal. It is closer to the hypothesis of signal composed by pieces of polynomial functions, with a degree lower than the last non-vanishing moments of the mother wavelet. That is the reason why some information relative to the periodicity of the signal can be lost in the transform process.

**Frequency resolution** Despite the multi-resolution property of the wavelet transform, the frequency resolution of the DWT, based on the octave, is often too coarse to enable proper audio analysis of sounds. That is one of the reason of the development by Evangelista of the frequency-warped wavelet transform (Evangelista and Cavaliere 1998a).

# 3 Applications of the wavelet transform in MIR

## 3.1 Image recognition extension

The wavelet transform has been used for a long time in image processing. Applications were drawn by Mallat in his first papers about this new transform (Mallat 1989). Numerous applications exist in image compression (e.g., JPEG2000), noise removing and edge/object detection. This last application has an immediate application in MIR for optical music recognition (OMR). The removal of specific objects such as stavelines is a crucial point in this domain and the use of the wavelet, in cooperation with or instead of existing methods, could bring interesting results compared to the literature. Unfortunately, this opportunity seems to be mostly unexplored and only few examples can be found (George 2004).

## 3.2 Digital watermarking

The applications of watermarking are numerous (Li and Xue 2003) including annotation, authentication, broadcast monitoring or tamper proofing. But the most critical one is certainly the copyright protection. To build a proper watermarking, it should be *imperceptible* (the watermarked signal has an equivalent quality), *non-complex* (the algorithm is simple), *robust* (the watermark is not much modified by usual processing and attacks on the signal), *deterministic*, and *safe*. The particular properties of the wavelet transform fit well these requirements and that is the reason why research in MIR have been done in that direction.

Some studies have been conducted on music scores watermarking. Actually, since a music score is primarily an image, wavelet-based techniques found in image processing can be used (Kundur and Hatzinakos 1998). But the most interesting approaches are in the field of audio files. The fact that some information about the original file, such as its time length or its quality, are difficult to know, some particular challenges are present in this domain.

**Audio watermarking on wavelet coefficients** Many watermarking methods are based on the modification of the larger wavelet-transform coefficients. These coefficients are chosen because a slight modification on them insure that the audio modification will be added on loud parts rather than on silent parts. In the study of Fu (Fu, Ma, and Song 2005), the coefficients are chosen in intermediate subbands of a wavelet transform with Daubechies db6 wavelets to insure robustness to high-pass and low-pass filtering. The watermark is a defined signal independent from the audio signal. In the study of Kim (Kim, Lee, and Lee 2007), a much more complex implementation is proposed in order to improve robustness to audio lossy compression and time-scaling. A communication channel approach is chosen with error detection and redundancy. On both examples, conclusive experiments prove the robustness of the two algorithms.

**Audio watermarking on wavelet coefficient statistics** Another approach is presented in the work of Li (Li and Xue 2003). To improve robustness of his algorithm to time-scaling and random-cropping attacks, he chooses to embed the watermark in the mean of signal-frame lowest-subband coefficients. Extensive experiments on many processing attack classes prove the robustness of these implementation.

## 3.3 Feature extraction

Many applications in MIR are based on the feature extraction performed on audio files. One can cite classification, similarity, pitch-detection, beat-tracking, indexing problems. In all these categories, wavelet transform has been used and compared to usual techniques such as Fourier analysis or Mel-frequency cepstrum coefficients (MFCCs).

As expected when we explained the lack of intuitiveness of the wavelet transform, very few high-level features derived from the low-level coefficients and they are only for pitch and rhythm measurement. No timbre features equivalent to brightness has been found in the reviewed papers.

### 3.3.1 Feature extraction requirements

A good feature extraction requires to have the following characteristics (Li, Ogihara, and Li 2003). it should be *comprehensive* (representing the music very well), *compact* (requiring little storage space) and *effective* (low extraction computational cost). The wavelet transform appears to bring interesting perspective considering these three conditions. Indeed, the transform is already more compact and effective that Fourier analysis (FFT/STFT) or MFCCs. And we expect it to be more comprehensive considering the improvements obtained in the reviewed papers and in other fields.

### 3.3.2 Feature categories

Here, we use a feature classification derived from the one introduced by Tzatenakis (Tzanetakis 2002; Tzanetakis and Cook 2002). This work is widely considered as fundamental in the introduction of the wavelet as music features. The classification is aimed to classify features in musically meaningful sets in order to improve the development of automatic genre classification of music.

**Statistical features** The statistical features are popular due to their simplicity and their compactness. However, it is often hard to tell if they are really meaningful, since it is always possible to tell which high-level characteristics they could represent. The statistical features are numerous:

- 1st order statistics: Mean (1st moment), variance (2nd moment), standard deviation, skewness (3rd moment), kurtosis, minimum, maximum, median
- 2nd order statistics: Angular second moment, correlation, entropy

The 1st and 2nd order statistics have been extended in many papers to the wavelet transform coefficients, and they are the main wavelet-based features present in literature. Most of the time, they are calculated for each subband of the transform (which represents one octave in the case of DWT). The basic features such as mean or variance are particularly popular due to their simplicity in the first studies based on the wavelet transform (Khan, Al-Khatib, and Moinuddin 2004; Ntalampiras and Fakotakis 2008; Kapur, Bening, and Tzanetakis 2004; Li and Khokhar 2000; Lambrou et al. 1998). In some similarity applications (Rein and Reisslein 2004), these features appeared to be too broad to allow satisfying efficiency.

In his work, Tzatenakis presents some wavelet-based statistical features (Tzanetakis 2002; Tzanetakis, Essl, and Cook 2001). He tries to characterize them as high-level features related to the texture of the piece:

- The mean of the absolute value of the coefficients in each subband that characterizes the frequency distribution
- The standard deviation in each subband that characterizes the amount of change of the frequency distribution
- The ratios of the mean values between adjacent subbands that increase the information about frequency distribution

**Timbral features** The typical timbral features used (Li, Ogihara, and Li 2003) are statistics of the STFT coefficients or MFCCs. The last ones are particularly popular due to their demonstrated efficiency in speech recognition. The usual features are spectral centroid, spectral rolloff, spectral flux, energy, zero crossings, linear prediction coefficients

Classical features such as subband energy have also been explored with the wavelet transform (Lin et al. 2005). This approach is particularly developed in the study of Didiot (Didiot et al. 2010). In this paper, the author uses the traditional subband energy definition, but develops also new features called the teager energy and the hierarchical energy, which have already been used in speech recognition. These derived energy appears in the experiments as being better than the basic energy and the MFCC features for speech/music discrimination. In particular, the derived energies appeared to be more noise-robust and more comprehensive. Furthermore, the wavelet appears to outperform significantly MFCCs in the case of music/non-music discrimination, and in the case of detection of speech super-imposed on music. This last observation demonstrates the ability of wavelet-based features to deal with non-periodic signals.

