Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 205369, 9 pages
http://dx.doi.org/10.1155/2015/205369

Research Article

Guitar Chords Classification Using Uncertainty Measurements of Frequency Bins

Jesus Guerrero-Turrubiates,[1] Sergio Ledesma,[1] Sheila Gonzalez-Reyna,[2] and Gabriel Avina-Cervantes[1]

[1]*Division de Ingenierias, Universidad de Guanajuato, Campus Irapuato-Salamanca, Carretera Salamanca-Valle de Santiago km 3.5 + 1.8 km, Comunidad de Palo Blanco, 36885 Salamanca, GTO, Mexico*
[2]*Universidad Politecnica de Juventino Rosas, Hidalgo 102, Comunidad de Valencia, 38253 Santa Cruz de Juventino Rosas, GTO, Mexico*

Correspondence should be addressed to Jesus Guerrero-Turrubiates; jdj.guerreroturrubiates@ugto.mx

Received 28 May 2015; Accepted 2 September 2015

Academic Editor: Matteo Gaeta

This paper presents a method to perform chord classification from recorded audio. The signal harmonics are obtained by using the Fast Fourier Transform, and timbral information is suppressed by spectral whitening. A multiple fundamental frequency estimation of whitened data is achieved by adding attenuated harmonics by a weighting function. This paper proposes a method that performs feature selection by using a thresholding of the uncertainty of all frequency bins. Those measurements under the threshold are removed from the signal in the frequency domain. This allows a reduction of 95.53% of the signal characteristics, and the other 4.47% of frequency bins are used as enhanced information for the classifier. An Artificial Neural Network was utilized to classify four types of chords: major, minor, major 7th, and minor 7th. Those, played in the twelve musical notes, give a total of 48 different chords. Two reference methods (based on Hidden Markov Models) were compared with the method proposed in this paper by having the same database for the evaluation test. In most of the performed tests, the proposed method achieved a reasonably high performance, with an accuracy of 93%.

1. Introduction

A chord, by definition, is a harmonic set of two or more musical notes that are heard as if they was simultaneously sounding [1]. A musical note refers to the pitch class set of C, $C\sharp/D\flat$, D, $D\sharp/E\flat$, E, F, $F\sharp/G\flat$, G, $G\sharp/A\flat$, A, $A\sharp/B\flat$, B, and the intervals between notes are known as half-note interval or semitone interval. Thus, chords can be seen as musical features and they are the principal harmonic content that describes a musical piece [2, 3].

A chord has a basic construction known as triad that includes notes identified as a fundamental (the root), a third, and a fifth [4]. The root can be any note chosen from the pitch class set, and it is used as the first note to construct the chord; besides, this note gives the name to the chord. The third has the function of making the chord be minor or major. For a minor chord the third is located at 3 half-notes intervals from the root. On the other hand, a major chord has the third placed at 4 half-note intervals from the root. The perfect fifth, which completes the triad, is located at 7 half-note intervals from the root. If a note is added to the triad at 11 half-note intervals from the root, then the chord will become a 7th chord. For instance, a C major chord (CMaj) will be composed of a root C note, a major third E note, and a perfect fifth G note; the C major with a 7th (CMaj7) is composed of the same triad of C major plus the 7th note B.

Chord arrangements, melody and lyrics, can be grouped in written summaries known as lead sheets [5]. All kind of musicians, from professionals to amateur, make use of these sheets because they provide additional information about when and how to play the chords or some other arrangement on a melody.

Writing lead sheets of chords by hand is a task known as chord transcription. It can only be performed by an expert; however this is a time-consuming and expensive process. In engineering, the automatization of chord transcription has been considered a high-level task and has some applications such as key detection [6, 7], cover song identification [8], and audio-to-lyrics alignment [9].

Chord transcription requires recognizing or estimating the chord from an audio file by applying some signal preprocessing. The most common method for chord recognition is based on templates [10, 11]; in this case a template is a vector of numbers. Then, this method suggests that only chord definition is necessary to achieve recognition. The simplest chord template has a binary structure, for this kind of template, the notes that belong to the chord will have unit amplitude, and the remaining ones will have null amplitude. This template is described by a 12-dimensional vector; each number in the vector represents a semitone in the chromatic scale or pitch class set. As an illustration, the C major chord template will be [1 0 0 0 1 0 0 1 0 0 0 0]. The 12-dimensional vectors obtained from an audio frame signal are known as *chroma vectors*, and they were proposed by Fujishima [12] for chord recognition using templates. In his work, chroma vectors are obtained from the Discrete Fourier Transform (DFT) of the input signal. Fujishima's method (Pitch Class Profile, PCP) is based on an intensity map on the Simple Auditory Model (SAM) of Leman [13]. This allows chroma vector to be formed by the energy of the twelve semitones of the chromatic scale. In order to perform chord recognition, two matching methods were tested: the Nearest Neighbors [14] (Euclidean distances between the template vectors and the chroma vectors) and the Weighted Sum Method (dot product between chroma vectors and templates).

Lee [11] applied the Harmonic Product Spectrum (HPS) [15] to propose the Enhanced Pitch Class Profile (EPCP). In his study, chord recognition is performed by maximizing the correlation between chord templates and chroma vectors.

Template matching models have poor recognition performance on real life songs, because chords change with time, and consequently chroma vectors will have semitones of two different chords. Therefore, statistical models became popular methods for chord recognition [16–18]. Thus, Hidden Markov Models [19, 20] (HMM) are probabilistic models for a sequence of observed variables assumed to be independent of each other, and it is supposed that there is a sequence of hidden variables that are related with the observed variables.

