Audio Analysis using the Discrete Wavelet Transform

George Tzanetakis, Georg Essl, Perry Cook*
Computer Science Department *also Music Department
Princeton
35 Olden Street, Princeton NJ 08544
USA
gtzan@cs.princeton.edu http://www.cs.princeton.edu/~gtzan

Abstract: - The Discrete Wavelet Transform (DWT) is a transformation that can be used to analyze the temporal and spectral properties of non-stationary signals like audio. In this paper we describe some applications of the DWT to the problem of extracting information from non-speech audio. More specifically automatic classification of various types of audio using the DWT is described and compared with other traditional feature extractors proposed in the literature. In addition, a technique for detecting the beat attributes of music is presented. Both synthetic and real world stimuli were used to evaluate the performance of the beat detection algorithm.

Key-Words: - audio analysis, wavelets, classification, beat extraction

1 Introduction

Digital audio is becoming a major part of the average computer user experience. The increasing amounts of available audio data require the development of new techniques and algorithms for structuring this information. Although there has been a lot of research on the problem of information extraction from speech signals, work on non-speech audio like music has only appeared recently.

The Wavelet Transform (WT) and more particularly the Discrete Wavelet Transform (DWT) is a relatively recent and computationally efficient technique for extracting information about non-stationary signals like audio. This paper explores the use of the DWT in two applications. The first application is the automatic classification of non-speech audio data using statistical pattern recognition with feature vectors derived from the wavelet analysis. The second application is the extraction of beat attributes from music signals.

The paper is organized as follows: Section 2 describes related work. An overview of the DWT is given in Section 3. Section 4 describes the DWT-based feature extraction and compares it with standard feature front ends that have been used in the past. Results from automatic classification of three data collections are provided. Beat detection using the DWT is described in Section 5. The results of the beat detection are evaluated using synthetic and real world stimuli. Implementation information is provided in section 6. Some directions for future research are given in Section 6.

2 Related Work

A general overview of audio information retrieval for speech and other types of audio is given in [1]. A robust classifier for Music vs Speech is described in [2] and [3] describes a system for content-based retrieval of short isolated sounds. Automatic beat extraction and tempo analysis is explored in [4]. Introductions to wavelets can be found in [5,6].
Wavelets for audio and especially music have been explored by [7]. The most relevant work to our research are the two systems for content-based indexing and retrieval based on wavelets that are described in [8,9]. In both cases Query-by-Example (QBE) similarity retrieval is studied.

3 The Discrete Wavelet Transform

The Wavelet Transform (WT) is a technique for analyzing signals. It was developed as an alternative to the short time Fourier Transform (STFT) to overcome problems related to its frequency and time resolution properties. More specifically, unlike the STFT that provides uniform time resolution for all frequencies the DWT provides high time resolution and low frequency resolution for high frequencies and high frequency resolution and low time resolution for low frequencies. In that respect it is similar to the human ear which exhibits similar time-frequency resolution characteristics.

The Discrete Wavelet Transform (DWT) is a special case of the WT that provides a compact representation of a signal in time and frequency that can be computed efficiently.

The DWT is defined by the following equation:

$$W(j,k) = \sum_j \sum_k x(k) 2^{-j/2} \psi(2^{-j} n - k) \qquad (1)$$

where $\psi(t)$ is a time function with finite energy and fast decay called the mother wavelet. The DWT analysis can be performed using a fast, pyramidal algorithm related to multirate filterbanks [10].

As a multirate filterbank the DWT can be viewed as a constant Q filterbank with octave spacing between the centers of the filters. Each subband contains half the samples of the neighboring higher frequency subband. In the pyramidal algorithm the signal is analyzed at different frequency bands with different resolution by decomposing the signal into a coarse approximation and detail information. The coarse approximation is then further decomposed using the same wavelet decomposition step. This is achieved by successive highpass and lowpass filtering of the time domain signal and is defined by the following equations:

$$y_{high}[k] = \sum_n x[n] g[2k - n] \qquad (2)$$

$$y_{low}[k] = \sum_n x[n] h[2k - n] \qquad (3)$$

where $y_{high}[k], y_{low}[k]$ are the outputs of the highpass (g) and lowpass (h) filters, respectively after subsampling by 2. Because of the downsampling the number of resulting wavelet coefficients is exactly the same as the number of input points. A variety of different wavelet families have been proposed in the literature. In our implementation, the 4 coefficient wavelet family (DAUB4) proposed by Daubechies [11] is used.