Two examples of specific wavelet-based timbral features have been found in the reviewed literature. The first one is the *Daubechies wavelet coefficient histograms* (DWCHs), which were developed

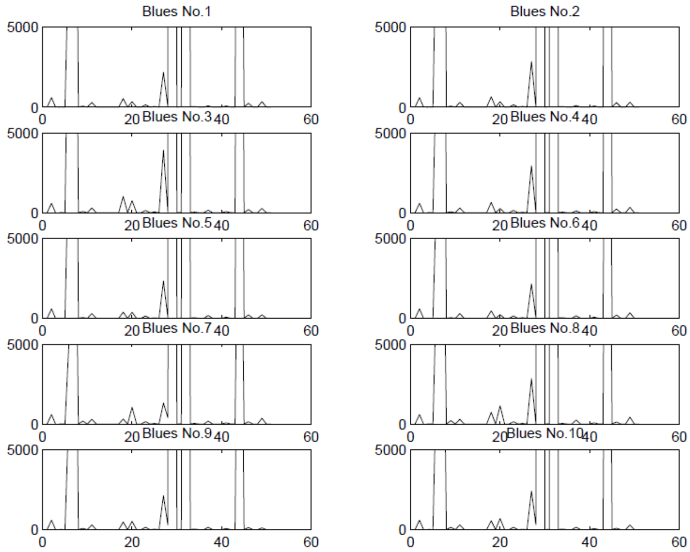


Figure 5: DWCHs of 10 blues songs. The feature representations are similar. (Li, Ogihara, and Li 2003)

by Li (Li, Ogihara, and Li 2003) to improve music classification. With this new features, they explored genre and emotion classification. The feature consists in building the histograms associated with the wavelet coefficients of each subband of the wavelet transform, and then taking the first three moments of these distributions as features. The intuition was that the underlying distribution of these histograms (Fig. 5) could be meaningful genre characteristics.

The second specific feature is the wavelet dispersion vector developed by Rein (Rein and Reisslein 2006) with the CWT coefficients. After having notice the weaknesses of traditional features such as wavelet coefficient envelope or wavelet coefficient statistics, he experimented a new feature measuring the dispersion of high coefficients on the CWT coefficient 2D-plane in his implementation for classical music composition identification, where the aim is to identify the piece played in an unknown performance among a set of pieces (which may not contain this particular performance).

**Rhythmic features** The particular ability of wavelet to detect onset events makes it suitable for beat-tracking applications. In his feature list, Tzanetakis (Tzanetakis and Cook 2002) presents an algorithm for building beat histograms from the DWT coefficients. Their construction is explained further in Sec. 4.2. These features have been used in the query-by-beating study of Kapur (Kapur, Bening, and Tzanetakis 2004). Actually, even if the histograms seem to quite properly detect the tempo of the audio signal, this feature has been demonstrated as inefficient for genre classification compared to traditional timbre features such as MFCCs.

Further studies on beat and onset detection can be found in other papers of our bibliography (Daudet 2001; Bello et al. 2005).

**Pitch features** Some very efficient pitch detection methods have been developed using the wavelet transform. We have examples in our bibliography. The first one is the Fitch's algorithm (Fitch and Shabana 1999), which presents a very simple method. He experiments several different wavelets on his algorithm and tests its robustness to noise. The algorithm looks for simultaneous maxima in several subbands and then calculate the distance between two consecutive peaks as being the signal period. This technique allows the detection of the fundamental frequency and of the more energetic frequency. The results are robust to noise, and the algorithm detect pitch with frames significantly lower than the autocorrelation function. The experiments on different instruments are promising and the algorithm is even capable of identifying non-harmonic instruments. The algorithm seems also not to be sensitive to the size of the used frame. At last, the simplicity of the Haar wavelet transform makes a real-time implementation possible.

The second technique is the Specmurt algorithm developed by Sagayama (Sagayama et al. 2005). It is derived from the cepstrum technique which evaluates the pitch with the inverse Fourier transform of the logarithmic magnitude spectrum of the signal. This new technique is based on the inverse Fourier transform of the wavelet-transform of the signal and seems to be capable of detecting pitches on monophonic and also on multi-pitched signals.

In his paper, Ghias (Ghias et al. 1995) underlines that he planes to include a wavelet-based pitch tracking to his query-by-humming system to improve the results and, what's most important in this case, the computational time of the algorithm.

Other recent wavelet-based pitch-tracking techniques can also be found in our bibliography (Paradzinets et al. 2007; Kondo and Tanaka 2008).

**Feature encoding** A very interesting approach is presented in the work of Woojay (Woojay, Changxue, and Yan 2009). Rather than using wavelet-based features, he chooses to use wavelet-based encoding of his features (fundamental frequency envelopes). This choice is due to the fast retrieval time in the wavelet representation, since the wavelet coefficients are easy to index on a binary tree, and the the distance between two features are easy to compute. The experiments are promising but the results stays comparable to state-of-the-art algorithms.

### 3.3.3 Auditory model

The similarity between human perception models and the wavelet transform is used in some applications. It is used in the multi-timbre chord classification developed by Su (Su and Jeng 2001) where the recognition is performed with a wavelet processing, to simulate the cochlea processing, and a neural network, to simulate the brain functioning. The used wavelet is the Gabor-complex wavelet, which offer the optimal time and frequency localization. These wavelets are considered as being the closest one to the human vision so their extension to audition seems interesting. The authors present few experiments, but the preliminaries results are promising. In particular, a strong robustness to noise is demonstrated, with a successful recognition at a SNR of 0dB.

## 3.4 Source separation

Two papers use a similar approach for source separation (Moussaoui, Rouat, and Lefebvre 2006; Lampropoulou, Lampropoulos, and Tsihrintzis 2008). Both of them use the wavelet packet transform in order to pre-process the input signal. The aim is to minimize the mutual information present in the signals by projecting them on a proper base. In (Moussaoui, Rouat, and Lefebvre 2006), the lowest-entropy coefficients are chosen. The reconstructed signal is then used to build the inverse mixing filter which separates the different sources. Experiments on artificial and real signals have shown promising results in both articles.