Barbancho et al. [21] proposed a method using HMM to perform a transcription of guitar chords. The chord types used in their study are major, minor, major 7th, and minor 7th of each root of the pitch class set. That is a total of 48 chord types. All of them can be played in many different forms; thus, to play the same chord several finger positions can be used. In their work, 330 different forms for 48 chord types are proposed (for details see the reference); in this case every single form is a hidden state. Feature extraction is achieved by the algorithm presented by Klapuri [22], and a model that constrains the transitions between consecutive forms is proposed. Additionally, a cost function that measures the physical difficulty of moving from one chord form to another one is developed. Their method was evaluated using recordings from three musical instruments: an acoustic guitar, an electric guitar, and a Spanish guitar.

Ryynänen and Klapuri [23] proposed a method using HMM to perform melody transcription and classification of bass line and chords in polyphonic music. In this case, fundamental frequencies (F_0's) are found using the estimator in [21]; after that, these are passed through a PCP algorithm in order to enhance them. A HMM of 24 states (12 states for major chords and 12 states for minor ones) is defined. The transition probabilities between states are found using the Viterbi algorithm [24]. The method does not detect silent segments; however, it provides chord labeling for each analyzed frame.

The aforementioned methods achieve low accuracies, and the most recent cited one, the method from Barbancho et al., achieves high accuracy by combining probabilistic models. However, the uses of a HMM and the probabilistic models in their work make such method somewhat complex.

In this paper, we propose a method based on Artificial Neural Networks (ANNs) to classify chords from recorded audio. This method classifies chords from any octave for a six-string standard guitar. The chord types are major, minor, major 7th, and minor 7th, that is, the same variants for the chords used by Barbancho et al. [21]. First, time signals are converted to the frequency domain, and timbral information is suppressed by spectral whitening. For feature selection, we propose an algorithm that measures the uncertainty for the frequency bins. This allows reducing the dimensionality of the input signal and enhances the relevant components to improve the accuracy of the classifier. Finally, the extracted information is sent to an ANN to be classified. Our method avoids the calculation of transition probabilities and probabilistic models working in combination; nevertheless the accuracy achieved in this study has superior performance over the most mentioned methods.

The rest of this paper is organized as follows. In Section 2, fundamental concepts related to this study are presented. Section 3 details the theoretical aspects of the proposed method. Section 4 presents experimental results that validate the proposed method, and Section 5 includes our conclusions and directions for future work.

2. General Concepts

For clarity purposes, this section presents two important concepts widely used in Digital Signal Processing (DSP). These concepts are the Fourier Transform and spectral whitening.

2.1. Transformation to Frequency Domain. The human hearing system is capable of performing a transformation from the time domain to the frequency domain. There is evidence that humans are more sensitive to magnitude than phase information [25]; as a consequence humans can perceive harmonic information. This is the main idea to perform the classification of guitar audio signals in this work. Therefore, a frequency domain representation of the original signal has to be calculated.

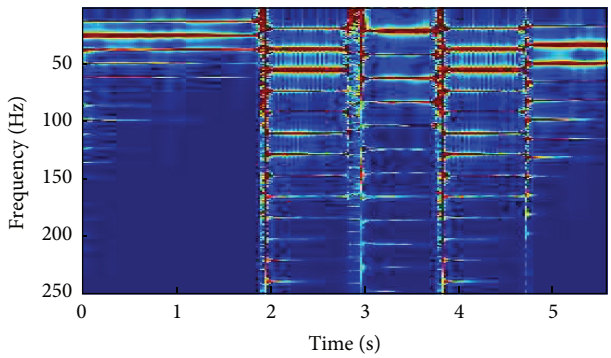

Figure 1: Example of a spectrogram.

The time to frequency domain transformation is obtained by applying the Fast Fourier Transform (FFT) to the input signal $x[n]$ and is represented by

$$X = \mathcal{F}\{x[n]\}. \tag{1}$$

Equation (1) describes the transformation of $x[n]$ at all times. However, this is not convenient because songs or signals, in general, are not stationary. For this reason, a window function, $w[n]$, is applied to the time signal as

$$z[n] = x[n]\, w[n], \tag{2}$$

where $w[n]$, for this study, is the Hamming window function according to

$$w[n] = \varphi - (1-\varphi)\cos\left(\frac{2\pi n}{N-1}\right), \tag{3}$$

where $\varphi = 0.54$, $n = [0, N-1]$, and N is the number of samples in the frame analysis. A study about the use of different window types can be found in Harris [26]. Equations (2) and (3) divide the signal in different frames that allowing the analysis of the signal in the frequency domain by

$$X_w = \mathcal{F}\{z[n]\}. \tag{4}$$

For this work, windowing functions will have 50% of overlapping to analyze the entire signal and thus obtain a set of frames $z_i[n]$ (for simplicity in the notation z_i will be used). Those frames can be concatenated to construct a matrix $\mathbf{Z} = [z_1\ z_2\ \cdots\ z_i]$, and, then, compute the FFT for every column. The result is a representation in the frequency domain as in Figure 1; this representation is known as spectrogram [27]. This is the format that the signals will be presented to the classifier for training.

2.2. Spectral Whitening. This process allows obtaining a uniform spectrum of the input signal, and it is achieved by boosting the frequency bins of the FFT. There exist different methods to perform spectral whitening [28–31].

Thus, inverse filtering [22] is the whitening method used in our experiments, and it is described next.

