# AN IMPROVED IMPLEMENTATION FOR AN AUDITORY-INSPIRED FFT MODEL WITH APPLICATION IN AUDIO CLASSIFICATION

*Wei Chu and Benoît Champagne*

Department of Electrical and Computer Engineering
McGill University, Montreal, Quebec, Canada, H3A 2A7
wei.chu@mail.mcgill.ca, benoit.champagne@mcgill.ca

## ABSTRACT

In this paper, we present an improved implementation for an auditory-inspired FFT-based model which calculates a noise-robust FFT spectrum. Through the use of characteristic frequency (CF) values of the cochlear filters in an *early* auditory (EA) model for power spectrum selection, and the use of a pair of running averages for the implementation of self-normalization, the proposed FFT model allows more flexibility in the extraction of audio features. To evaluate the performance of the proposed FFT model, a speech/music/noise classification task is carried out wherein a decision tree learning algorithm (C4.5) is used as the classifier. Audio features used for classification include the mel-frequency cepstral coefficient (MFCC) features, a set of conventional spectral features, and spectral features calculated using the proposed FFT model. Compared to the conventional MFCC and spectral features, the spectral features based on the proposed FFT model show more robust performance in noisy test cases.

## 1. INTRODUCTION

The past decade has seen extensive research on audio classification and segmentation algorithms. Many audio classification algorithms have been proposed along with excellent performance being reported. However, the issue of background noise, specifically, the effect of background noise on the performance of classification, has not been investigated widely. Test results in [1–4] indicate that a classification algorithm trained using clean sequences may fail to work properly when the actual testing sequences contain background noise with certain SNR levels. Recently, an early auditory (EA) model [5] that calculates a so-called auditory spectrum, has been employed in audio classification where excellent noise-robust performance is reported [2]. The EA model introduced in [5] can be simplified as the three-stage process shown in Fig. 1, which describes the transformation of an audio signal into an internal neural representation referred to as auditory spectrum.

According to [5], conceptually, the auditory spectrum is an averaged ratio of quantities $\mathcal{E}_d$ and $\mathcal{E}_c$, where $\mathcal{E}_d$ and $\mathcal{E}_c$ are the signal energies passing through the differential filters $\partial_s h(t,s)$ and the cochlear filters $h(t,s)$ respectively. Considering that the cochlear filters are broad while the differential filters are narrow and centered around the same frequencies, $\mathcal{E}_c$ can be viewed as a smoothed version of $\mathcal{E}_d$. Therefore, the auditory spectrum is a self-normalized spectral profile [5]. Further analysis reveals that a spectral peak receives a relatively small normalization factor (i.e., $\mathcal{E}_c$ is relatively small) whereas a spectral valley receives a relatively large normalization factor. The difference in the normalization is known as spectral enhancement or noise suppression.

Unfortunately, this EA model is characterized by high computational requirements and the use of nonlinear processing. In [4], inspired by the self-normalization property of this EA model, we have proposed a simplified FFT-based model whose noise-robustness has been verified through a three-class audio classification task. The proposed FFT model employs a simple grouping scheme to reduce the dimension of the power spectrum vector. However, this scheme fails to give a clear interpretation of the meaning of the frequency index. In applications where frequency-dependent audio features need to be extracted (e.g., spectral centroid, bandwidth), it would be more appropriate, instead of the simple grouping scheme we have proposed, to group or select power spectrum components based on the original constant-$Q$ bandpass filters $h(t,s)$.

In this paper, we present an improved implementation for the FFT model proposed in [4]. With the proposed implementation, the FFT model allows more flexibility in the extraction of audio features. The introduced improvements include the use of characteristic frequency (CF) values of the cochlear filters in an EA model for power spectrum selection, and the use of a pair of running averages for the implementation of self-normalization. A speech/music/noise classification task is carried out to evaluate the performance of the new FFT model wherein a decision tree learning algorithm (C4.5 [6]) is used as the classifier. Mel-frequency cepstral coefficient (MFCC) features, so-called conventional spectral features (which include energy, spectral flux, spectral rolloff point, spectral centroid and bandwidth) and the spectral features based on the proposed FFT model are calculated for audio classification. Compared to the conventional MFCC and spectral features, the spectral features based on the proposed FFT model show more robust performance in noisy test cases.