# 4 Application examples

## 4.1 Optical music recognition (George 2004)

As said before, the wavelet transform has been used for a long time in image processing. Its efficiency has been proven in knowledge-free applications. Thus, testing this approach for optical music recognition (OMR) was an obvious way to apply this perspective. George made preliminaries experiments of the wavelet transform as a tool to remove super-imposed objects on music scores, or at least to pre-process the image to make that task easier. The two kind of objects studied were on the one hand the stavelines, and on the other hand the other kind of extra symbols. This approach is very interesting since it brings a lot of questions such as:

- if it is possible to segregate specific objects with a general purpose filtering technique.
- if it is possible to find wavelet adapted to each family of symbols.
- if it is possible to evaluate objectively the efficiency of the algorithms.
- if the choice of the parameters of scanning (e.g., image definition) influences the results.

In her experiments, the wavelet tested are chosen among the Daubechies and the Coifman wavelets. Three tests have been done: Staveline location, image segmentation between the symbols and the stavelines with different scanning parameters, and visualization of the transform with different wavelets.

The conclusions of the study are that:

- the pre-processing with the wavelet transform is a good way to make stavelines easier to detect.
- filtering images emphasizes non-stavelines objects on the score
- the best wavelet depends on the kind of aimed symbols

This study demonstrates clearly the potential of such approach but further work is needed to bring proper algorithms in the OMR domain.

## 4.2 Wavelet-based beat histograms (Tzanetakis and Cook 2002)

The wavelet-based beat histogram is a feature presented by Tzatenakis in 2002. The purpose is to use the property of multi-resolution of the wavelet transform to improve the results compared

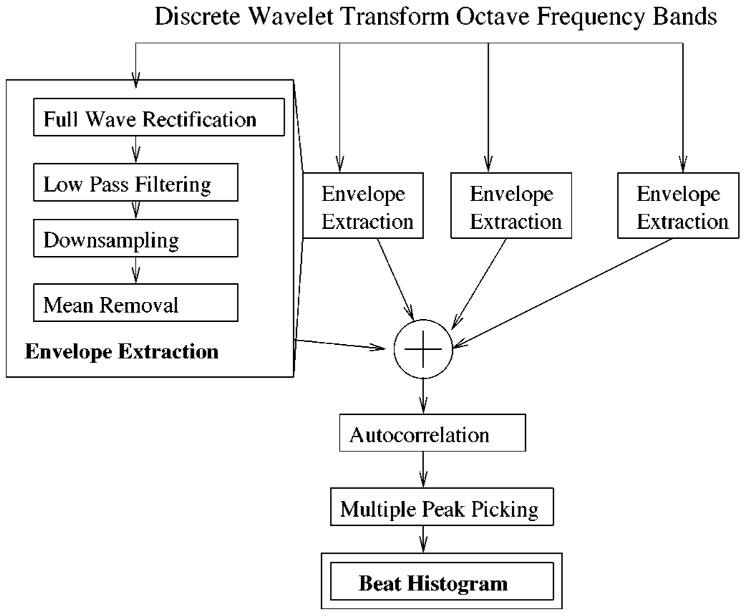


Figure 6: Beat histogram calculation flow diagram (Tzanetakis and Cook 2002)

to algorithms based on less fine frequency-resolution representations such as STFT. The wavelet is intended to be able to catch the sharp variation of the signal due to the onset of notes, and to use the fact that in western music, a majority of the notes are played on the tempo beats.

In his implementation, the beat histogram extraction is performed in four steps (Fig. 4.2):

1. FWT using the Daubechies db4 wavelet
2. Envelope extraction on each subband
3. Autocorrelation on the sum of the envelopes: $y[k] = \frac{1}{N}\sum_n x[n]x[n-k]$
4. Multiple peak picking and histogram calculation: the first three peaks of the autocorrelation in the appropriate range (40-200 beats-per-minute) are selected and added to the histogram.

The envelope extraction from a subband $x[n]$ is a common technique based on full wave rectification:

1. Full wave rectification: $y[n] = |x[n]|$
2. Low pass filtering with a one-pole filter: $y[n] = (1-\alpha)x[n] + \alpha y[n]$ with $\alpha = 0.99$
3. Downsampling: $y[n] = x[kn]$ with $k = 16$
4. Mean removal: $y[n] = x[n] - E\left[x[n]\right]$

The experiments showed that on most of the case (18/20), the piece tempo correspond to one of the two highest histogram peaks. Examples of beat histograms extracted on 30 ms samples from

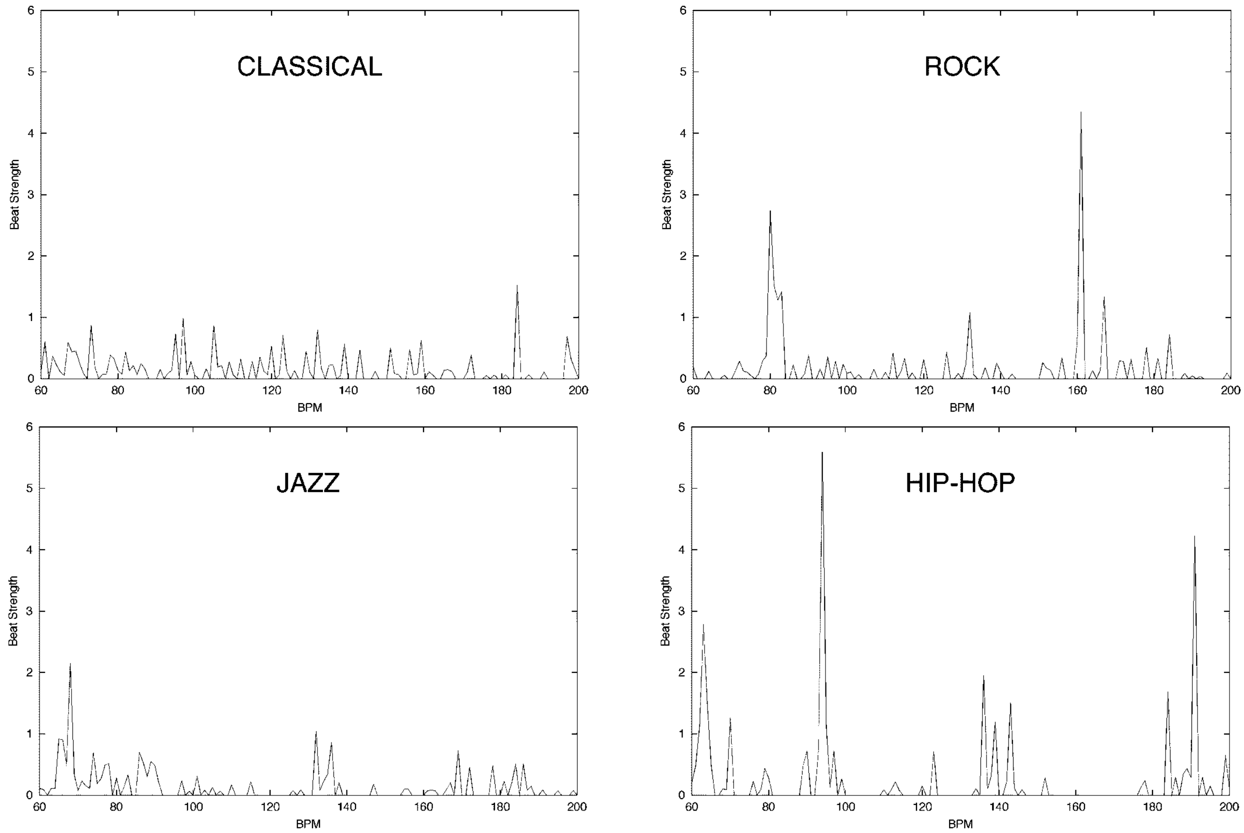


Figure 7: Beat histogram examples (Tzanetakis and Cook 2002)