First, the original windowed signal is zero-padded to twice its length as

$$y_i = [z_i\ \ 0\ \ 0\ \cdots\ 0]^{\mathsf{T}}, \tag{5}$$

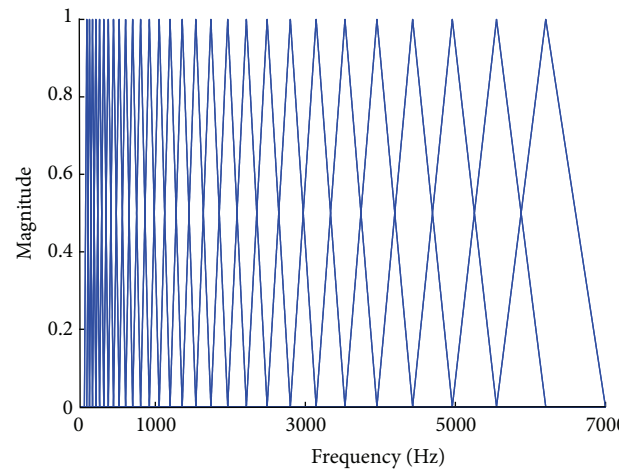

Figure 2: Responses $H_b(k)$ applied in spectral whitening.

and its FFT, represented by $\mathbf{\Gamma}_i$, is calculated. The resulting frequency spectrum will have an improved amplitude estimation because of the zero-padding. Next, a filter bank is applied to $\mathbf{\Gamma}_i$; the central frequencies of this bank are given by

$$c_b = 229\left(10^{(b+1)/21.4} - 1\right), \tag{6}$$

where $b = 0, \ldots, 30$. In this case, each filter in the bank has a triangular response H_b; in fact, this bank tries to simulate the inner ear basilar membrane. The band-pass frequencies for each filter are from c_{b-1} to c_{b+1}. Because there is no more relevant information at higher frequencies than 7000 Hz, the maximum value for the parameter b was 30.

Subsequently, the standard deviations σ_b are calculated as

$$\sigma_b = \left(\frac{1}{K}\sum_k H_b(k)\left|\mathbf{\Gamma}_i(k)\right|^2\right)^{1/2} \quad \text{for } k = 0, 1, \ldots, K-1, \tag{7}$$

where uppercase K is the length of the FFT series.

Later on, the compression coefficients for the central frequencies $c_1, c_2, \ldots, c_b$ are calculated as $\gamma_b = \sigma_b^{\nu-1}$, where $\nu = [0, 1]$ is the amount of spectral whitening applied to the signal. The coefficients γ_b are those that belong to the frequency bin of the "peak" of each triangle response; observe Figure 2. The rest of the coefficients $\gamma(k)$ for the remaining frequency bins are obtained performing a linear interpolation between the central frequency coefficients γ_b.

Finally, the white spectrum is obtained with a pointwise multiplication of all compression coefficients with $\mathbf{\Gamma}_i$ as

$$\mathbf{I}_i(k) = \gamma(k)\,\mathbf{\Gamma}_i(k). \tag{8}$$

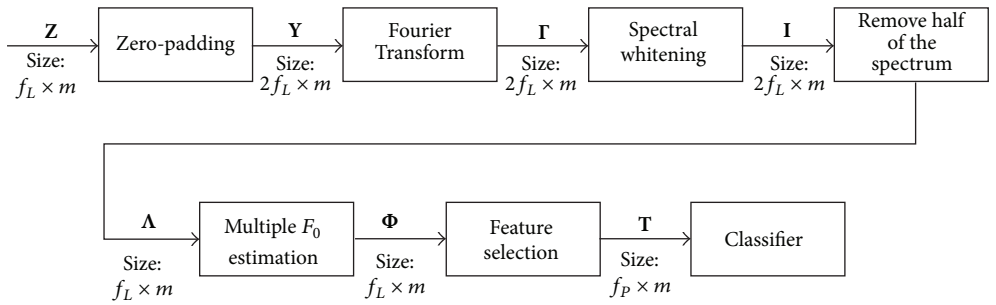

Figure 3: Overview of the proposed system for training with f_L frequency bins and m samples of audio.

3. Proposed Method

Our proposed method is described in the block diagram shown in Figure 3. The method begins by defining the columns of matrix $\mathbf{Z}$ as

$$\mathbf{Z} = \begin{bmatrix} z_1[0] & z_2[0] & \cdots & z_m[0] \\ z_1[1] & z_2[1] & \cdots & z_m[1] \\ \vdots & \vdots & \ddots & \vdots \\ z_1[f_L] & z_2[f_L] & \cdots & z_m[f_L] \end{bmatrix}, \tag{9}$$

where a single column vector $[z_m[0] \;\; z_m[1] \;\; \cdots \;\; z_m[f_L]]^\intercal$ represents the mth Hamming windowed audio sample. These columns are zero-padding to twice their length f_L as

$$\mathbf{Y} = [\mathbf{Z} \mid \mathbf{0}]^\intercal = \begin{bmatrix} y_1 & y_2 & \cdots & y_m \end{bmatrix}, \tag{10}$$

where $\mathbf{0}$ is a zero matrix of the same size of $\mathbf{Z}$. Then, (10) indicates an augmented matrix.

After that, the signal spectrum for every column of $\mathbf{Y}$ is calculated by applying the FFT, and then these columns are passed through a spectral whitening step and the output matrix is represented as $\mathbf{I}$. Furthermore, by taking advantage of the symmetrical shape of the FFT, only the first half of the frequency spectrum (represented by $\mathbf{\Lambda}$) is taken in order to perform the analysis.

A multiple fundamental frequency estimation algorithm and a weighting function are applied to the whitened audio signals. These algorithms enhance the fundamental frequencies by adding their harmonics attenuated by the weighting function. The output matrix of this step is denoted as $\mathbf{\Phi}$.

The training set includes all data in a matrix of f_L frequency bins and m audio samples, where each row or frequency bin will be an input to the classifier. The number of inputs can be reduced from f_L to f_P ($\mathbf{T}$ matrix) by applying a method based on the uncertainty of the frequency bins, thus enhancing the pertinent information to perform a classification. Finally, enhanced data are used to train the classifier and then to validate its performance.

3.1. Multiple Fundamental Frequency Estimation. The fundamental frequencies of the semitones in the guitar are defined by

$$f_j = 2^{j/12} f_{\min}, \tag{11}$$

where $j \in \mathbb{Z}$ and $f_{\min}$ is the minimum frequency to be known; for example, in a standard six-string guitar, the lowest note is E having a frequency of 82 Hz.