The paper is organized as follows. The proposed implementation of the FFT model is detailed in Section 2. Section 3 explains the extraction of audio features. Section 4 discusses the setup of the classification tests. Test results are presented in Section 5.

## 2. A NEW IMPLEMENTATION FOR THE AUDITORY-INSPIRED FFT-BASED MODEL

In this work, by making use of the CF values of the bandpass filter set of the EA model [5], and by introducing a pair of running averages, we propose an improved implementation for the FFT-based model presented in [4], as illustrated in Fig. 2. The details of this model are presented below.

### 2.1. Normalization of the Input Signal

To make the algorithm adaptable to input signals with different energy levels, each input audio clip (with a length of one second) is

1-4244-1017-7/07/$25.00 ©2007 IEEE

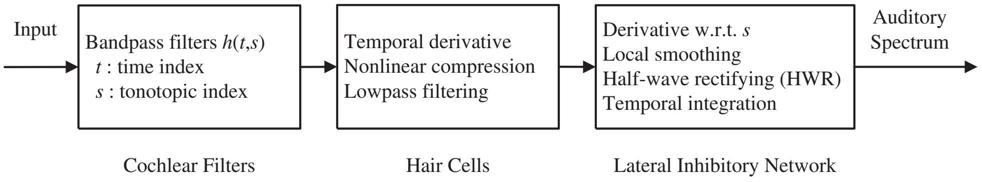

**Fig. 1**. Schematic description of the EA model.

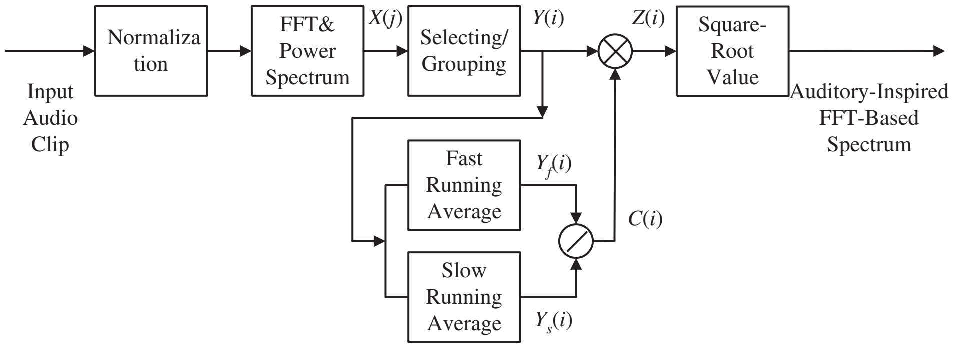

**Fig. 2**. Schematic description of the FFT-based model.

normalized with respect to the square-root value of its average energy.

### 2.2. Calculation of a Narrow-Band Power Spectrum

Using the normalized audio signal, a narrow-band power spectrum is calculated through an $M$-point FFT algorithm. To determine an appropriate value for $M$, we have to trade performance against complexity.

Based on [7], the cochlear filters are modeled as a set of constant-$Q$ bandpass filters. In [8], such a set of 129 bandpass filters is implemented where the corresponding CF values $F_k$ are determined by

$$F_k = 2^{(k-32)/24} F_0 \quad \text{(Hz)}, \quad k = 1, 2, \cdots, 129 \qquad (1)$$

where $F_0 = 440$ Hz. According to (1), the CF values cover a range from 180 Hz to 7246 Hz. The difference between two neighboring CF values is as low as about 5.27 Hz (for $k = 1$ and 2). For a signal sampled at 16 kHz which is assumed in this study, even with a 2048-point FFT, such a small frequency interval cannot be resolved. Meanwhile, since the CF values are logarithmically located, the frequency resolution achieved from a 2048-point or even higher-order FFT algorithm is more than necessary for the high frequency bands. In this work, we use an $M = 1024$ point FFT to achieve a trade-off between frequency resolution and computational complexity. The length of the analysis window is 30 ms and the overlap is 20 ms.