pieces of four different genres are provided in the paper (Fig. 7) and show some basic properties such as the fact that regular beats are more present on certain types of music (e.g., hip-hop) than other ones (e.g., classical). However, the evaluation presented in the paper shows that this feature is not a very efficient one for genre classification. This conclusion doesn't imply that the beat-tracking is not enough accurate, but that the tempo features are meaningless in genre discrimination. The beat histogram is part of MARSYAS, the software framework for evaluation of computer audition applications developed by Tzatenakis.

## 4.3 Daubechies wavelet coefficient histograms for genre and emotion classification (Li, Ogihara, and Li 2003)

In his paper, Li notices that the traditional feature extraction, such as the one developed by Tzatenakis (Tzanetakis and Cook 2002) capture incomplete information. That's why he develops a very interesting wavelet-based timbral features called Daubechies wavelet coefficient histograms (DWCH). His aim is to obtain characteristic features of the distribution of the amplitude of the signal over the time, and histograms are a good way to identify distribution attributes. However, in order to extract the information of the frequency domain, he chooses to first represent the signal with its DWT. Considering that the information present in each octave is unique, he uses the logarithmic scale property of wavelets to extract one histogram for each subband. Each histogram

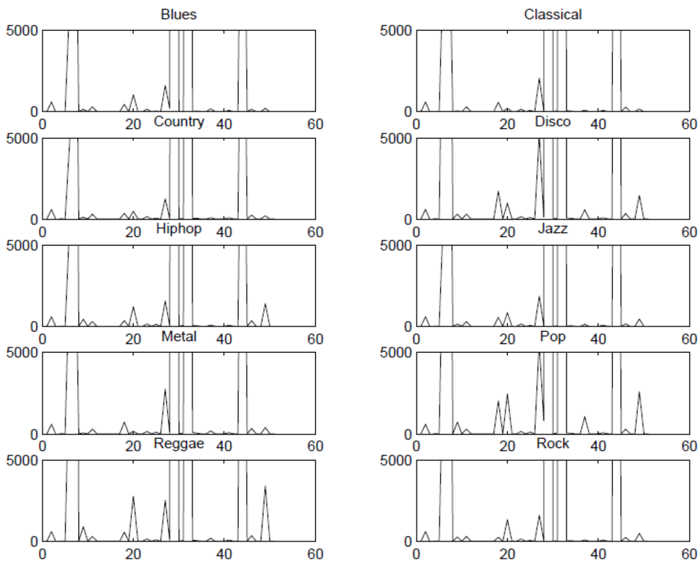


Figure 8: DWCHs of 10 music signals in different genre. (Li, Ogihara, and Li 2003)

is build using the rounded wavelet coefficients. The final features consist in the three first moments of the subband histogram distribution and the subband energy. Examples of histograms for ten music signals in different genre are given (Fig. 8).

This new features are aimed to be used on genre classification, in cooperation with classical features (e.g. MFCCs) from the Tzanetakis' MARSYAS framework (Tzanetakis 2002). Several classifiers (e.g., support vector machines, k-nearest neighbor) are experimented on the same ten-genre classification dataset used by Tzanetakis. In the experiments, Li uses a Daubechies db8 wavelet with 7 levels of decomposition. After the preliminary experiments with classifiers, he decides to keep only the features corresponding to four selected subbands. The final feature vector has a size of 35.

The feature vectors tested on a one-versus-the-rest support vector machine classify accurately 78.5% of the pieces. It is significantly better than the results obtain with the Tzanetakis' feature vector (71.9%) or than the results of the same ten-genre classification made with trained collee students (70%) (Tzanetakis 2002).

# 5 Discrete wavelet tranform with Daubechies wavelets for jAudio

In the frame of this project, a class implementing the calculation of the discrete wavelet tranform with Daubechies wavelets have been implemented in jAudio. The source code is provided in App. A.

## 5.1 General features

**Available wavelets** All the Daubechies wavelets from db2 (Haar) to db20 are stored.

**Algorithm** The implementation uses the pyramidal algorithm of the FWT.

**Remark on computation efficiency** A binary re-ordering of the coefficients of the input signal could enable to perform the algorithm by self-processing the output vector. Anyway, the current implementation can be comparable in computation time, and the use of the input vector as temporary buffer implies that no additional storage space is needed.

### 5.2 Class overview

**Package** The class *DWT_db* is inserted in the *jAudioFeatureExtractor.jAudioTools* package as the FFT transform algorithm.

**Constructor** The constructor performs the wavelet transform and stores the result. Its inputs are the signal to be processed, the index of the desired Daubechies wavelet, and the number of levels of the transform. Input signals are zero-padded if they do not have a length equal to a power of 2. It stores the result in a vector of the same length as the modified input. Errors are thrown if the number of levels and the signal length are incompatible, or if the Daubechies index is not valid.

**Private fields** The class contains two private fields. The first one *output* is the wavelet transform coefficients computed by the constructor. The second one *nb_levels* is the number of levels computed.

**Static fields** Two 2-dimension arrays stores the coefficients of the low-pass and high-pass filters associated respectively with the scaling and the wavelet functions for each Daubechies wavelet. They are respectively called *scaling* and *wavelet*.

**Public methods** Two public methods are provided in the class. The first one *getSubband* returns the octave (or subband) specified in input as a vector. The second one *getSubbandLabels* returns a vector containing the subband boundaries corresponding to the sampling rate specified in input.

## Conclusion

We have seen in this report that many promising MIR research has been carried with wavelet-based approaches. However, one can notice that the popularity of this technique stays quite low compared to classical techniques such as STFT or MFCCs. On the one hand, I think that this is due to the fact that this theory is quite recent (about 20 years old now) and so most applications are preliminary studies, and extensive experiments are missing to evaluate accurately the possibilities of the wavelet transform. Furthermore, we can see in the references that most of the papers have been written in the last decade. On the other hand, the wavelet transform is one of the first technique that is appearing at the same time in speech and music research. Historically, works on speech processing were done earlier since this field is more critical. And many MIR applications are derived from speech processing literature. This is no longer the case for the wavelet transform. However, we saw that in many papers, the approaches are still derived from previous work on speech.

