ECAI 2016 - International Conference – 8th Edition
Electronics, Computers and Artificial Intelligence
30 June -02 July, 2016, Ploiesti, ROMÂNIA

Automatic Music Transcription Software Based on Constant Q Transform

Robert Alexandru Dobre, Cristian Negrescu
Telecommunications Department
Politehnica University of Bucharest
Bucharest, Romania
rdobre@elcom.pub.ro

***Abstract* – The paper presents an automatic music transcription software which uses a constant Q transform as a time-frequency analysis tool showing why this transform is more adequate than the discrete Fourier transform for the presented application. The software can transcribe melodic structures played in any musical key by any equal tempered instrument. The input of the software is represented by an audio file containing the recording to be transcribed and the output is the musical score with standard notation as a PDF file. MIDI (Musical Instrument Digital Interface) files can also be exported which are helpful in the musical production processes.**

Keywords-music transcription; constant Q transform; music production;

I. INTRODUCTION

Music is stored in documents in the form of scores or sheets. A musical score is a graphical representation of a musical work using standard symbols. These specific symbols are in correspondence with the characteristics of each sound, the most important being its pitch (which is directly linked by its frequency), its duration and its temporal placement in the song. The symbol which stores the information about the pitch and duration of a sound is called musical note. Songs are composed of two parts: one called harmonic and another called melodic. The harmonic structure is represented by the chords which form the accompaniment for the melody. The melodic structure is the melody itself, the part that is remembered by listeners. Being iconic for a musical work, the melodic structure has a great importance and it is the one that is targeted by this paper.

Music transcription[1][2] is the process of obtaining the musical score of a melody starting from a recording of it. Traditionally this is done by people which have absolute pitch and vast musical experience. Absolute pitch is the capability of relatively few people to recognize the pitch of a sound without a reference. The musical experience is needed in order to be able to recognize the rhythmic properties of the song and to accurately determine the duration and placement of the notes. Even for these gifted people, music transcription remains a process which takes a lot of time. Parts must be played repeatedly live or from a recording. Music scores are usually written by hand and then introduced note by note into a computer software in order to obtain a typography ready variant, just like a writing manuscript is transformed into a book using word processors.

The paper presents an algorithm which could do all the time consuming hard work delivering directly the final, ready for printing, version of the musical score starting from a recording which can be easily done by any songwriter since most of today's multimedia devices have a record function (smartphones, music players etc.). The algorithm does not need assistance[3].

MIDI files can also be exported by the developed software which will come in handy if the melody is to be made a part of a material whose production already started. Since many multimedia productions are done by teams which independently work far from one another and which can use different software, MIDI is a format accepted by most of multimedia tools and can easily form a way to digitally represent and transfer ideas. MIDI can also be visualized using programs called sequencers, and may be preferred over standard musical score in some modern styles of music.

The paper is structured in four sections starting with this introduction. Section II details the constant Q transform[4] with its advantages over the discrete Fourier transform (DFT) in this particular application, Section III describes the actual music transcription algorithm and Section IV presents results obtained using the software.

II. THE CONSTANT Q TRANSFORM

Since a musical score contains information about the pitch (frequency) and the duration of the notes, it is very similar, at a first glance, with a spectrogram. A spectrogram is a three dimensional representation of a signal in which the time and frequency are two orthogonal axes and the third dimension is the magnitude of the spectral components illustrated using different colors. These similarities suggest that starting from a well-chosen spectrogram, a musical transcription system can be developed. In order to obtain a spectrogram, a transformation must be applied on the signal to determine its spectral content. The most intensively used frequency analysis tool is the discrete Fourier transform[5]. In order to show why this is not the optimal tool for spectral analysis in the case of musical signals, the frequencies of musical

978-1-5090-2047-8/16/$31.00

notes in equal tempered keys must be discussed. There are 12 musical notes per octave. The way frequencies are allocated to each note depends on how an instrument is tuned. Most of today's instruments are equal tempered meaning that the frequencies of the notes are in a geometric progression. The ratio of the frequencies of two notes placed at an octave apart is 2. Combining this with the information about the number of notes in an octave, and the fact that their frequencies must respect a geometric progression, it results that its common ratio is:

$$\frac{f_{next\ note}}{f_{current\ note}} = \sqrt[12]{2}. \quad (1)$$

The DFT for a signal is computed using the following formula:

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi \frac{k}{N} n}. \quad (2)$$

N is the length of the sequence, $x(n)$ is the time domain signal, and $X(k)$ is its DFT. N complex spectral coefficients are obtained which are called bins. Considering F_s the sampling frequency, the frequency of each coefficient can be found using:

$$f = \frac{k \cdot F_s}{N}. \quad (3)$$

It can be observed that the frequency domain is uniformly split in N equal intervals and the spectral resolution of this analysis is constant. This kind of analysis is similar to having N band pass filters with the central frequencies given by (3). In order for the resolution to remain constant, the quality factor (Q) must be different for each filter. It can be calculated that for the filter centered on f_k frequency, with its bandwidth denoted with Δf, its quality factor is:

$$Q = \frac{f_k}{\Delta f} = \frac{\frac{k \cdot F_s}{N}}{\frac{F_s}{N}} = k. \quad (4)$$

This analysis is not adequate for signals whose spectral components are placed at frequencies which follow a geometric progression, like musical signals.

