FIATAL MŰSZAKIAK TUDOMÁNYOS ÜLÉSSZAKA XIX.

Kolozsvár, 2014. március 20–21.

AKKORD FELISMERŐ RENDSZER FEJLESZTÉSE

DEVELOPMENT OF A MUSICAL CHORDS RECOGNIZER SYSTEM

Thiago BONFIM⁽¹⁾, Ricardo RAMOS⁽²⁾, Brauliro LEAL⁽²⁾

(1) CAPES Foundation, Ministry of Education of Brazil, Brazil, 70.040-020 Brasilia - DF;
bonfimt@gmail.com

(2) Collegiate of Computer Engineering, Vale do São Francisco Federal University
Foundation, Brazil, 48.902-300, Juazeiro - BA, Country Club, Antônio Carlos Magalhães
av., 510; ricardo.aramos@univasf.edu.br, brauliro.leal@univasf.edu.br

Abstract

The complete automatic musical transcription has been a challenge for the scientists for many decades, and it is still an open issue. This work presents a system capable of recording an audio sample containing a natural musical chord, recognizes it, and shows its name to the user. To achieve the desired results, the concepts about physics of sound, logarithmic distribution of frequency ranges, equal temperament intervals, musical scales and formation of the natural musical chords were studied. It was explained the recording and sampling of the musical chords. The Fast Fourier Transform (FFT) was used to detect the frequency of the musical notes present in the recorded audio samples. These notes are compared to the notes that form each chord, making the recognition possible. The Java programming language under the platform Java Standard Edition was used implement the system. The instruments used to test the system were guitars and electric guitars. Finally, possible future works are listed.

Keywords: digital signal processing, fast Fourier transform, computer music, musical chords recognition.

Összefoglaló

A teljes automatikus zenei átírás volt kihívás a tudósok évtizedek óta, és még mindig nyitott kérdés. Ez a munka bemutatja a rendszer, amely képes rögzíteni az audio mintát tartalmazó természetes zenei akkord, felismeri, és megmutatja a nevét, hogy a felhasználó számára. Ahhoz, hogy a kívánt eredményt, a fogalmakat a fizikai hang, logaritmus eloszlás a frekvenciatartomány, egyenlő temperamentum időközönként, zenei mérlegek és megalakult a természetes zenei akkordok vizsgálták. Elmagyarázták, a felvétel és a mintavétel a zenei akkordok. A gyors Fourier -transzformáció (FFT) alkalmaztunk, hogy észleli a frekvenciáját a hangjegyek jelen a rögzített hang mintákban. Ezek a megjegyzések, mint a megjegyzések a formában minden akkord, így a felismerés lehetővé. A Java programozási nyelv a platform Java Standard Edition -t használjuk végre a rendszer. A használt eszközök, hogy teszteljék a rendszer volt, a gitár és az elektromos gitár. Végül a lehetséges jövőbeli munkák vannak felsorolva.

Kulcsszavak: digitális jelfeldolgozás, fast Fourier transzformáció, computer music, akkordok, zene.

1. Introduction

The automatic musical transcription is a computational process of data digital audio analysis and consequent extraction of symbolic data information to be related with musical structures in higher level representations [11]. This challenge has attracted musicians and

scientists for decades [10]. Between the main problems of this process there are the detection of pitch (frequency) and duration of the notes, rhythm detection, duration of pauses, and the identification and separation of the musical instruments.

The aim of this work is to propose a software project capable of performing the recognition of natural musical chords executed in a musical instrument. Some constraints were defined in the development of this work. Any information related with the instrument used to perform the music sample will be ignored, as well as information about beat, rhythm, bar, movement, dynamics and execution technics. The transcription process will be restricted to the recognition of one natural musical chord in one audio sample. The chosen musical notation for the result information acquired on the transcription process is the Chord Symbol due its simplicity and adequacy to the software requirements.

2. Physics of Music

A Musical Scale is a set of frequencies (that match the musical notes) and the intervals in an octave [2]. The present occidental music uses 12 intervals making the Diatonic Scale, also known as Chromatic Scale. Such intervals are proportionally disposed, that is the reason why this scale is also referred as Scale of Equal Temperament.