Signal theory establishes that the harmonic partials (or just harmonics) of a fundamental frequency are defined by

$$f_{h_r} = h_r f_j, \tag{12}$$

where $h_r = 2, 3, 4, \ldots, M+1$. In this study M represents the number of harmonics to be considered. As an illustration, for a fundamental frequency $f_j = 131$ Hz of a C note, the first three harmonics will be the set $\{262, 393, 524\}$.

In this work, if a frequency is located at ±3% of the semitones frequencies, then this frequency is considered to be correct. This approach was proposed in [22].

In an mth frame under analysis, fundamental frequencies can be raised if harmonics are added to its fundamentals [22], by applying

$$\mathbf{\Lambda}\left(f_j, m\right) = \mathbf{\Lambda}\left(f_j, m\right) + \sum_{h_r=2}^{M+1} \mathbf{\Lambda}\left(h_r f_j, m\right), \tag{13}$$

and, then, all harmonics $\mathbf{\Lambda}(h_r f_j, m)$ and their fundamental frequencies $\mathbf{\Lambda}(f_j, m)$, described in (13), are removed from the frequency spectrum. When the resulting signal is again analyzed, with the described method, a different fundamental frequency will be raised.

A common issue with (13) is when two or more fundamentals share a same harmonic. For instance, the fundamental frequency of 65.5 Hz of C note has a harmonic located at 196.5 Hz. When the Euclidean distances [32] between the analysis frequency and the frequencies of the semitones are computed, the minimum distance or nearest frequency will correspond to the G note. This implies that if those two notes are present in the same analysis frame, then the harmonic of G will be summed and eliminated with the harmonics of the C note. This is because the 196 Hz harmonic is located in the range of ±3% of the frequency of a G note.

There are some methods that deal with this problem. In [33], a technique that makes use of a moving average filter is proposed. In that work, the fundamental frequency takes its original amplitude and a moving average filter modifies the amplitude of its harmonics. Then, only part of their amplitude is removed from the original frequency spectrum.

In [22], a weighting function that modifies the amplitude of the harmonics is proposed. Also, an algorithm to find

multiple fundamental frequencies is suggested. The weighting function is given by

$$g_{\tau,h_r} = \frac{f_s/\tau_{\max} + \alpha}{h_r f_s/\tau + \beta}, \tag{14}$$

where $f_s/\tau_{\max}$ represents the low limit frequency (e.g., 82 Hz), f_s/τ is the fundamental frequency f_j under analysis, and f_s is the sampling frequency. The parameters α and β are used to optimize the function and minimize the amplitude estimation error (see [22] for details). In the work [22], the analyzed f_j in a whitened signal $\mathbf{\Lambda}(k,m)$ is used to find its harmonics with

$$\widehat{s}(\tau) = \sum_{h_r=1}^{M} g(\tau, c) \max_q |\mathbf{\Lambda}(k,m)|, \tag{15}$$

where q is a range of frequency bins in the vicinity of f_j analyzed. The parameter q indicates that the signal spectrum is divided into analysis blocks, to find the fundamental frequencies. Thus, $\widehat{s}(\tau)$ becomes a linear function of the magnitude spectrum $\mathbf{\Lambda}(k,m)$. Then, a residual spectrum $\mathbf{\Lambda}_R(k,m)$ is initialized to $\mathbf{\Lambda}(k,m)$, and a fundamental period τ is estimated using $\mathbf{\Lambda}_R(k,m)$. The harmonics of τ are found in $h_r f_s/\tau$, and then they are added to a vector $\mathbf{\Lambda}_D(k,m)$ in their corresponding position of the spectrum. The new residual spectrum is calculated as

$$\mathbf{\Lambda}_R(k,m) \longleftarrow \max\left(0, \mathbf{\Lambda}_R(k,m) - d\mathbf{\Lambda}_D(k,m)\right), \tag{16}$$

where $d = [0,1]$ is the amount of subtraction. This process iteratively computes a different fundamental frequency using the methodology described above. The algorithm finishes until there are no more harmonics in $\mathbf{\Lambda}_R(k,m)$ to be analyzed. Equation (15) was adapted to keep the notation of our work; refer to [22] for further analysis.

In this study, we propose a modification of Klapuri's algorithm, in an attempt to achieve a better estimate of the multiple fundamental frequencies. Using (14) and the mth whitened signal $\mathbf{\Lambda}(k,m)$, the multiple fundamental frequencies can be found by using

$$\mathbf{\Phi}(k,m) = d^k \sum_{h_r} g(\tau, h_r) \left|\mathbf{\Lambda}(h_r k, m)\right|, \tag{17}$$

where $h_r = \{n \mid n \in \mathbb{Z}, n > 1, nk < K/2\}$ for $k = 0, 1, \ldots, K/2$. Equation (17) analyzes all frequency bins and its harmonics in the signal spectrum. This equation adds to the kth frequency bin, all its harmonics in $h_r k$ of the entire spectrum. Besides, the weighting function performs an estimation of the harmonic amplitude that must be added to the kth frequency bin. Observe that the weighting function does not modify the original amplitude of the harmonics.

Finally when all frequency bins have been analyzed, the resulting signal has all its fundamental frequencies with high amplitude. This will help the classifier to have an accurate performance.

3.2. Feature Selection. The objective of this paper is to classify frequencies. Then, the inputs of the classifier are all frequency bins that come from the FFT. However, not all frequency bins will have relevant information. Therefore, a method to remove unnecessary data and enhance the relevant data has to be performed. This will result in a reduction of the number of inputs to the classifier.

We propose a method based on the uncertainty of the frequency bins. This method will discriminate all those that are not relevant for the classifier in order to improve its performance.

In Wei [34], it is stated that, similarly to the entropy, the variance can be considered as a measure of uncertainty of a random variable, if and only if the distribution has one central tendency. The histograms for all frequency bins of the 48 chord types were calculated. This can be used to verify whether the distribution could be approximated to any distribution with only one central tendency. For simplicity, Figure 4 represents one frequency bin distribution of a C major and a C minor chord, respectively; it can be seen that the distribution fits into a Gaussian distribution. This same behavior was observed in the other samples of the 48 different chords. This demonstrates that the variance can be used in this study as an uncertainty measure in the frequency bins.