### 2.3. Power Spectrum Selection

To reduce the dimension of the obtained power spectrum vector, a simple selection scheme is proposed as follows. First, we extend the values of $k$ in (1), i.e., from -8 to 132. Or equivalently, (1) is modified as

$$F_k = 2^{(k-41)/24} F_0 \quad \text{(Hz)}, \quad k = 1, 2, \cdots, 141. \qquad (2)$$

**Table 1**. Frequency index values of $N_k$ and $\phi_i$

| $k$ | $N_k$ | $i$ | $\phi_i$ |
|---|---|---|---|
| 1 | 8 | 1 | 8 |
| 2 | 9 | 2 | 9 |
| 3 | 9 | - | - |
| 4 | 9 | - | - |
| 5 | 9 | - | - |
| 6 | 10 | 3 | 10 |
| 7 | 10 | - | - |
| 8 | 10 | - | - |
| 9 | 11 | 4 | 11 |
| 10 | 11 | - | - |
| ⋮ | ⋮ | ⋮ | ⋮ |
| 140 | 491 | 119 | 491 |
| 141 | 505 | 120 | 505 |

For each $F_k$, the corresponding frequency index value $N_k$ is determined by

$$N_k = \text{int}\left(\frac{F_k M}{F_s}\right), \quad k = 1, 2, \cdots, 141 \qquad (3)$$

where the function int$(x)$ returns the greatest integer less than or equal to $x$, and $F_s$ is the sampling frequency. After discarding the repeated $N_k$ values and renumbering the remaining values, we obtain a set of 120 characteristic frequency index values $\phi_i$, $i = 1, 2, \cdots, 120$, as illustrated in Table 1.

Using frequency index values $\phi_i$, the power spectrum selection is as follows

$$Y(i) = X(\phi_i), \quad i = 1, 2, \cdots, 120. \qquad (4)$$

Based on this selection scheme, a set of $M/2$, i.e., 512, power spectrum components is transformed into a 120-dimensional vector, with its frequency index corresponding to a specific CF value of the original cochlear filters.

### 2.4. Spectral Self-Normalization

In [4], a local self-normalization is implemented through the use of a pair of wide and narrow windows in the frequency domain. Below, we propose a new implementation which is simpler and easier to use than the one in [4].

According to [5], the cochlear filters are broad and highly asymmetric, and the differential filters are narrowly tuned and centered around the same frequencies. Based on the magnitude responses of a pair of cochlear filter and differential filter given in [5], an iterative running average is defined over the frequency index $i$ as follows

$$Y_r(i) = (1-\alpha)Y_r(i-1) + \alpha Y(i) \tag{5}$$

where $0 \leq \alpha \leq 1$, and $Y(i)$ and $Y_r(i)$ are the input and averaged output respectively. A relatively large $\alpha$ corresponds to a "fast" running average, while a relatively small $\alpha$ results in a "slow" running average. A slow and fast running average are employed here to simulate a cochlear filter and a differential filter respectively.

Let $Y_f(i)$ and $Y_s(i)$ represent the outputs from a fast and a slow running averages, respectively. $Y_s(i)$ may be viewed as a smoothed version of $Y_f(i)$. Based on $Y_f(i)$ and $Y_s(i)$, a self-normalization coefficient at frequency index $i$, $C(i)$, is defined as

$$C(i) = \frac{Y_f(i)}{Y_s(i)}, \quad i = 1, 2, \cdots, 120. \tag{6}$$

Finally, the proposed auditory-inspired FFT-based spectrum at frequency index $i$ is obtained by multiplying the selected power spectrum at frequency index $i$, i.e., $Y(i)$, with the corresponding self-normalization coefficient $C(i)$, and applying a square-root operation.

Compared to the self-normalization scheme in [4], the new implementation proposed here is simpler and easier to use since it only involves two parameters to adjust, i.e., a fast and a slow running average coefficients. Besides, by making use of the CF values of the original bandpass filters, a relationship is created between the frequency index of the proposed FFT spectrum vector and the physical frequency value. Therefore, the proposed FFT spectrum allows more flexibility in the extraction of different audio features.

## 3. AUDIO FEATURES

In this work, three sets of frame-level audio features are calculated which include mel-frequency cepstral coefficient (MFCC) features, the conventional spectral features which include energy, spectral flux, spectral rolloff point, spectral centroid and bandwidth, and spectral features based on the proposed FFT model. The corresponding clip-level features, which are calculated over a one-second time window, are the statistical mean and variance values of these frame-level features. The clip-level features are used for the training and testing of the algorithm. The details of the frame-level features are presented below.

### 3.1. MFCC Features

Being widely used in speech/speaker recognition, MFCCs [9] are also useful in audio classification. For the purpose of performance comparison, the conventional MFCCs are used in this work. A Matlab toolbox developed by Slaney [10] is used to calculate a set of 13 conventional MFCCs.