A time-frequency analysis tool which exhibit human hearing characteristics, having a better spectral resolution at low frequencies than at high frequencies, will be more efficient in this case. Using the aforementioned notions, this translates in having a transformation which is similar to a filter bank with band pass filters having the same quality factor. The bandwidths of the filters will be equal on a logarithmic frequency scale, just like the human hearing model suggests. The central frequencies of the filters must follow the rule:

$$f_k = f_0 \cdot 2^{\frac{k}{b}}, k = 0, 1 \ldots \quad (5)$$

where b indicates the number of filters per octave and f_0 is the frequency of the first filter (the lowest frequency that is analyzed). In order to cover the entire frequency domain, the bandwidth of the filter centered at f_k is denoted Δ_k^{CQT} and can be computed using:

$$\Delta_k^{CQT} = f_{k+1} - f_k = f_k(2^{\frac{1}{b}} - 1) \quad (6)$$

It can be easily shown that such a transformation uses the same quality factor for each filter and it can be calculated using the following equation:

$$Q = \frac{f_k}{\Delta_k^{CQT}} = \frac{1}{\frac{1}{(2^{\overline{b}} - 1)}}. \quad (7)$$

By choosing b =12 and f_0 the lowest frequency to be taken into account (depending on the analyzed musical instrument) each frequency bin will be placed exactly at a frequency of a note so the analysis tool is perfectly adapted to the content to be analyzed.

The filter analogy is useful for understanding the advantages of the constant Q transform over the DFT in this particular situation. The expression for actually computing the constant Q transform is obtained by modifying (2) in order to impose constant quality factor filters. Since the spectral resolution differs, this implies that the temporal resolution will also differ meaning that a different number of samples N_k will be used for the computation of each bin. A normalization factor $\frac{1}{N_k}$ is needed and introduced. This way:

$$N \longrightarrow N_k = Q \frac{F_s}{f_k}, \quad (8)$$

$$k \longrightarrow Q, \quad (9)$$

$$X_{CQT}(k) = \frac{1}{N_k} \sum_{n=0}^{N_k - 1} x(n) e^{-j2\pi \frac{Q}{N_k} n}. \quad (10)$$

The algorithm to determine the constant Q transform for a signal is:

- Choose the lowest frequency used in analysis, denoted with f_0, based on the musical instrument that plays the melody.
- Choose the number of octaves which will represent the interest domain (the frequency range of the instrument).
- Choose the number of bins per octave, denoted b (12 to have one bin for each musical note).
- Determine the number of bins to be calculated using the following formula:

$$K = ceil\left(b \cdot log_2\left(\frac{f_{max}}{f_0}\right)\right) \quad (11)$$

where "ceil" means to take the immediately larger integer value.

- Compute the quality factor using (7), N_k using (8) and then $X_{CQT}(k)$ using (10) for $k < K$.

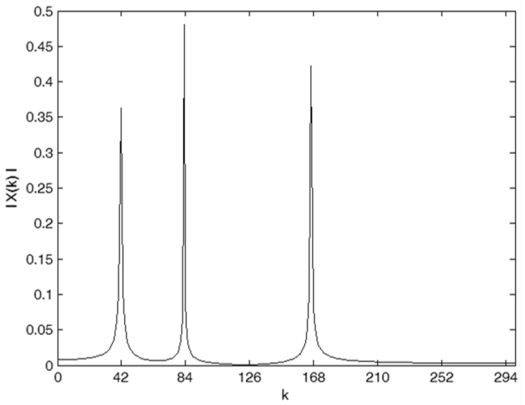

Figure 1. DFT of a signal composed of three sine waves with equal amplitudes and frequencies an octave apart.

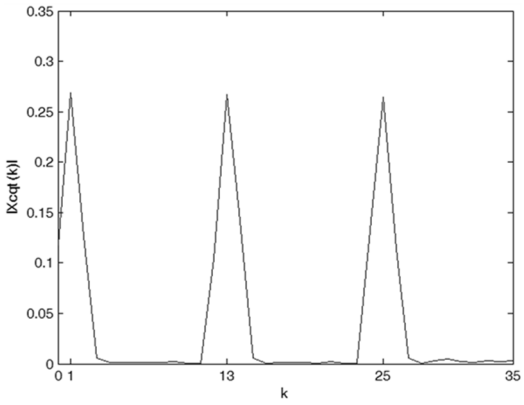

Figure 2. Constant Q transform of a signal composed of three sine waves with equal amplitudes and frequencies an octave apart.