In order to perform the feature selection using the uncertainty of the frequency bins, first consider a matrix $\mathbf{\Phi}$ defined by

$$\mathbf{\Phi} = \begin{bmatrix} \overrightarrow{a_1} \\ \overrightarrow{a_2} \\ \vdots \\ \overrightarrow{a_f} \end{bmatrix}, \tag{18}$$

where $\overrightarrow{a_f}$ is a vector formed by the magnitudes of the fth-component frequency bin of all audio samples. The variances of each $\overrightarrow{a_f}$ can be computed with

$$\sigma_f^2 = \frac{1}{m} \sum_{q=1}^{m} \left(\overrightarrow{a_f}(q) - \mu\right)^2, \quad \text{for } f = 1, 2, \ldots, f_L, \tag{19}$$

where

$$\mu = E\left\{\overrightarrow{a_f}\right\}. \tag{20}$$

If $\sigma_f^2 \approx 0$, then it means that for that particular frequency bin the input is quasi-constant; consequently this frequency bin can be eliminated from all audio samples. This can be achieved if we consider

$$\sigma_{\max}^2 = \max_f \left\{\vec{\sigma}_f^2\right\}, \tag{21}$$

and a vector $\vec{v}_{\text{ind}}$ formed with the indexes f of $\vec{\sigma}_f^2$ that are defined by

$$\vec{v}_{\text{ind}} = \left\{\vec{\sigma}_f^2 \mid \left(\vec{\sigma}_f^2 \geq \xi \sigma_{\max}^2\right)\right\}, \quad \text{where } 0 \leq \xi \leq 1. \tag{22}$$

Once feature selection has been performed, the remaining frequency bins will form the input to the classifier.

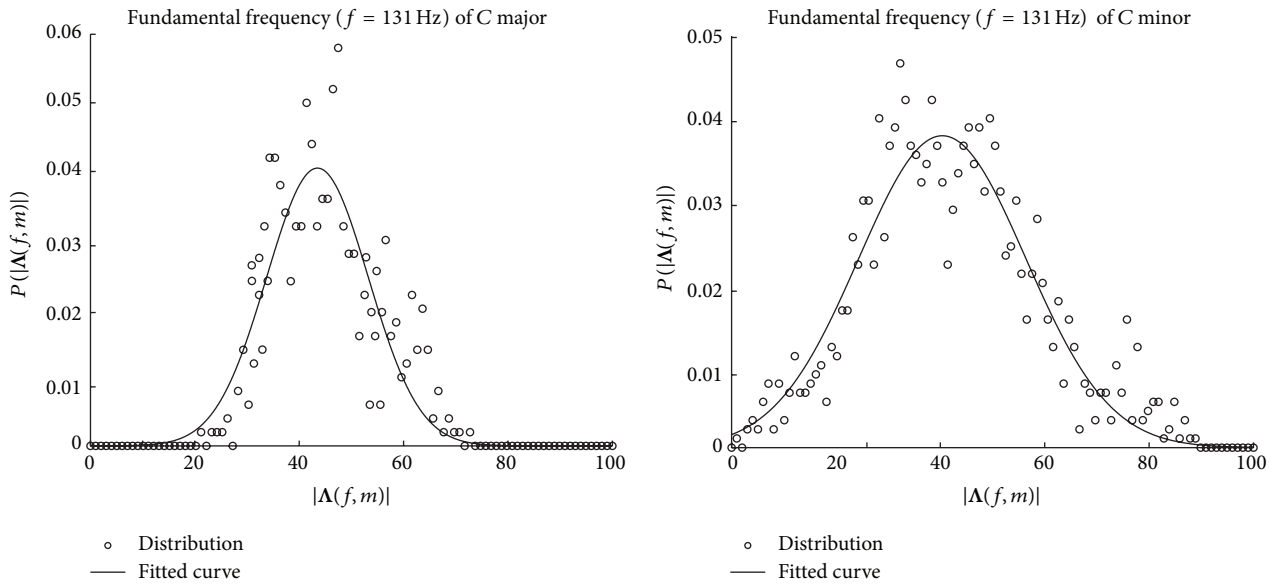

Figure 4: Central tendency of the fundamental frequency of a C chord.

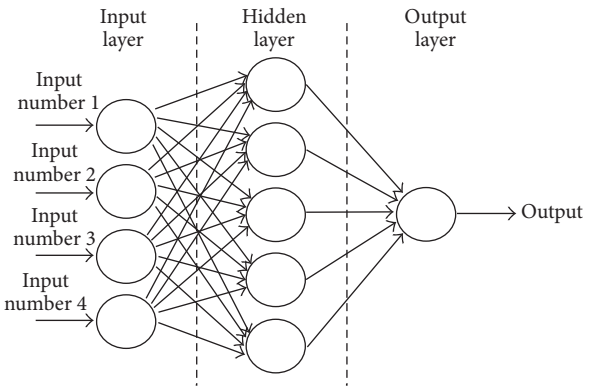

Figure 5: Multilayer perceptron.

3.3. Classifier. Classification is an important part for chord transcription. In order to perform a good classification, important data will be generated from the original information. Then, a classification algorithm will be able to label the chords. Artificial Neural Networks [35] (ANNs) can be considered as "massively parallel computing systems consisting of an extremely large number of simple processors with many interconnections"; according to Jain et al. [36] ANNs have been used in chord recognition as a preprocessing method or as a classification method. Gagnon et al. [37] proposed a method with ANN to preclassify the number of strings plucked in a chord. Humphrey and Bello [38] used labeled data to train a convolutional neural network. In this study, an Artificial Neural Network was used to perform classification. Figure 5 represents the configuration for the ANN used in this work. The ANN was trained using the Back Propagation algorithm [39].