4 Feature Extraction & Classification

The extracted wavelet coefficients provide a compact representation that shows the energy distribution of the signal in time and frequency. In order to further reduce the dimensionality of the extracted feature vectors, statistics over the set of the wavelet coefficients are used. That way the statistical characteristics of the "texture" or the "music surface" of the piece can be represented. For example the distribution of energy in time and frequency for music is different from that of speech.

The following features are used in our system:

- The mean of the absolute value of the coefficients in each subband. These features provide information about the frequency distribution of the audio signal.
- The standard deviation of the coefficients in each subband. These features provide information about the amount of change of the frequency distribution
- Ratios of the mean values between adjacent subbands. These features also provide information about the frequency distribution.

A window size of 65536 samples at 22050 Hz sampling rate with hop size of 512 seconds is used as input to the feature extraction. This corresponds to approximately 3 seconds. Twelve levels (subbands) of coefficients are used resulting in a feature vector with 45 dimensions (12 + 12 + 11).

To evaluate the extracted features they were compared in three classification experiments with two feature sets that have been proposed in the literature. The first feature set consists of features extracted using the STFT [3]. The second feature set consists of features extracted from Mel-Frequency Cepstral Coefficients (MFCC) [12] which are perceptually motivated features that have been used in speech recognition research.

The following three classification experiments were conducted:

- **MusicSpeech (Music, Speech)**
  126 files
- **Voices (Male, Female, Sports Announcing)**
  60 files
- **Classical (Choir, Orchestra, Piano, String Quartet)**
  320 files

where each file is 30 seconds long. Figure 1 summarizes the classification results. The evaluation was performed using a 10-fold paradigm where a random 10% of the audio files where to test a classifier trained on the remaining 90%. This random partition process was repeated 100 times and the results were averaged. Feature vectors extracted from the same audio file where never split into testing and training datasets to avoid false high accuracy. A Gaussian classifier was trained using the three feature sets (DWTC, MFCC, STFTC). The results of figure 3 show much better results than random classification are achieved in all cases and that the performance of the DWTC feature set is comparable to the other two feature sets.

5 Beat detection

Beat detection is the automatic extraction of rhythmic pulse from music signals. In this section we describe an algorithm based on the DWT that is capable of automatically extracting beat information from real world musical signals with arbitrary timbral and polyphonic complexity. The work in [6] shows that it is possible to construct an automatic beat detection algorithm with these capabilities.

The beat detection algorithm is based on detecting the most salient periodicities of the signal. Figure 3 shows a flow diagram of the beat detection algorithm. The signal is first decomposed into a number of octave frequency bands using the DWT. After that the time domain amplitude envelope of each band is extracted separately. This is achieved by low pass filtering each band, applying full wave rectification and downsampling. The envelopes of each band are then summed together and an autocorrelation function is computed. The peaks of the autocorrelation function correspond to the various periodicities of the signal's envelope. More specifically the following stages are performed:

1. LPF

Low pass filtering of the signal with a One Pole filter with alpha value 0.99 given by the equation:

$$y[n] = (1-\alpha)x[n] - \alpha y[n] \qquad (3)$$

2. FWR

Full wave rectification given by the equation:

$$y[n] = abs(x[n]) \qquad (4)$$

3. DOWNSAMPLING

$$y[n] = x[kn] \qquad (5)$$

4. **NORM**

Normalization in each band (mean removal) :

$$y[n] = x[n] - E[x[n]] \qquad (6)$$

5. ACRL

Autocorrelation given by the equation:

$$y[k] = \frac{1}{N}\sum_{n=0}^{N-1} x[n]x[n+k] \qquad (7)$$

The first five peaks of the autocorrelation function are detected and their corresponding periodicities in beats per minute (bpm) are calculated and added in a histogram. This process is repeated by iterating over the signal. The periodicity corresponding to the most prominent peak of the final histogram is the estimated tempo in bpm of the audio file. A block diagram of this process is shown in figure 2.

A number of synthetic beat patterns were constructed to evaluate the beat detection algorithm. Bass-drum, snare, hi-hat and cymbal sounds were used, with addition of woodblock and three tom-tom sounds for the unison examples. The simple unison beats versus cascaded beats of differing sounds are used to check for phase sensitivity in the algorithm. Simple and complex beat patterns are used to gauge the possibility to find the dominant beat timing. For this purpose, the simple beat pattern in the diagram above has been synthesized at 5 different speeds. The simple rhythm pattern has a simple rhythmic substructure with single-beat regularity in the single instrument sections, whereas the complex pattern shows a complex rhythmic substructure across and within the single instrument sections. Figure 3 shows the synthetic beat patterns that were used for evaluation in tablature notation. In all cases the algorithm was able to detect the basic beat of the pattern.