A class implementing the wavelet transform with Daubechies wavelets have been implemented for the jAudio framework. It enables to include in this software wavelet-based features such as the ones presented in this report (e.g., DWCHS, beat histograms) for classification purpose.

The wavelet transform will probably strongly improved the processing capacity in many domains thanks to its very useful properties. One can only hope that further studies will be made in MIR to improve knowledge on this technique and enable a wider use of the wavelets.

## References


Azizi, A., K. Faez, and A. Delui. 2009. *Emerging Intelligent Computing Technology and Applications*, Chapter Automatic music transcription based on wavelet transform, 158–65. Heidelberg: Springer Berlin.

Baluja, S., and M. Covell. 2006. Content fingerprinting using wavelets. *Proceedings of the 3rd European Conference on Visual Media Production*: 198–207.

Baluja, S., and M. Covell. 2007. Audio fingerprinting: Combining computer vision & data stream processing. *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing 2*: 213–16.

Bello, J., L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. Sandler. 2005. A tutorial on onset detection in music signals. *IEEE Transactions on Speech and Audio Processing* 13 (5): 1035–47.

Busch, C., E. Rademer, M. Schmucker, and S. Wolthusen. 2000. Concepts for an watermarking technique for music scores. *Proceedings of the International Conference on Visual Computing*.

Cai, R., L. Lu, H. Zhang, and L. Cai. 2004. Improve audio representation by using feature structure patterns. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing 4*: 345–8.

Chien, Y., and S. Jeng. 2002. An automatic transcription system with octave detection. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing 2*: 1865–8.

Daubechies, I. 1988. Orthonormal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics 41*: 909–96.

Daubechies, I. 1992. Philadelphia: Society for Industrial and Applied Mathematics.

Daudet, L. 2001. Transients modeling by pruned wavelet trees. *Proceedings of the International Computer Music Conference*: 18–21.

Didiot, E., I. Illina, D. Fohr, and O. Mella. 2010. A wavelet-based parameterization for speech/music discrimination. *Computer Speech and Language* 24 (2): 341–57.

Dinh, P., C. Dorai, and S. Venkatesh. 2002. Video genre categorization using audio wavelet coefficients. *Proceedings of the 5th Asian Conference on Computer Vision*.

Dordevic, V., N. Reljin, and I. Reljin. 2005. Identifying and retrieving of audio sequences by using wavelet descriptors and neural network with users assistance. *Proceedings of the International Conference on Computer as a Tool 1*: 167–70.

Endelt, L., and A. la Cour-Harbo. 2004. Wavelets for sparse representation of music. *Proceedings of the 4th International Conference on Web Delivering of Music*: 10–4.

Evangelista, G. 1993. Pitch synchronous wavelet representations of speech and music signals. *IEEE Transactions on Signal Processing* 41 (12): 3313–30.

Evangelista, G. 1994. Comb and multiplexed wavelet transforms and their applications to signal processing. *IEEE Transactions on Signal Processing* 42 (2): 292–303.

Evangelista, G. 2001. Flexible wavelets for music signal processing. *Journal of New Music Research* 30 (1): 13–22.

Evangelista, G., and S. Cavaliere. 1998a. Discrete frequency warped wavelets: Theory and applications. *IEEE Transactions on Signal Processing* 46 (4): 874–85.

Evangelista, G., and S. Cavaliere. 1998b. Frequency warped filter banks and wavelet transform: A discrete-time approach via laguerre expansions. *IEEE Transactions on Signal Processing* 46 (10): 2638–50.

Evangelista, G., and S. Cavaliere. 2005a. Event synchronous thumbnails: Experiments. *Proceedings of the SMC05 Sound and Music Computing* (4).

Evangelista, G., and S. Cavaliere. 2005b. Event synchronous thumbnails: Statistical properties. *Proceedings of the 5th International Conference Understanding and Creating Music* (4).

Evangelista, G., and S. Cavaliere. 2005c. Event synchronous wavelet transform approach to the extraction of musical thumbnails. *Proceedings of the 8th international Conference on Digital Audio Effects* (4): 232–6.

Fitch, J., and W. Shabana. 1999. A wavelet-based pitch detector for musical signals. *Proceedings of 2nd Workshop on Digital Audio Effects*: 101–4.

Fu, Y., Z. Ma, and G. Song. 2005. A robust audio watermarking algorithm based on wavelet transform. *Journal of Information and Computational Science* 2 (1): 7–11.

George, S. 2004. *Visual perception of music notation: On-line and off line recognition.* Hershey, PA: IRM Press.

Ghias, A., J. Logan, D. Chamberlin, and B. Smith. 1995. Query by humming: Musical information retrieval in an audio database. *Proceedings of the 3rd ACM International Conference on Multimedia*: 231–36.

Grimaldi, M., P. Cunningham, and A. Kokaram. 2002. Discrete wavelet packet transform and ensembles of lazy and eager learners for music genre classification. *Multimedia Systems* 11 (5): 422–37.

Grimaldi, M., P. Cunningham, and A. Kokaram. 2003. A wavelet packet representation of audio signals for music genre classification using different ensemble and feature selection techniques. *Proceedings of the 5th ACM SIGMM International Workshop on Multimedia Information Retrieval*: 102–8.

Grimaldi, M., A. Kokaram, and P. Cunningham. 2002. Classifying music by genre using the wavelet packet transform and a round-robin ensemble. Technical report.

Hughes, J. 2006. *An auditory classifier employing a wavelet neural network implemented in a digital design.* Ph. D. thesis, Rochester Institute of Technology, Rochester, NY.

Kapur, A., M. Bening, and G. Tzanetakis. 2004. Query by beat-boxing: Music retrieval for the dj. *Proceedings of the 5th International Conference on Music Information Retrieval*: 170–7.

Khan, K., W. Al-Khatib, and M. Moinuddin. 2004. Automatic classification of speech and music using neural networks. *Proceedings of the 2nd ACM international workshop on Multimedia databases*: 94–9.

Kim, H., B. Lee, and N. Lee. 2007. Wavelet-based audio watermarking techniques: robustness and fast synchronization.

Klapuri, A., and M. Davy. 2006. New York: Springer.

Kobayakawa, M., M. Hoshi, and K. Onishi. 2005. A method for retrieving music data with different bit rates using mpeg-4 twinvq audio compression. *Proceedings of the 13th ACM International Conference on Multimedia*: 459–62.