4. Experimental Results

Computer simulations were performed to quantitatively evaluate the proposed method. The performance of two state-of-the-art references [21, 23] was compared with the present work.

Databases for training and testing containing four chord types (major, minor, major 7th, and minor 7th) with different versions of the same chord are considered. Electric and acoustic guitar recordings were used to construct the training data set. A total of 25 minutes were recorded from an electric guitar, and a total of 30 minutes were recorded from an acoustic guitar. Recordings include sets of chords played consecutively (e.g., C-$C\sharp$-D-$D\sharp$...), as well as some parts of songs. The database used for evaluation was provided by Barbancho et al. [21]. This database has 14 recordings: 11 recordings from two different Spanish guitars played by two different guitar players, 2 recordings from an electric guitar, and 1 recording from an acoustic guitar, making a total duration of 21 minutes and 50 seconds. The sampling frequency f_s is of 44100 Hz for all audio recordings.

The training data set was divided into frames of 93 ms, leading to a FFT of 4096 frequency bins. In the spectral whitening, the signal was zero-padded to twice its length before applying the frequency domain transform, so a FFT of 8192 data was obtained. For the spectral whitening, the K parameter takes the original length of the FFT but the length of the whitened signals remains at 4096 frequency bins. For the multiple fundamental frequency estimation, the α and β parameters are constant and set to 52 and 320, respectively, as in Klapuri [22], while the parameter d was adjusted to improve performance. An optimum value of 0.99 was found. This parameter differs from the value in [22] because, in our method, the signal is modified in every cycle that h_r in (15) increases; on the other hand, Klapuri [22] modifies the signal after h_r increases to its higher value.

Table 1: Frequency variances and threshold $\xi\sigma_{max} = 0.050$.

Variance	0.012	**0.052**	⋯	0.037	**0.055**	⋯	0.048	**0.060**	0.010
Frequency bin	130	**131**	⋯	163	**164**	⋯	195	**196**	197

Table 2: Classifier inputs with threshold $\xi\sigma_{max} = 0.050$.

Classifier input	⋯	j_{th}	$j_{th} + 1$	$j_{th} + 2$	⋯
Frequency bin	⋯	131	164	196	⋯

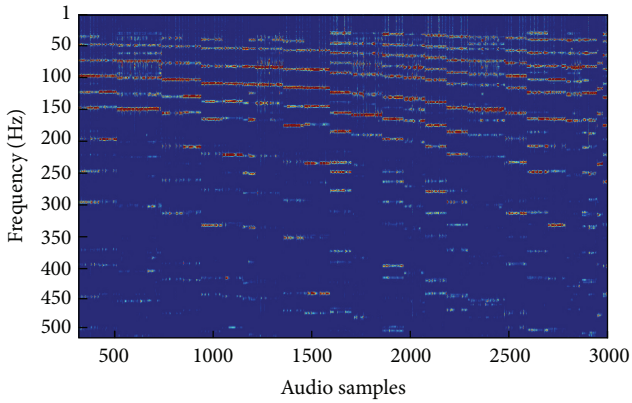

Figure 6: Training set before feature extraction.

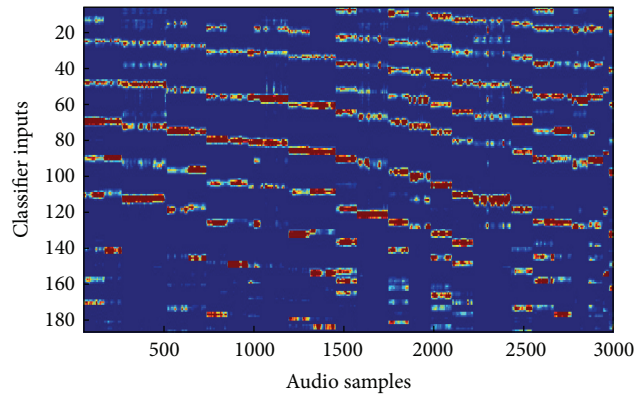

Figure 7: Training set after feature extraction.

Table 3: Comparison between methods.

	PM	**95%**
Reference method [21] [48 possibilities: major, minor, major 7th, and minor 7th]	PHY	83%
	MUS	86%
	PC	75%
Reference method [23] [24 (major/minor) and 48 (major, minor, major 7th, and minor 7th) possibilities evaluated separately]	MM	91%
	MMC	80%
	CC	70%
Proposed method (48 possibilities: major, minor, major 7th, and minor 7th)	**VTH**	**93%**

These processes were applied to all audio samples to build a training data set. In this case, the data set is a matrix of 4096 rows (frequency bins) by 5000 columns (audio samples). In (21), the maximum variance for all frequency bins in the audio samples is computed. Equation (22) proposes a threshold to remove all those frequency bins that remain quasi-constant. For instance, suppose that a threshold of 0.05 is set, and some frequency bins variances (shown in Table 1) are evaluated. Only those above the threshold will be taken as inputs to the classifier, as is shown in Table 2.

Performance tests were made to find the optimal value for ξ. This parameter was varied; then the ANN was trained and evaluated. The process was repeated until the best result was obtained. The ξ parameter was found to be optimal at 0.01326. This allows a 95.6% reduction of the total of the frequency bins, while keeping the relevant information. Therefore, we concluded that, for a ξ value lower than 0.01326, some information required for a correct classification is lost. Figure 6 shows part of the training data set, in fact only frequency bins in the range $[0, 500]$, and 3000 audio samples are depicted. Figure 7 shows the same data set of Figure 6 after the feature extraction algorithm was applied. It can be observed that the algorithm maintains sufficient information to train the classifier.

An ANN was used as a classification method with 183 inputs and 48 outputs. The applied performance metric was the ratio of the number of correctly classified chords to the total number of frames analyzed.