An important concept in the Chromatic Scale is the relation between two sounds whose first frequency is double, or half, of the other one. That relation is called octave. A note and its octaves have the same name. For example, the A note at 440 Hz is one octave higher than the note A at 220 Hz and two octaves higher than the note A at 110 Hz. The human hearing perception has ten octaves from 20 Hz up to 20,000 Hz.

We can obtain the value of the frequency of a note in any octave by the multiplier factor δ . Multiplying the frequency of a note by δ gives the value of the frequency of the same note in the next octave [6]. As it was seen:

$$f_{13} = \delta f_{12} \Rightarrow f_{13} = \delta(\delta f_{11}) \Rightarrow f_{13} = \delta^2 f_{11} \Rightarrow f_{13} = \delta^{12} f_1 \Rightarrow$$

$$2f_1 = \delta^{12} f_1 \Rightarrow 2 = \delta^{12} \Rightarrow \sqrt[12]{2} = \delta \Rightarrow \delta = 1.0594631 \quad (1)$$

Thus, the frequencies in a octave of the Chromatic Scale are results of a geometric progression where the common ratio is $\sqrt[12]{2}$ [5]. In 1953, the International Standards Organization (ISO) set the frequency 440 Hz as the standard frequency to enable the calculation of the other ones [7]. Based on this frequency, it is possible to obtain all other frequencies of the Chromatic Scale in any octave by using the Equation 2, where m is the number of the octave (1-7) and n is the interval number (1-12).

$$f = (440)2^{m-5}\delta^{n+2} \quad (2)$$

3. Musical Theory

As it was seen, a musical note is defined by a set of frequencies that are part of a musical scale, where the notes are separated by intervals. In the Latin notation the musical notes are: Dó, Ré, Mi, Fá, Sol, Lá and Si, matching C, D, E, F, G, A and B in the Anglo-Saxon notation. These musical notes are combined with the Musical Accidents, that are represented by adding Sharp (#) or a Flat (b) to the note name, making the 12 intervals of the chromatic scale: C, C#, D, D#, E, F, F#, G, G#, A, A# and B, that are equivalent to C, Db, D, Eb, E, F, Gb, G, Ab, A, Bb and B [2]. At the end of the octave another one begins with the same sequence of notes, in which each note has the double of the frequency of its correspondent in the previous

octave. Observe that between the notes that are not accidents the interval is one step except between the notes E and F, and between the notes B and C. Because of it there is no E $\sharp$ and B $\sharp$ and also there is no F $\flat$ and C $\flat$. A Chord is a set of three or more notes executed simultaneously. The notes of the chords must respect the harmony rules [1]. The chords and notes have similar names. For instance, it is possible to have the C note and the chord C Major. The main musical chords are the Natural Chords that can be Major or Minor. Those are Triad Chords because they are formed by three notes of the Major Natural Scale or Minor Natural Scale. To recognize musical chords it is necessary to identify the performed notes and compare them to the notes of the natural chords.

The Guitar was chosen to allow the development of this software system. But it doesn't mean that the software cannot recognize chord performed in other instruments, such as pianos, violins, or even the human voice, since they share the same musical theory.

4. The Recognizer System

The recognizer system described by this work is capable of recording and processing audio samples containing one major or minor natural musical chord. From this sample the system can conclude which chord has been performed. The sound of the chord is captured by a microphone that transforms the sound in an electric signal that is sampled in an audio interface in a computer, which runs the system. The audio interface transforms the analogic signal in a digital signal [3]. The samples are recorded in signed PCM (Pulse Code Modulation) format at 44.1 kHz and 16 bits. That samples are filtered eliminating amplitudes less than 10% of the greatest one. The intention of the filtering is to reduce the noise.

Any periodic signal, continuous in time, with period $T_0 > 0$, can be described as a sum of sinusoids with frequencies $T_0 = 1/f_0$ and all its harmonics [8]. Discrete signals, like we have after the sampling, can be represented by the contribution of each frequency by using the Discrete Fourier Transform (DFT). The Fast Fourier Transform (FFT), an improvement of the DFT in terms of algorithmic complexity, has been chosen to be used in the system due its execution time is much less than the DFT's. The first one has its algorithmic complexity in order of $O(n \log_2^n)$ while the second has $O(n^2)$ [9]. That is an important point in possible future works approaching real time requirements.