Kondo, Y., and T. Tanaka. 2008. Automatic music scoring based on wavelet transform. *Proceedings of the SICE Annual Conference*: 1540–3.

Kosina, K. 2002. *Music genre recognition.* Ph. D. thesis, Technical College of Hagenberg,.

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

Kronland-Maninet, R., I. Morlet, and A. Grossmann. 1987. Analysis of sound patterns through wavelet transforms. *International Journal Pattern Recognition Artificial Intelligence* 1 (2): 97–126.

Kundur, D., and D. Hatzinakos. 1998. Digital watermarking using multiresolution wavelet decomposition. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing 5*: 2969–72.

Kwong, M. 2004. *Detection de transitoires dans un signal audio.* Ph. D. thesis, Universite de Sherbrooke.

Lambrou, T., P. Kudumakis, R. Speller, M. Sandler, and A. Linney. 1998. Classification of audio signals using statistical features on time and wavelet transform domains. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing 6*: 3621–4.

Lampropoulou, P., A. Lampropoulos, and G. Tsihrintzis. 2008. Musical instrument category discrimination using wavelet-based source separation. In *New Directions in Intelligent Interactive Multimedia*, 127–36. Heidelberg: Springer Berlin.

Li, G., and A. Khokhar. 2000. Content-based indexing and retrieval of audio data using wavelets. *Proceedings of the International Conference on Multimedia and Expo 2*: 885–8.

Li, T., Q. Li, S. Zhu, and M. Ogihara. 2002. A survey on wavelet applications in data mining. *SIGKDD Explorations Newsletter* 4 (2): 49–68.

Li, T., and M. Ogihara. 2004. Content-based music similarity search and emotion detection. *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing 5*: 705–8.

Li, T., and M. Ogihara. 2006. Toward intelligent music information retrieval. *IEEE Transactions on Multimedia* 8 (3): 564–74.

Li, T., M. Ogihara, and Q. Li. 2003. A comparative study on content-based music genre classification. *Proceedings of the 26th International ACM SIGIR Conference on Research and Development in Information Retrieval*: 282–9.

Li, W., and X. Xue. 2003. An audio watermarking technique that is robust against random cropping. *Computer Music Journal* 27 (4): 58–68.

Lidy, T., and A. Rauber. 2005. Evaluation of feature extractors and psycho-acoustic transformations for music genre classification. *Proceedings of the 6th International Conference on Music Information Retrieval*: 34–41.

Lin, C., S. Chen, T. Truong, and Y. Chang. 2005. Audio classification and categorization based on wavelets and support vector machine. *IEEE Transactions on Speech and Audio Processing* 13 (5): 644–51.

Lin, R., and L. Chen. 2005. A new approach for audio classification and segmentation using gabor wavelets and fisher linear discriminator. *International Journal of Pattern Recognition and Artificial Intelligence* 19 (6): 807–22.

Lippens, S., J. P. Martens, and T. De Mulder. 2004. A comparison of human and automatic musical genre classification. *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing 4*: 233–6.

Liu, Y., Q. Xiang, Y. Wang, and L. Cai. 2009. Cultural style based music classification of audio signals. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*: 57–60.

Lukasik, E. 2005. Wavelet packets features extraction and selection for discriminating plucked sounds of violins. In *Computer Recognition Systems*, 867–75. Heidelberg: Springer Berlin.

Mallat, S. 1989. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 11 (7): 674–93.

Miyamoto, K., H. Kameoka, H. Takeda, T. Nishimoto, and S. Sagayama. 2007. Probabilistic approach to automatic music transcription from audio signals. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing 2*: 697–700.

Moussaoui, R., J. Rouat, and R. Lefebvre. 2006. Wavelet based independent component analysis for multi-channel source separation. *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*.

Ntalampiras, S., and N. Fakotakis. 2008. Speech/music discrimination based on discrete wavelet transform. *Proceedings of the 5th Hellenic conference on Artificial Intelligence: Theories, Models and Applications*: 205–11.

Paradzinets, A., H. Harb, and L. Chen. 2006. Use of continuous wavelet-like transform in automated music transcription. *Proceedings of the European Signal Processing Conference*.

Paradzinets, A., O. Kotov, H. Harb, and L. Chen. 2007. Continuous wavelet-like transform based music similarity features for intelligent music navigation. *Proceedings of the International Workshop on Content-Based Multimedia Indexing*: 165–72.

Rein, S., and M. Reisslein. 2004. *Identifying the classical music composition of an unknown performance with wavelet dispersion vector and neural nets*. Ph. D. thesis, Arizona State University,.

Rein, S., and M. Reisslein. 2006. Identifying the classical music composition of an unknown performance with wavelet dispersion vector and neural nets. *Information Sciences* 176 (12): 1629–55.

Sagayama, S., H. Kameoka, S. Saito, and T. Nishimoto. 2005. 'specmurt anasylis' of multi-pitch signals. *Proceedings of the IEEE-Eurasip Workshop on Nonlinear Signal and Image Processing*: 42–7.

Su, B., and S. Jeng. 2001. Multi-timbre chord classification using wavelet transform and self-organized map neural networks. *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing 5*: 3377–80.

Subramanya, S., and A. Youssef. 1998. Wavelet-based indexing of audio data in audio/multimedia databases. *Proceedings of MultiMedia Database Management Systems*: 46–53.

Turnbull, T. 2005. *Automatic music annotation.* Ph. D. thesis, UC San Diego.

Tzanetakis, G. 2002. *Manipulation, analysis and retrieval systems for audio signals.* Ph. D. thesis, Princeton University.

Tzanetakis, G., and P. Cook. 2002. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing* 10 (5): 293–302.

Tzanetakis, G., A. Ermolinskyi, and P. Cook. 2002. Pitch histograms in audio and symbolic music information retrieval. *Proceedings of the International Society for Music Information Retrieval Conference*: 31–8.

Tzanetakis, G., G. Essl, and P. Cook. 2001. Audio analysis using the discrete wavelet transform. *Proceedings of the Conference in Acoustics and Music Theory Applications*.

Woehrmann, R., and L. Solbach. 1995. Preprocessing for the automated transcription of polyphonic music: Linking wavelet theory and auditory filtering. *Proceedings of the International Computer Music Conference*: 396–9.

Woojay, J., M. Changxue, and C. Yan. 2009. An efficient signal-matching approach to melody indexing and search using continuous pitch contours and wavelets. *Proceedings of the International Society for Music Information Retrieval Conference*: 681–6.