The validation test had the same structure as the one presented in Figure 3. First, audio data was loaded. Second, a frequency domain transformation and a spectral whitening are applied to the signal. Finally, the multiple fundamental frequency estimation algorithm is used. At this point, the signal has 4096 frequency bins. To reduce the number of frequency bins, only those that meet (22) are taken from the signal and then passed through the classifier.

The results of the proposed method *VTH (Variance Threshold)* in this work were compared with two state-of-the-art methods. The best are shown in Table 3; specifically 48 chord types with different variants of the same chord were evaluated. For reference method proposed by Barbancho et al. [21], experiments with different algorithms were performed. This method is denoted by *PM* and includes all models described next. The *PHY* model describes the probability of the physical transition between chords. These probabilities are computed by measuring the cost of moving the fingers from one position to another. The *MUS* model is based on musical transition probabilities, that is, the probabilities of switching between chords. These were estimated from the first eight albums of The Beatles. And, the *PC* model is equal to the proposed method but without the transition probabilities; instead, uniform transition probabilities are used. All models were separately tested; an accuracy of 86%

was achieved at most. The best result was obtained from using the combination of all methods; a 95% accuracy was achieved in this case.

For the reference method proposed by Ryynänen and Klapuri [23], the evaluation results were taken from [21]; in this case, three tests were performed. First, *MM* tests (only major and minor chords) were carried on; for all three tests, this was the one with the highest accuracy (91%). Second, *MMC* tests were executed, all chords were taken into account; however 7th major/minor chords labeled as major/minor were correctly classified; that is, a *C*Maj7 labeled as *C*Maj was correct. Finally, *CC* tests were set with the 48 possibilities; that is, 7th major/minor chords labeled as major/minor were incorrect; this results in an accuracy of 70%.

The proposed method on this paper achieves an accuracy of 93% in the evaluation test. This classification performance was achieved with a 95% confidence interval of [91.4, 94.6]. The results are competitive with the two reference methods. Even though Barbancho et al. [21] have a 95% of accuracy, it is only achieved when all algorithms *PHY*, *MUS*, and *PC* are combined. Besides, HMM needs the calculations of probability transitions between the states of the model (48 chord types). This makes their method more complex than the one presented in this work. This paper focuses only on chord recognition, so the comparison with [21] does not take into consideration the finger configuration.

5. Conclusions

A method to classify chords of an audio signal is proposed in this work. This is based on a frequency domain transformation, where harmonics are the key to find the fundamental frequencies that compose the input signal. It was found that all remaining frequency bins after feature extraction were in the range from 40 Hz to 800 Hz. This means that the relevant information for the classifier is located on the low frequency end.

The chords considered were major, minor, major 7th, and minor 7th. Two state-of-the-art methods, which used the same chords, were taken to compare our study. All computer simulations were performed using the same database. The reference method from Ryynänen and Klapuri [23] had the best performance when only 24 chord types were considered. Our method outperforms the method of Ryynänen and Klapuri by 2%, even when, in our work, 48 chord types were classified. The reference method of Barbancho et al. [21] had an accuracy of 95%; however, they performed a signal analysis to propose two statistical models and a third one that does not consider probability transitions between states. Their best performance is achieved with all models working together; if they are separately tested, the performance is at most 86%. Also, their classification method is based on a Hidden Markov Model that needs interconnected states.

The method presented in this work avoids designing statistical models and interconnected states for the HMM. The Artificial Neural Network as a classification method works with a high precision when the data presented have been processed with an appropriate algorithm. The proposed method for feature selection achieves high accuracy, because the data presented to the classifier have the pertinent information to be trained.

The sampling frequency of 44100 Hz and the windowing of 4096 data result in a frequency resolution of 10 Hz. With this frequency resolution it is not possible to distinguish the low frequencies of the guitar, for example, an *E* with 82 Hz and an *F* with 87 Hz. However, the original signal has six sources (strings), where three of them are octaves from the other three (except for 7th chords). Then, because the proposed method for multiple fundamental frequency estimation adds the harmonics for every single *k*th bin, the high octaves can be raised. For example, for an *E* of 82 Hz, the octave at 164 Hz will also be raised. Then, this octave with the other fundamentals gives a correct classification of the chord. In the case of an *F*, the fundamental at 87 Hz can not be distinguished from the frequency of 82 Hz. Nevertheless, the octave at 174 Hz will be perfectly raised; so with the other fundamentals frequencies of *F*, the ANN performs a correct classification.

The present work due to its simplicity can be applied to chord recognition in some devices, for example, a Field-Programmable Gate Array (FPGA) or some microcontrollers. This study leaves for a future work the source separation of each string in the guitar. Once a played chord is known, we can make some assumptions about where the hand playing the chord is. Thus, we can apply some methods of blind source separation to obtain the audio of each guitar string. Besides, with the information of separated strings, the classifier can be extended for a wide set of chord families. Because the classification can be performed by a single string instead of the mixture of six strings, this can lead to the complete transcription of guitar chords and identification of strings being played.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research has been supported by the "National Council on Science and Technology" of Mexico (CONACYT) under Grant no. 429450/265881 and by Universidad de Guanajuato through DAIP. The authors would like to thank A. M. Barbancho et al., for providing the database used for comparison.

References

[1] O. Karolyi, *Introducing Music*, Penguin Books, 1965.

[2] Hal Leonard, *The Real Book*, Hal Leonard, Milwaukee, Wis, USA, 2004.

[3] J. Brent and S. Barkley, *Modalogy: Scales, Modes and Chords*, Hal Leonard Corporation, 2011.

[4] D. Latarski, *An Introduction to Chord Theory*, DoLa Publisher, 1982.

[5] J. Weil, T. Sikora, J. Durrieu, and G. Richard, "Automatic generation of lead sheets from polyphonic music signals," in *Proceedings of the 10th International Society for Music Information Retrieval Conference*, pp. 603–608, 2009.

