Recognition of Musical Chord Notes

GIOVANNI COSTANTINI, DANIELE CASALI
Department of Electronic Engineering
University of Rome “Tor Vergata”
Via del Politecnico 1, 00133 - Rome
ITALY

Abstract: - We developed a new algorithm for automatic recognition of musical chord notes. In the chord recognition field, we often refer to the “Chord Spectrum”, that is the chord representation in the frequency domain. The main concept on which the “chord recognition” theory is based is that we need to find similar groups of sinusoidal tones (tone patterns) belonging to the chord spectrum, through which we can describe the chord as an acoustic profile, with the help of “generative subspectra”. The work done by A. Tanguiane during last decade was the starting point to our study that considers chords played by one or more instruments. In his research, Tanguiane described the chord features and used the information taken from the autocorrelation of chord frequency components to recognize it. He considered that partials forming the chord was equally spaced in a logarithmic way in the frequency domain, implying equal distances correspond to equal musical intervals. To obtain the components equally spaced in a logarithmic way, we use the QFT (Q-constant Fourier Transform), introduced by J.C. Brown in 1991, and also used for the recognition of the fundamental frequency of each note. The QFT allows to show the energy of the singular frequencies in a logarithmic scale spectrum. Autocorrelation is evaluated over the components of the QFT that exceed a certain threshold value. The developed algorithm allows us to obtain good results both for recognition of two, three and four notes chords.

Key-Words: - Recognition, Musical Chord, Pattern Recognition in Music.

1 Introduction

By “music recognition”, we mean studies in computer simulation of music perception, which contribute to developing systems for automatic notation of performed music. The performance of a system for automatic notation is supposed to be analogous to that of speech recognition systems: acoustical data at the input and music score printing at the output.

Music recognition is needed in :
- Music education (computer listening to pupils and judging their skills, automatic accompaniment);
- Computer-aided performance (simulation ensemble interaction of the computer and the soloist);
- Music publishing (acoustical input of music data);
- Musicology (automatically transcribing and analyzing live performance, in particular, folk music; facilities for music data input for music data bases);
- Recording engineering (visualizing recording and tape editing).

Music recognition systems are also desirable for music lovers and amateur bands who would like to initiate their favorite performers. Music recognition devices could meet the demand for precise transcriptions of musical pieces which are not available form music publishers.

A complete system of music recognition consists of three interfaced subsystem for:
- Acoustical recognition
- Music analysis
- Music printing

The task for acoustical recognition is to determine the number of simultaneously sounding parts along with their dynamic and timbre specifications, to recognize instruments, to segment the signal into locally constant and transient segments, to determine periodicity, and to derive pitch trajectories for all voices.

Music analysis includes recognition of time, tempo, tonality, note values and their relative durations, techniques of execution, dynamics, and other musical characteristics, which are fixed by notation. At the given stage of data processing, the acoustical information is interpreted semantically and represented symbolically.

Music printing, i.e. printing of musical scores, is rather a technical problem which has been solved already [8]. Most of the problems arising in music recognition have been considered in different disciplines. For example, pitch and timbre recognition are studied in psychoacoustics, more technical items as signal segmentation are developed in speech recognition, tonality determination is discussed in quantitative musicology, etc. For the computer pitch recognition, we can mention the work by E. Terhardt [1] and J.C. Brown [2], while, for the timbre recognition, the works by J.C. Brown [3], K. D. Martin [4] and one of the authors [5,6,7].

The problems of rhythm/tempo recognition and chord recognition (main task for voice separation) are very important in this field. In the last mentioned branch, we recall the work done by A. Tanguiane [9]; it was the starting
point to our study that considers chords played by one or more instruments.

This paper is organized as follows: section 2 illustrates the QFT (Q-constant Fourier Transform) by J.C. Brown; section 3 describes the used musical signal processing for chord recognition; experimental results will be reported and discussed in section 4.