# A class *DWT_db* - Source code

```
/*

 * @(#)DWT_db.java 1.0 December 21, 2009.

 *

 * Francois Germain

 * McGill University

 */

package jAudioFeatureExtractor.jAudioTools;
```

```
/**
 * This class performs fast wavelet transform (FWT) using the Daubechies wavelets
 * <p>The FWT is performed by this class' constructor. The coefficients are stored
 * in a vector of the same size as the input vector. The input is considered as being
 * a real vector
 * No inverse transform algorithm is provided
 * @author Francois Germain
 */
public class DWT_db
{
/* FIELDS ****************************************************************/
// The results of the FWT.
private double[] output;
// Number of levels of the FWT
private int nb_levels;
/* CONSTRUCTOR ***********************************************************/
/**
 * Performs the FWT transform and stores the result.
 * Input signals are zero-padded if they do not have a length equal to a
 * power of 2.
 *
 * @param input The signal to be transformed.
 * @param db_indice Indice of the desired Daubechies wavelet.
 * @param depth The number of desired levels of the FWT.
 * @throws Exception Throws an exception if less than
 * 2^(depth-1)+1 input samples are provided
 * or if the Daubechies index is odd or exceeds 20
 */
```

```
public DWT_db( double[] input,

int db_indice,

int depth )

throws Exception

{

// Throw an exception if less than needed samples are provided

if ( input.length < jAudioFeatureExtractor.GeneralTools.Statistics.pow(2, depth-1) + 1 )

throw new Exception( "Only " + input.length + " samples provided.\n" +

"At least " + (jAudioFeatureExtractor.GeneralTools.Statistics.pow(2, depth-1) + 1)

+ " are needed." );

// Throw an exception if the Daubechies indice is not valid

if ( db_indice > 20 || (db_indice % 2) == 1 )

throw new Exception( db_indice + " is not a valid Daubechies indice" );

// Verify that the input size has a number of samples that is a

// power of 2. If not, then increase the size of the array using

// zero-padding. Also creates a zero filled imaginary component

// of the input if none was specified.

int valid_size =
jAudioFeatureExtractor.GeneralTools.Statistics.ensureIsPowerOfN(input.length, 2);

if (valid_size != input.length)

{

double[] temp_array = new double[valid_size];

for (int i = 0; i < input.length; i++)

temp_array[i] = input[i];

for (int i = input.length; i < valid_size; i++)

temp_array[i] = 0.0;

input = temp_array;

}

// Instantiate the arrays to hold the output and copy the input
```

```
// to them
output = new double[valid_size];
System.arraycopy(input, 0, output, 0, valid_size);
// Set the number of levels
nb_levels = depth;
// Perform the spectral recombination stage by stage
int stage;
int frame_size = output.length;
for ( stage = 0 ; stage < depth ; stage++, frame_size /= 2 )
{
int left;
for (left = 0; left < frame_size - wavelet.length; left += 2)
{
for (int i = left ; i < left + wavelet.length ; i++)
{
input[left/2] += output[i] * scaling[db_indice/2][i];
input[(left + frame_size)/2] += output[i] * wavelet[db_indice/2][i];
}
}
//Specific case of the last coefficients
for (; left < frame_size; left += 2)
{
for (int i = left ; i < frame_size ; i++)
{
input[left/2] += output[i] * scaling[db_indice/2][i];
input[(left + frame_size)/2] += output[i] * wavelet[db_indice/2][i];
}
}
```

```
System.arraycopy(input, 0, output, 0, frame_size);

}

}

/* PUBLIC METHODS *********************************************************/

/**

 * Returns a frequency subband

 * @param subband_num Number of the subband

 * @return The subband vector.

 * @throws Exception Throws an exception if the subband number

 * exceeds the number of levels + 1

 */

public double[] getSubband ( int subband_num )

throws Exception

{

// Output vector

double[] subband;

// Throw an exception if the desired subband doesn't exist

if ( subband_num > nb_levels + 1 || subband_num < 1 )

throw new Exception( "There is no subband " + subband_num + "." );

if(subband_num == 1){
// Specific case of the approximate subband (the lowest one)

subband = new double[output.length /
jAudioFeatureExtractor.GeneralTools.Statistics.pow(2, nb_levels)];

System.arraycopy(output, 0, subband, 0, subband.length);

} else {
// Detailed subbands

subband = new double[output.length /
jAudioFeatureExtractor.GeneralTools.Statistics.pow(2, nb_levels + 2 - subband_num)];

System.arraycopy(output,
jAudioFeatureExtractor.GeneralTools.Statistics.pow(2,
nb_levels + 2 - subband_num), subband, 0, subband.length);
```

```
}

return subband;

}


/**

 * Returns the frequency subband labels for each subband referred to by the

 * output values as determined by the given sampling rate.

 *

 * @param sampling_rate The sampling rate that was used to perform

 * the DWT.

 * @return The subband labels.

 */

public double[] getSubbandLabels(double sampling_rate)

{

int number_bins = nb_levels + 1;

double[] labels = new double[number_bins + 1];

labels[0] = 0.0;

for (int subband = 2; subband < labels.length + 1; subband++)

labels[subband-1] = sampling_rate / (double)
jAudioFeatureExtractor.GeneralTools.Statistics.pow(2, nb_levels + 2 - subband);

return labels;

}

/* STATIC FIELDS *******************************************************/

// The coefficients of the Daubechies scaling function filters (low-pass)

final static double[][] scaling;

// The coefficients of the Daubechies db8 wavelet filter (high-pass)

final static double[][] wavelet;

static {
scaling = new double[][]{ {
7.071067811865475244008443621048490392848359376884740365883398e-01,
```

```
7.071067811865475244008443621048490392848359376884740365883398e-01
},
{
4.829629131445341433748715998644486838169524195042022752011715e-01,
8.365163037378079055752937809168732034593703883484392934953414e-01,
2.241438680420133810259727622404003554678835181842717613871683e-01,
-1.294095225512603811744494188120241641745344506599652569070016e-01
},
{
3.326705529500826159985115891390056300129233992450683597084705e-01,
8.068915093110925764944936040887134905192973949948236181650920e-01,
4.598775021184915700951519421476167208081101774314923066433867e-01,
-1.350110200102545886963899066993744805622198452237811919756862e-01,
-8.544127388202666169281916918177331153619763898808662976351748e-02,
3.522629188570953660274066471551002932775838791743161039893406e-02
},
{
2.303778133088965008632911830440708500016152482483092977910968e-01,
7.148465705529156470899219552739926037076084010993081758450110e-01,
6.308807679298589078817163383006152202032229226771951174057473e-01,
-2.798376941685985421141374718007538541198732022449175284003358e-02,