[6] A. Shenoy and Y. Wang, "Key, chord, and rhythm tracking of popular music recordings," *Computer Music Journal*, vol. 29, no. 3, pp. 75–86, 2005.

[7] K. Lee and M. Slaney, "Acoustic chord transcription and key extraction from audio using Key-dependent HMMs trained on synthesized audio," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 16, no. 2, pp. 291–301, 2008.

[8] D. P. W. Ellis and G. E. Poliner, "Identifying "cover songs" with chroma features and dynamic programming beat tracking," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '07)*, pp. IV1429–IV1432, Honolulu, Hawaii, USA, April 2007.

[9] M. Mauch, H. Fujihara, and M. Goto, "Integrating additional chord information into HMM-based lyrics-to-audio alignment," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, no. 1, pp. 200–210, 2012.

[10] C. Harte and M. Sandler, "Automatic chord identification using quantized chromagram," in *Proceedings of the Audio Engineering Society Convention*, pp. 28–31, 2005.

[11] K. Lee, "Automatic chord recognition from audio using enhanced pitch class profile," in *Proceedings of the International Computer Music Conference (ICMC '06)*, New Orleans, La, USA, 2006.

[12] T. Fujishima, "Real time chord recognition of musical sound: a system using common lisp music," in *Proceedings of the International Computer Music Conference (ICMC '99)*, pp. 464–467, 1999.

[13] M. Leman, *Music and Schema Theory*, Springer, 1995.

[14] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[15] M. R. Schroeder, "Period histogram and product spectrum: new methods for fundamental-frequency measurement," *The Journal of the Acoustical Society of America*, vol. 43, no. 4, pp. 829–834, 1968.

[16] A. Sheh and D. Ellis, "Chord segmentation and recognition using EM-trained hidden Markov models," in *Proceedings of the 4th International Society for Music Information Retrieval Conference*, pp. 183–189, Taipei, Taiwan, 2006.

[17] T. Cho and J. P. Bello, "Real-time implementation of HMM-based chord estimation in musical audio," in *Proceedings of the International Computer Music Conference (ICMC '09)*, pp. 117–120, August 2009.

[18] K. Martin, "A blackboard system for automatic transcription of simple polyphonic music," Tech. Rep. 385, Massachusetts Institute of Technology Media Laboratory Perceptual Computing Section, 1996.

[19] L. R. Rabiner and B.-H. Juang, "An introduction to hidden Markov models," *IEEE ASSP Magazine*, vol. 3, no. 1, pp. 4–16, 1986.

[20] L. R. Rabiner, "Tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[21] A. M. Barbancho, A. Klapuri, L. J. Tardon, and I. Barbancho, "Automatic transcription of guitar chords and fingering from audio," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 3, pp. 915–921, 2012.

[22] A. Klapuri, "Multiple fundamental frequency estimation by summing harmonic amplitudes," in *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR '06)*, pp. 1–6, Victoria, Canada, 2006.

[23] M. P. Ryynänen and A. P. Klapuri, "Automatic transcription of melody, bass line, and chords in polyphonic music," *Computer Music Journal*, vol. 32, no. 3, pp. 72–76, 2008.

[24] G. D. Forney Jr., "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973.

[25] D. Deutch, *The Psychology of Music*, Academic Press, New York, NY, USA, 1999.

[26] F. J. Harris, "On the use of windows for harmonic analysis with the discrete Fourier transform," *Proceedings of the IEEE*, vol. 66, no. 1, pp. 51–83, 1978.

[27] M. J. Bastiaans, "A sampling theorem for the complex spectrogram, and Gabor's expansion on a signal in Gaussian elementary signals," *Optical Engineering*, vol. 20, no. 4, Article ID 204597, 1981.

[28] Y. C. Eldar and A. V. Oppenheim, "MMSE whitening and subspace whitening," *IEEE Transactions on Information Theory*, vol. 49, no. 7, pp. 1846–1851, 2003.

[29] C.-Y. Chi and D. Wang, "An improved inverse filtering method for parametric spectral estimation," *IEEE Transactions on Signal Processing*, vol. 40, no. 7, pp. 1807–1811, 1992.

[30] F. M. Hsu and A. A. Giordano, "Digital whitening techniques for improving spread spectrum communications performance in the presence of narrowband jamming and interference," *IEEE Transactions on Communications*, vol. 26, no. 2, pp. 209–216, 1978.

[31] T. Tolonen and M. Karjalainen, "A computationally efficient multipitch analysis model," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 6, pp. 708–716, 2000.

[32] P.-E. Danielsson, "Euclidean distance mapping," *Computer Graphics and Image Processing*, vol. 14, no. 3, pp. 227–248, 1980.

[33] A. P. Klapuri, "Multiple fundamental frequency estimation based on harmonicity and spectral smoothness," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 6, pp. 804–816, 2003.

[34] Y. Wei, *Variance, entropy, and uncertainty measure [Ph.D. thesis]*, Department of Stastistics, People's University of China, 1987.

[35] J. J. Hopfield, "Artificial neural networks," *IEEE Circuits and Devices Magazine*, vol. 4, no. 5, pp. 3–10, 1988.

[36] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: a review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 4–37, 2000.

[37] T. Gagnon, S. Larouche, and R. Lefebvre, "A neural network approach for pre-classification in musical chord recognition," in *Proceedings of the Record of the 37th Asilomar Conference on Signals, Systems, and Computers*, pp. 2106–2109, Monterrey, Mexico, November 2003.

[38] E. J. Humphrey and J. P. Bello, "Rethinking automatic chord recognition with convolutional neural networks," in *Proceedings of the 11th IEEE International Conference on Machine Learning and Applications (ICMLA '12)*, pp. 357–362, December 2012.

[39] A. T. C. Goh, "Back-propagation neural networks for modeling complex systems," *Artificial Intelligence in Engineering*, vol. 9, no. 3, pp. 143–151, 1995.