In addition to the synthetic beat patterns the algorithm was applied to real world music signals. To evaluate the algorithm's performance it was compared to the bpm detected manually by tapping the mouse with the music. The average time difference between the taps was used as the manual beat estimate. Twenty files containing a variety of music styles were used to evaluate the algorithm (5 HipHop, 3 Rock, 6 Jazz, 1 Blues, 3 Classical, 2 Ethnic). For most of the files the prominent beat was detected clearly (13/20) (i.e the beat corresponded to the highest peak of the histogram). For (5/20) files the beat was detected as a histogram peak but it was not the highest, and for (2/20) no peak corresponding to the beat was found. In the pieces that the beat was not detected there was no dominant periodicity (these pieces were either classical music or jazz). In such cases humans rely on more high-level information like grouping, melody and harmonic progression to perceive the primary beat from the interplay of multiple periodicities. Figure 4 shows the histograms of two classical music pieces and two modern pop music pieces. The prominent peaks correspond to the detected periodicities.

6 Implementation

The algorithms described have been implemented and evaluated using *MARSYAS* [13] a free software framework for rapid development of computer audition applications written in C++. Both algorithms (feature extraction-classification, beat detection) can be performed in real-time using this framework. *MARSYAS* can be downloaded from:
http://www.cs.princeton.edu/marsyas.html

7 Future work

The exact details of feature calculation for the STFT and MFCC have been explored extensively in the literature. On the other hand, wavelet based features have appeared relatively recently. Further improvements in classification accuracy can be expected with more careful experimentation with the exact details of the parameters. We also plan to investigate the use of other wavelet families. Another interesting direction is combining features from different analysis techniques to improve classification accuracy. A general methodology for audio segmentation based on "texture" without attempting classification is described in [14]. We plan to apply this methodology using the DWT.

From inspecting the beat histograms it is clear that more information than just the primary beat is contained (see Figure 4). For example it is easy to separate visually classical music from modern popular music based on their beat histograms. Modern popular music has more pronounced peaks corresponding to its strong rhythmic characteristics. We plan to use the beat histograms to extract genre related information. A comparison of the beat detection algorithm described in [4] with our scheme on the same dataset is also planned for the future. In addition, more careful studies using the synthetic examples are planned for the future. These studies will show if it is possible to detect the phase and periodicity of multiple lines using the DWT.

References:

[1] Jonathan Foote, An Overview of Audio Information Retrieval, ACM Multimedia Systems, Vol.7, 1999, pp. 2-10.

[2] Eric D. Scheirer, Malcolm Slaney, "Construction and evaluation of a robust multifeature speech/music discriminator," *IEEE Transactions on Acoustics, Speech and Signal Processing*, 1997, 1331-1334.

[3] E. Wold *et al.*, "Content-based classification, search and retrieval of audio data." *IEEE Multimedia Magazine*, Vol. 3, No. 2, 1996.

[4] Eric D. Scheirer, "Tempo and beat analysis of acoustic musical signals," *J. Acoust. Soc. Am.* Vol. 103, No. 1, 1998, 558-601.

[5] Special issue on Wavelets and Signal Processing, *IEEE Trans. Signal Processing*, Vol. 41, Dec. 1993.

[6] R. Polikar, The Wavelet Tutorial, http://www.public.iastate.edu/rpolikar/wavelets/Wttutorial.html.

[7] R.Kronland-Martinet, J.Morlet and A.Grossman Analysis of sound patterns through wavelet transform", *International Journal of Pattern Recognition and Artificial Intelligence,*Vol. 1(2), 1987, 237-301.

[8] S. R. Subramanya, "Experiments in Indexing Audio Data," Tech. Report, GWU-IIST, January 1998.

[9] S. R. Subramanya *et al.*, "Transform-Based Indexing of Audio Data for Multimedia Databases," *IEEE Int'l Conference on Multimedia Systems*, Ottawa, June 1997.