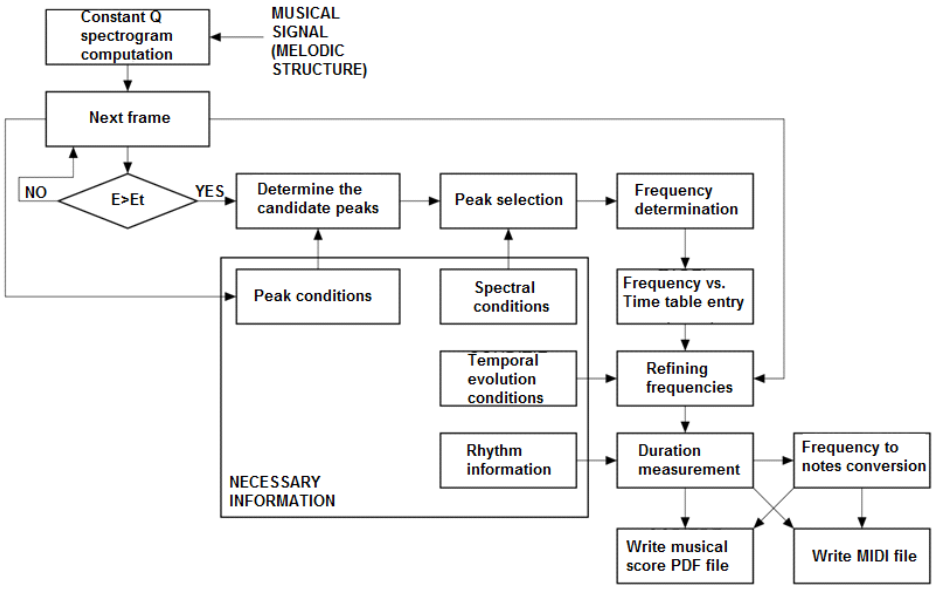

Figure 3. The block diagram of the automatic music transcription algorithm.

The difference between DFT and constant Q transform for a sum of sine waves with equal amplitudes and frequencies respecting a geometrical progression with the common ratio equal with 2 can be observed in Fig. 1 and Fig. 2. The amplitudes are correctly determined using constant Q transform because the frequencies of the signals are perfectly aligned with the corresponding frequency bins. Some components fall between bins in the DFT case and amplitudes are not correctly determined. The constant Q transform has also other applications in music[6].

III. THE MUSIC TRANSCTRIPTION ALGORITHM

The block diagram of the automatic music transcription algorithm is depicted in Fig. 3. It was implemented using Matlab. The input file containing the recording of the melody is loaded into the program. The constant Q spectrogram of the signal using 24 bins per octave is computed and it will serve as a base for all the remaining steps. The resolution is half of a semitone (frequency difference between two consecutive notes). The user must only set the rhythm information – the tempo in beats per minute (BPM or usually quarter notes per minute). This is easy to determine manually, but very hard to be done automatically using only a melodic structure.

The other operations are done on frames, not on the whole signal. The temporal length of each frame is dependent on the parameters of the spectral analysis tool (f_0, b). Overlapping fames can be used. For each frame, if its energy (E) is greater than a threshold value (Et), the candidate peaks are determined. These must have an amplitude greater than 75% of an "average maximum" peak value computed using the greatest peaks from 10 past and 10 future frames. The candidates must also be spaced at least by one semitone. From the candidates, the peak which will give the pitch of the note is selected using a recursive sequence: the candidate peak placed at the lowest frequency is called the main candidate. It is checked if another peak placed at an octave below the main candidate's frequency has an amplitude greater than 10% of the main candidate's amplitude, it will become the new main candidate. This runs until the main candidate is not replaced. The position of the peak (bin number) will easily give the frequency of the note using (5). An entry which contains the temporal information (given by the frame number) and the frequency of the note is added to a time-frequency table.

Since the durations of the notes can only have certain values depending on the BPM, corrections can

be made if errors occur in the phase in which the time-frequency table is determined. This way very short notes (shorter that 1/32 of a quarter note) are considered errors and are aligned with their neighbors. The mechanism can be observed in Fig. 4.

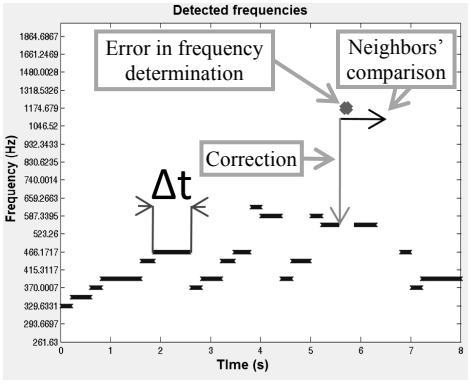

Figure 4. Highlight of frequency determination error correction and duration measurement.