-1.870348117190930840795706727890814195845441743745800912057770e-01,
3.084138183556076362721936253495905017031482172003403341821219e-02,
3.288301166688519973540751354924438866454194113754971259727278e-02,
-1.059740178506903210488320852402722918109996490637641983484974e-02
},
{
1.601023979741929144807237480204207336505441246250578327725699e-01,
6.038292697971896705401193065250621075074221631016986987969283e-01,
7.243085284377729277280712441022186407687562182320073725767335e-01,
1.384281459013207315053971463390246973141057911739561022694652e-01,
-2.422948870663820318625713794746163619914908080626185983913726e-01,
-3.224486958463837464847975506213492831356498416379847225434268e-02,
7.757149384004571352313048938860181980623099452012527983210146e-02,
-6.241490212798274274190519112920192970763557165687607323417435e-03,
-1.258075199908199946850973993177579294920459162609785020169232e-02,
3.335725285473771277998183415817355747636524742305315099706428e-03
},
{
1.115407433501094636213239172409234390425395919844216759082360e-01,
4.946238903984530856772041768778555886377863828962743623531834e-01,
7.511339080210953506789344984397316855802547833382612009730420e-01,
3.152503517091976290859896548109263966495199235172945244404163e-01,
-2.262646939654398200763145006609034656705401539728969940143487e-01,
-1.297668675672619355622896058765854608452337492235814701599310e-01,
9.750160558732304910234355253812534233983074749525514279893193e-02,
2.752286553030572862554083950419321365738758783043454321494202e-02,
-3.158203931748602956507908069984866905747953237314842337511464e-02,
5.538422011614961392519183980465012206110262773864964295476524e-04,
4.777257510945510639635975246820707050230501216581434297593254e-03,
-1.077301085308479564852621609587200035235233609334419689818580e-03
},
{
7.785205408500917901996352195789374837918305292795568438702937e-02,
3.965393194819173065390003909368428563587151149333287401110499e-01,
7.291320908462351199169430703392820517179660611901363782697715e-01,
```

```
4.697822874051931224715911609744517386817913056787359532392529e-01,
-1.439060039285649754050683622130460017952735705499084834401753e-01,
-2.240361849938749826381404202332509644757830896773246552665095e-01,
7.130921926683026475087657050112904822711327451412314659575113e-02,
8.061260915108307191292248035938190585823820965629489058139218e-02,
-3.802993693501441357959206160185803585446196938467869898283122e-02,
-1.657454163066688065410767489170265479204504394820713705239272e-02,
1.255099855609984061298988603418777957289474046048710038411818e-02,
4.295779729213665211321291228197322228235350396942409742946366e-04,
-1.801640704047490915268262912739550962585651469641090625323864e-03,
3.537137999745202484462958363064254310959060059520040012524275e-04
},
{
5.441584224310400995500940520299935503599554294733050397729280e-02,
3.128715909142999706591623755057177219497319740370229185698712e-01,
6.756307362972898068078007670471831499869115906336364227766759e-01,
5.853546836542067127712655200450981944303266678053369055707175e-01,
-1.582910525634930566738054787646630415774471154502826559735335e-02,
-2.840155429615469265162031323741647324684350124871451793599204e-01,
4.724845739132827703605900098258949861948011288770074644084096e-04,
1.287474266204784588570292875097083843022601575556488795577000e-01,
-1.736930100180754616961614886809598311413086529488394316977315e-02,
-4.408825393079475150676372323896350189751839190110996472750391e-02,
1.398102791739828164872293057263345144239559532934347169146368e-02,
8.746094047405776716382743246475640180402147081140676742686747e-03,
-4.870352993451574310422181557109824016634978512157003764736208e-03,
-3.917403733769470462980803573237762675229350073890493724492694e-04,
6.754494064505693663695475738792991218489630013558432103617077e-04,
-1.174767841247695337306282316988909444086693950311503927620013e-04
},
{
3.807794736387834658869765887955118448771714496278417476647192e-02,
2.438346746125903537320415816492844155263611085609231361429088e-01,
6.048231236901111119030768674342361708959562711896117565333713e-01,
6.572880780513005380782126390451732140305858669245918854436034e-01,
1.331973858250075761909549458997955536921780768433661136154346e-01,
-2.932737832791749088064031952421987310438961628589906825725112e-01,
-9.684078322297646051350813353769660224825458104599099679471267e-02,
1.485407493381063801350727175060423024791258577280603060771649e-01,
3.072568147933337921231740072037882714105805024670744781503060e-02,
-6.763282906132997367564227482971901592578790871353739900748331e-02,
2.509471148314519575871897499885543315176271993709633321834164e-04,
2.236166212367909720537378270269095241855646688308853754721816e-02,
-4.723204757751397277925707848242465405729514912627938018758526e-03,
-4.281503682463429834496795002314531876481181811463288374860455e-03,
1.847646883056226476619129491125677051121081359600318160732515e-03,
2.303857635231959672052163928245421692940662052463711972260006e-04,
-2.519631889427101369749886842878606607282181543478028214134265e-04,
3.934732031627159948068988306589150707782477055517013507359938e-05
},
{
2.667005790055555358661744877130858277192498290851289932779975e-02,
1.881768000776914890208929736790939942702546758640393484348595e-01,
5.272011889317255864817448279595081924981402680840223445318549e-01,
6.884590394536035657418717825492358539771364042407339537279681e-01,
2.811723436605774607487269984455892876243888859026150413831543e-01,
```

```
-2.498464243273153794161018979207791000564669737132073715013121e-01,
-1.959462743773770435042992543190981318766776476382778474396781e-01,
1.273693403357932600826772332014009770786177480422245995563097e-01,
9.305736460357235116035228983545273226942917998946925868063974e-02,
-7.139414716639708714533609307605064767292611983702150917523756e-02,
-2.945753682187581285828323760141839199388200516064948779769654e-02,
3.321267405934100173976365318215912897978337413267096043323351e-02,
3.606553566956169655423291417133403299517350518618994762730612e-03,
-1.073317548333057504431811410651364448111548781143923213370333e-02,
1.395351747052901165789318447957707567660542855688552426721117e-03,
1.992405295185056117158742242640643211762555365514105280067936e-03,
-6.858566949597116265613709819265714196625043336786920516211903e-04,
-1.164668551292854509514809710258991891527461854347597362819235e-04,
9.358867032006959133405013034222854399688456215297276443521873e-05,
-1.326420289452124481243667531226683305749240960605829756400674e-05
}};

wavelet = new double[10][];
for(int i=0;i < 10;i++)
{
wavelet[i] = new double[2*i];
for(int j=0;j < 2*i;j++)
{
wavelet[i][j] = scaling[i][2*i-1-j] * (1 - 2*(j%2));
}
}
}
}
```