[10] S.G Mallat "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation" IEEE.Transactions on Pattern Analysis and Machine Intelligence, Vol.11,1989,674-693

[11] I.Daubechies "Orthonormal Bases of Compactly Supported Wavelets" Communications on Pure and Applied Math. Vol.41 1988, 909-996

[12] M.Hunt, M.Lenning and P.Mermelstein. Experiments in syllable-based recognition of continuous speech", Proc. Inter.Conference on Acoustics, Speech and Signal Processing (ICASS), 1980

[13] G. Tzanetakis, P.Cook "MARSYAS: A framework for audio analysis", *Organised sound,*Vol.4(3), 2000

[14] G. Tzanetakis, P.Cook "Multi-feature Audio Segmentation for Browsing and Annotation", *Proc.IEEE Workshop on Appl. Signal Proc.. to Audio and Acoustics* (WASPAA),1999

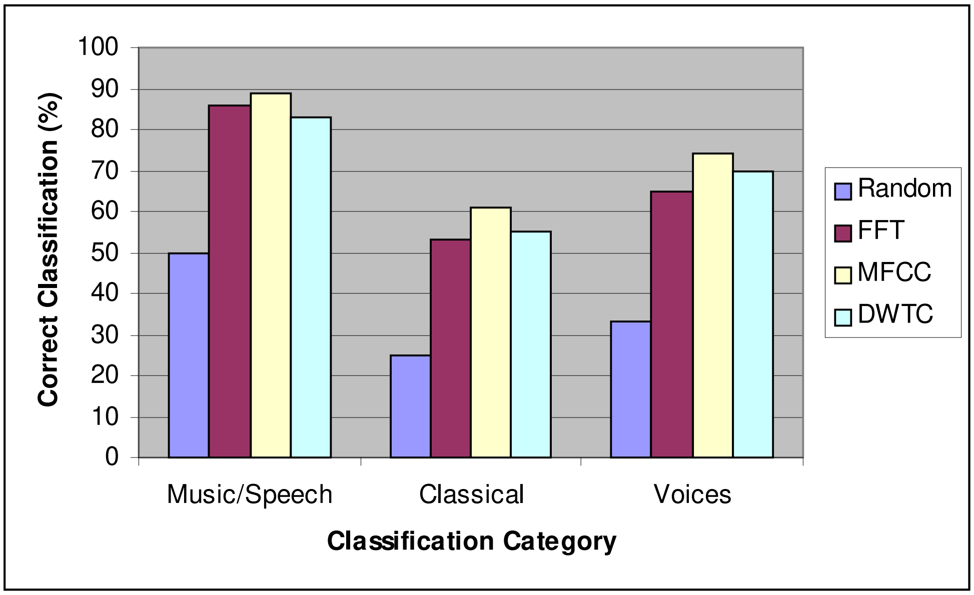

Figure 1: Comparison of classification methods on various data-sets.