Based on the aforementioned table, each frequency is converted in note names (Do, Re, Mi etc.). The duration of each note in seconds is measured, denoted Δt in Fig. 4, then using the BPM information it is converted into musical durations (whole, half, quarter, eight etc.) using the formula:

$$duration = 4\frac{60}{\text{BPM}}\frac{1}{\Delta t}. \qquad (12)$$

A duration equal to 1 means whole note, 2 means half note, 4 means quarter note etc.

In the end a free external program (LilyPond) is called which, based on the table now containing musical durations and pitches, it generates the PDF file representing the musical score. A Matlab MIDI free library[7] can be used to also write the MIDI files.

IV. Results

Tests were done using piano, violin and guitar. The score obtained for the well-known jazz theme "Pink panther" played by a synthesized piano is presented in Fig. 5 as it was given by the proposed software.

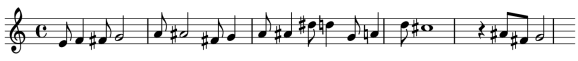

Figure 5. The musical score obtained using the proposed software for the "Pink panther" jazz theme.

The time-frequency representation which was used to generate the musical score presented in Fig. 5 is illustrated in Fig. 6. It can be observed that the algorithm successfully corrected the error in frequency detection highlighted in Fig. 4. The algorithm gives good results even for fast musical phrases. The score presented in Fig. 7 was obtained for fast violin arpeggios. The representation of the time-frequency table associated with this experiment is depicted in Fig. 8. The proposed software gives good results using simple melodic structures which form most of today's music. For more complex melodies which contain more than one note at a time, multiple fundamental frequency estimators must be included.

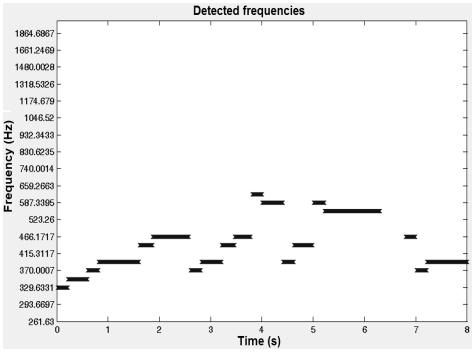

Figure 6. The final, corrected, representation of the time-frequency table which was used to generate the musical score for the "Pink panther" jazz theme.

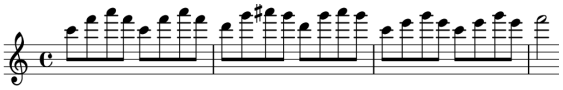

Figure 7. The score obtained for violin fast arpeggios.

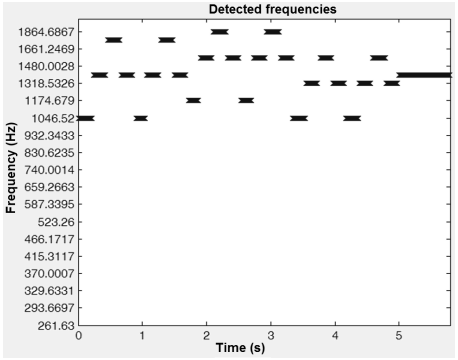

Figure 8. The representation of the time-frequency table which was used to generate the musical score for the violin melody.

References

[1] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, A. Klapuri, "Automatic music transcription: challenges and future directions," Journal of Intelligent Information Systems, vol. 41, pp. 407–434, July 2013.

[2] A. Klapuri, M. Davy, "Signal processing methods for music transcription, " Springer Science & Business Media, 2007.

[3] S. Siddharth, E. Benetos, N. Boulanger-Lewandowski, T. Weyde, A. S. d'Avila Garcez, S. Dixon, "A hybrid recurrent neural network for music transcription," 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP*)*, pp. 2061–2065, 2015.

[4] C. Schörkhuber and A. Klapuri, "Constant-Q transform toolbox for music processing, " 7th Sound and Music Computing Conference, 2010.

[5] C. Marghescu and A. Drumea, "Modelling and simulation of energy harvesting with solar cells," Proc. of the Advanced Topics in Optoelectronics, Microelectronics, and Nanotechnologies 2014, pp. 92582L-92582L8, February 2015.

[6] C. Schörkhuber, A. Klapuri, A. Sontacchi. "Audio pitch shifting using the constant-q transform," Journal of the Audio Engineering Society, vol. 61, issue 7/8, pp. 562–572, July 2013.

[7] E. Tuomas and P. Toiviainen. "MIDI toolbox: MATLAB tools for music research," 2004.