### 3.2. Spectral Features

A set of spectral features are calculated using the conventional FFT spectrum and the proposed FFT spectrum. These features include energy, spectral flux, spectral rolloff point, spectral centroid, and bandwidth.

**Energy:** The energy is a simple yet reliable feature for audio classification. In this work, we calculate for each frame the total energy and the energies of 3 subbands covering frequency ranges of 0-1 kHz, 1-2 kHz and 2-4 kHz respectively.

**Spectral flux:** The spectral flux is a measure of spectral change which comes in different forms. The *1st*-order spectral flux is defined as the 2-norm of the frame-to-frame magnitude spectrum difference vector [11, 12]:

$$SF1_n = \sqrt{\sum_{k=1}^{K} (A_{n+1}[k] - A_n[k])^2}. \tag{7}$$

where $A_n[k]$ is the *kth* component of the magnitude spectrum vector (either the conventional spectrum vector or the proposed FFT spectrum vector) for the *nth* frame signal, and $K$ is the size of the magnitude spectrum vector $A_n$. The *2nd*-order spectral flux, $SF2_n$, is calculated similarly as follows

$$SF2_n = \sqrt{\sum_{k=1}^{K} (\Delta A_{n+1}[k] - \Delta A_n[k])^2} \tag{8}$$

where $\Delta A_n[k] = A_{n+1}[k] - A_n[k]$.

**Spectral rolloff point:** Scheirer and Slaney defined the spectral rolloff point as the *95th* percentile of the power spectrum distribution [11]. It is a measure of the skewness of the spectral shape. In this work, two spectral rolloff points are calculated which correspond to the *50th* and *90th* percentiles of the power spectrum distribution respectively.

**Spectral centroid:** As a measure of the centroid of the magnitude spectrum, the spectral centroid, or brightness, can be defined as [13, 14]

$$SC_n = \left( \sum_{k=1}^{K} k A_n[k] \right) \Big/ \sum_{k=1}^{K} A_n[k] \tag{9}$$

where $SC_n$ denotes the spectral centroid.

**Bandwidth:** Here, the bandwidth is obtained as the magnitude-weighted average of the differences between the frequency indices and the centroid [13, 14]. The bandwidth can be expressed as follows

$$BW_n = \sqrt{\left( \sum_{k=1}^{K} (k - SC_n)^2 A_n[k] \right) \Big/ \sum_{k=1}^{K} A_n[k]} \tag{10}$$

where $BW_n$ denotes the bandwidth and $SC_n$ is the spectral centroid as defined in (9).

In this work, all these spectrum-based features are grouped together to form a 10-dimensional spectral feature vector for audio classification task. To calculate the conventional spectral features, a 512-point FFT algorithm is used wherein the length of the analysis window is 30 ms and the overlap is 20 ms. For the spectral features based on the proposed FFT model, in the calculation of spectral rolloff point, spectral centroid and bandwidth, instead of the frequency indices $i$ in Table 1, the corresponding physical frequency values are used. As for the FFT model we proposed in [4], due to the use of a simple grouping scheme, it is not appropriate to extract frequency-dependent spectral features (e.g., spectral rolloff point, spectral centroid and bandwidth) based on that model.

## 4. SETUP OF CLASSIFICATION TESTS

### 4.1. Audio Sample Database

To carry out audio classification test, a generic audio database was built which includes speech, music and noise clips, sampled at the rate of 16 kHz. The audio classification decision is made on a one-second basis. Noise samples are selected from the NOISEX database which contains recordings of various noises. The total length of all the audio samples is 200 minutes. These samples are divided equally into two parts for training and testing respectively.

In the following, a clean test refers to a test wherein both the training set and testing set contain clean speech, clean music and noise. A test with a specific SNR value refers to a test wherein the training set contains clean speech, clean music and noise while the testing set contains noisy speech and noisy music (both with that specific SNR value), and noise.

### 4.2. Classification

A decision tree learning algorithm, i.e., C4.5 [6], is used for the classification. C4.5 is an algorithm for generating classification rules in the form of a decision tree based on a set of training examples. Due to the accuracy and speed, C4.5 is often taken as a reference for the development of other algorithms.