Using the FFT it is possible to produce the amplitude spectrum (the signal in the frequency domain) of the sampled signal that each component is $c_n = \sqrt{a_n^2 + b_n^2}$, where a_n and b_n are, respectively, the real and imaginary parts of the Fourier spectrum of the sampled signal [4]. The amplitude spectrum must be analyzed by the system looking for the six peaks of frequency that match the six guitar strings. When a peak f_a is detected the next candidate peak f_c only can be accepted if the difference is equal or greater than δf_a to avoid the detection of the same note in the same octave due to small variations in the oscillation frequency of the guitar string.

After the system recognizes the frequency peaks it is necessary to associate each one to a musical note. It has to compare the frequencies of each detected peak f_a to the frequencies f_t calculated by the Equation 2. The frequency f_t that has the lowest difference $|f_a - f_t|$ indicates the performed note. To associate the six recognized notes to the three notes of a natural chord, the system must verify if all of that notes match the triad of some chord.

5. Results

The system was developed in the programming language Java and uses the libraries: Java Sound for recording and processing the audio samples; the library JTransforms that

implements the FFT; the library Swing for the system graphic interface; and the library JFreeChart for generating the graphs of the sampled signal in the interface.

Fig.1. Main screen of the system recognizing the chord A# Major.

The Figure 1 shows the main screen of the system where it is possible to record an audio sample containing a musical chord. After recording the audio sample the user can start the analysis of it. This action fills the graphs of the signal in the time domain and in the frequency domain and shows the performed chord. Apart from the software resources, to develop the system it was necessary: a computer with a microphone and a musical instrument that can perform chords, for instance a guitar. The system was tested by using guitars and electric guitars. It was possible to achieve the aim recognizing all major and minor chords in all possible formats in the guitar (or electric guitar) neck using different rhythms of plucking and strumming. The software was tested running Microsoft Windows 7, Fedora 17 (BeefyMiracle) and Apple Mac OS X Mavericks.

6. Future Works and Acknowledgments

It is intended to expand the software to recognize non-triad chords (formed by more than three notes), recognize more than one chord in a single audio sample, and transcript the duration of each note and each Rest (the moment that no note or chord is performed).

The authors thank the *Coordenação de Aperfeiçoamento de Pessoal de Nível Superior* (CAPES) and the Science Without Borders program for the scholarship given to Thiago Bonfim under process number 88888.076197/2013-00.

References

- [1] J. R. Alvira. *Music theory* web. Available in <http://www.teoria.com/>, 2006.
- [2] J. Backus. *The acoustical foundations of music*. W.W. Norton and Company, New York, 2nd edition, 1997.
- [3] A. Balbinot and V. J. Brusamarello. *Instrumentação e Fundamentos de Medidas*. LTC, 2006.
- [4] S. P. Bores. *Training in DSP and media processing*. Available in <http://www.bores.com/>, 1998.
- [5] C. Dodge and T. A. Jerse. *Computer music synthesis, composition, and performance*. W.W. Norton and Company, New York, 2nd edition, 1997.
- [6] J. C. L. Dovicchi. *Novos coeficientes wavelets baseados em intervalos musicais para análise de timbres de instrumentos acústicos*. Tese de PHD, Universidade Federal de Uberlândia, Minas Gerais, 1999.
- [7] I. S. O. ISO. ISO 16:1975 – *acoustics – standard tuning frequency* (standard musical pitch). Available in <http://www.iso.org/>, 1975.
- [8] B. P. Lathi. *Linear Systems and Signals*. Bookman, 2nd edition, 2007.
- [9] S. Marchand. *An efficient pitch-tracking algorithm using a combination of Fourier transforms*. The cost G-6 conference on digital audio effects, 2003.
- [10] K. Martin. *Automatic transcription of simple polyphonic music: robust front end processing*. Technical Report 22, M.I.T. Media Laboratory Perceptual Computing Section Technical Report, Rio de Janeiro, April 2005.
- [11] E. Scheiner. *Extracting expressive musical performance information from recorded music*, 1995.