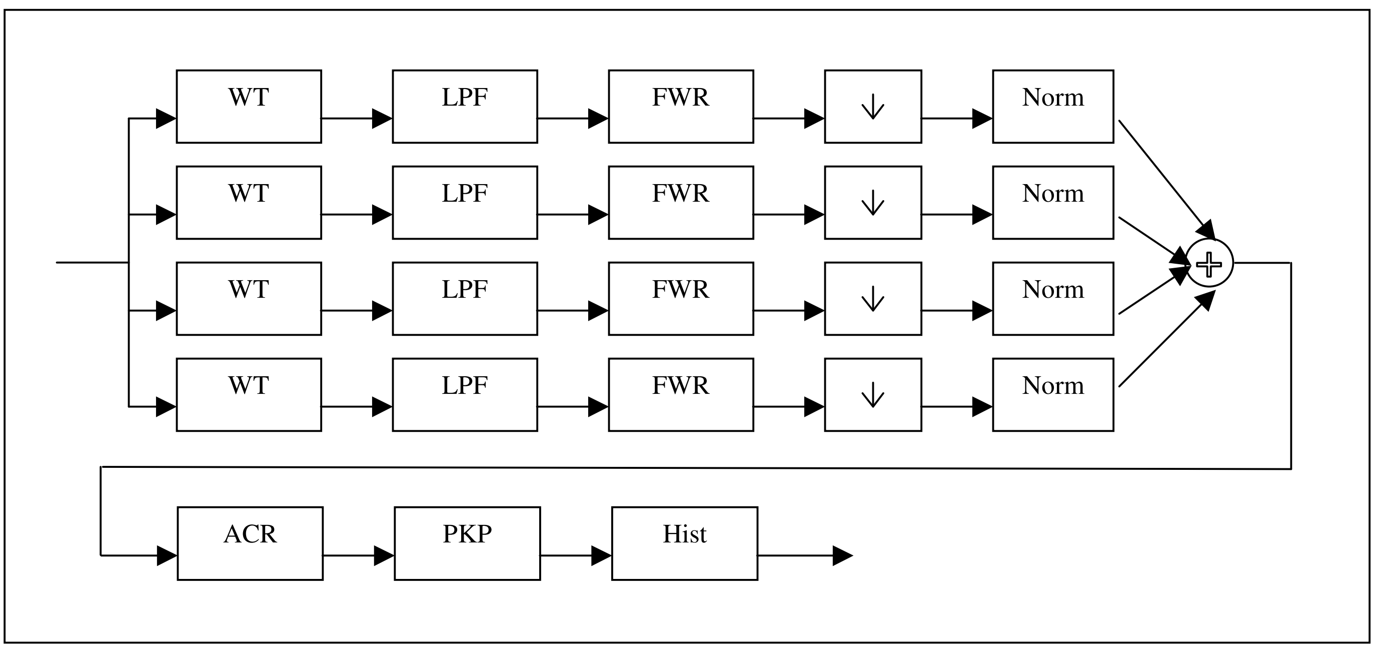

Figure 2: Block-diagram of beat-detection algorithm based on the Discrete Wavelet Transform (DWT).

CY															•	
HH	•		•		•		•		•		•		•		•	
SD					•								•			
BD	•								•							

CY			•		•	•		•			•		•	•	•	•
HH	•			•		•	•		•		•	•		•		•
SD					•			•					•			
BD	•								•		•	•				•

CY	•								•							
HH	•								•							
WB	•								•							
HT	•								•							
MT	•								•							
LT	•								•							
SD	•								•							
BD	•								•							

CY			•								•					
HH		•								•						
WB				•								•				
HT						•								•		
MT							•								•	
LT								•								•
SD					•								•			
BD	•								•							

Figure 3: Synthetic beat patterns (CY = cymbal, HH = hi-hat, SD = snare drum, BD = Base Drum, WB = wood block, HT/MT/LT = tom-toms)

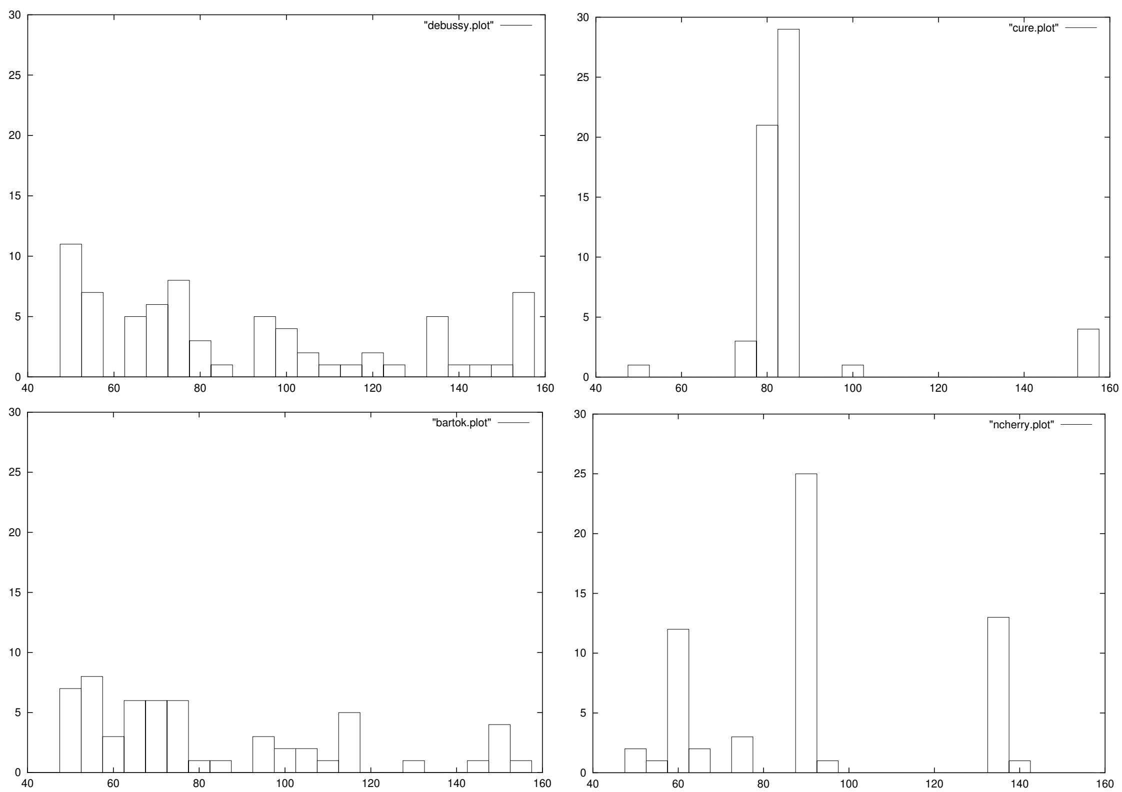

Figure 4: Beat histograms for classical (left) and contemporary popular music (right).