## 5. CLASSIFICATION TEST RESULTS

The test results (i.e., the error classification rates) are given in Table 2, where MFCC, SPEC-CON and SPEC-FFT represent the conventional MFCC features, the conventional spectral features and the spectral features based on the proposed FFT model respectively. Two equally-divided audio data sets (as mentioned in Section 4.1) are used for training and testing alternately, generating two classification error rates for each test case. By averaging over these two error rates, the average error rate corresponding to a specific test case is determined and the results are given in Table 2. Although the conventional MFCC and spectral features provide excellent performance in the clean case, their performance degrades rapidly as the SNR decreases, leading to a poor overall performance. On the other hand, the new spectral features based on the proposed FFT model are more robust in noisy test cases.

**Table 2**. Average error classification rates (%)

| SNR(dB) | $\infty$ | 20 | 15 | 10 | 5 |
|---|---|---|---|---|---|
| MFCC | 2.8 | 17.6 | 29.7 | 39.6 | 46.5 |
| SPEC-CON | 3.5 | 13.3 | 20.8 | 31.3 | 46.0 |
| SPEC-FFT | 2.9 | 4.0 | 6.8 | 13.0 | 29.4 |

## 6. CONCLUSIONS

In this paper, we have proposed an improved implementation for an auditory-inspired FFT-based model which calculates a noise-robust FFT spectrum. With the proposed improvements, the FFT model allows more flexibility in the extraction of audio features. A C4.5-based speech/music/noise classification task was conducted to evaluate the noise-robustness of the proposed FFT model. Compared to the conventional MFCC and spectral features, the new spectral features calculated using the proposed FFT model show more robust performance in noisy test cases.

## 7. REFERENCES

[1] N. Mesgarani, S. Shamma, and M. Slaney, “Speech discrimination based on multiscale spectro-temporal modulations,” in *Proc. ICASSP’04*, Montreal, Canada, May 2004, vol. 1, pp. 601–604.

[2] S. Ravindran and D. Anderson, “Low-power audio classification for ubiquitous sensor networks,” in *Proc. ICASSP’04*, Montreal, Canada, May 2004, vol. 4, pp. 337–340.

[3] W. Chu and B. Champagne, “A simplified early auditory model with application in speech/music classification,” in *Proc. CCECE’06*, Ottawa, Canada, May 2006, pp. 578–581.

[4] W. Chu and B. Champagne, “A noise-robust FFT-based spectrum for audio classification,” in *Proc. ICASSP’06*, Toulouse, France, May 2006, vol. 5, pp. 213–216.

[5] K. Wang and S. Shamma, “Self-normalization and noise-robustness in early auditory representations,” *IEEE Trans. Speech Audio Processing*, vol. 2, no. 3, pp. 421–435, July 1994.

[6] J. R. Quinlan, *C4.5:programs for machine learning*, Morgan Kaufmann, 1993.

[7] P.-W. Ru, *Perception-Based Multi-Resolution Auditory Processing of Acoustic Signals*, Ph.D. thesis, University of Maryland, 2000.

[8] Neural Systems Laboratory, University of Maryland, “NSL Matlab Toolbox,” http://www.isr.umd.edu/Labs/NSL/nsl.html.

[9] Douglas O’Shaughnessy, *Speech Communications–Human and Machines*, IEEE Press, second edition, 2000.

[10] M. Slaney, “Auditory toolbox: A matlab toolbox for auditory modeling work (version 2),” Tech. Rep. 1998-010, Interval Research Corporation, 1998. (See also: http://www.slaney.org/malcolm/pubs.html).

[11] E. Scheirer and M. Slaney, “Construction and evaluation of a robust multifeature speech/music discriminator,” in *Proc. ICASSP’97*, Munich, Germany, April 1997, vol. 2, pp. 1331–1334.

[12] C. Xu, N. C. Maddage, and X. Shao, “Automatic music classification and summarization,” *IEEE Trans. Speech Audio Processing*, vol. 13, no. 3, pp. 441–450, May 2005.

[13] E. Wold, T. Blum, D. Keislar, and J. Wheaten, “Content-based classification, search, and retrieval of audio,” *IEEE Multimedia*, vol. 3, no. 3, pp. 27–36, Fall 1996.

[14] S. Z. Li, “Content-based audio classification and retrieval using the nearest feature line method,” *IEEE Trans. Speech Audio Processing*, vol. 8, no. 5, pp. 619 – 625, Sept. 2000.