2 The Q-constant Frequency Transform

A frequency analysis is first performed on notes played by musical instruments, in order to detect the signal harmonics. Using the Fast Fourier Transform (FFT), which is very rapid and therefore suitable for real-time calculations, the frequency resolution could be not sufficient. In fact, a FFT with 4096 temporal samples $x[n]$ on a sound recorded with the usual sampling rate (SR) of 44100 Hz, carries out a resolution of about 10.8 Hz between two FFT samples. This is not sufficient for low frequency notes, where the distance between two adjacent semitones is about 8 Hz (C3, 131 Hz and C#3, 139 Hz). The frequency resolution will get better if a higher number of temporal samples are used (with 8192 samples the resolution is of about 5.4 Hz), but that requires larger temporal windows for a fixed SR. In this case, the analysis of the instantaneous spectral information of the musical signal makes worse. To solve this problem, a Q-constant Frequency Transform (QFT) is used to detect the fundamental frequency of the note. In the QFT transform, described in [1], the frequency channels are not linearly spaced – as in the FFT – but logarithmically spaced. It suits better for musical notes, that are based on a logarithmic scale. The algorithm presented in this article, needs to calculate the autocorrelation in a logarithmic spaced frequency, so the QFT give us ready data for this calculation. Moreover, the QFT uses a varying number of temporal samples $N[k]$ for calculating the frequency transform, more samples for calculating the transform of lower notes than of higher notes. This adaptive windowing yields to a good frequency resolution for all the notes of the tempered scale. As in a musical octave there are 12 semitones logarithmically spaced, the frequency value between two adjacent semitones increments of about 6%. As an example, if we consider two notes C4 (261.6 Hz) and C#4 (277.2 Hz), we have an increase of $\frac{277.2 - 261.6}{261.6} = 0.0596$, which corresponds to a Q-value of $Q = \frac{f}{\Delta f} = \frac{277.2}{0.0596} \approx 17$.

In the present work, we use different values of $Q$, according to the precision that we need. For the QFT calculation, it is necessary to establish two other parameters: the minimum frequency $f_1$ from which the analysis starts and the number of channels to consider $N_p$. Then, the QFT is computed as follows:

$$X[k] = \sum_{n=0}^{N[k]-1} w[k, n] \cdot x[n] \cdot e^{-j2\pi Q n/N[k]}$$

for $k=1 \ldots N_p$.

where $w[k, n]$ indicates the Hamming windowing function:

$$w[k, n] = \alpha + (1 - \alpha) \cdot \cos\left(2\pi n/N[k]\right)$$

with $\alpha = 25/46$.

The number of temporal samples to consider for each channel is:

$$N[k] = \frac{f_k \cdot Q}{f_1}$$

For example, assuming C4 (261.6 Hz) as the lowest note, the number of samples to consider is:

$$N[k]_{\text{max}} = N[1] = \frac{44100 \cdot 34}{261.6} \approx 5732$$

Once computing the QFT, the identification of sound fundamental frequency is made by a simple algorithm. In Fig.1 and Fig.2 we show a FFT and a QFT of violin scale from G3 to G5. It is very clear how in the FFT harmonics are equi-spaced but the pitch is not very easily recognizable; on the other way, in the QFT we can see how at each note of the violin corresponds a simple translation of the spectrum at right.

2 Musical Tones and Chord Representation

In chord recognition, we often refer to the “Chord Spectrum”, which is the spectrum of the chord, in the frequency domain. It can be considered as generated by the translation of the “Spectral Tone” in a logarithmic spaced frequency.

We now describe some fundamental concepts used in representation of tones. Through a musical tone (or harmonic), we recognize a sound with a clear Pitch Salience. In this way, the tone is seen as generated by a periodic or quasi-periodic oscillation of the air pressure. Its fundamental frequency is associated to tone’s pitch, and, for musical sounds, it refers to the pitch of the single musical note.

The logarithmic scale in pitch perception, together with the insensibility of human ear to the signal phase, gives us some important properties of perception. The use of logarithmic representation, with linearly structured patterns, like spectral tones (characterized by a ratio of partial frequencies like $1:2:3: \ldots : k$) yields a no linear compression on scale and a consequent irreducibility of them.

All of that is the reason why harmonic tones are perceived as whole music objects, in spite of a collection of
sinusoidal partials. Irreducibility if harmonic spectra is proved for power spectra with real positive coefficients, but not for the ones with complex coefficients. This implies that harmonic tones can be perceived as whole musical objects only if ear is insensible to the phase of the signal, and this shows that insensibility has to be seen as a advantageous property more than an imperfection of the ear. Together with logarithmic spaced scale, this property is necessary to the decomposition of a sound coming from different physical sources, contributing to recognize all acoustic sounds that are characteristics of surrounding environment. In Signal Processing the frequency axis is expressed in linear scale. For what concerns musical analysis, a logarithmic spaced frequency is preferable. In fact, equal distances in log2-scale correspond to equal musical intervals.

According to Fourier’s theorem, any musical tone with a periodic waveform can be decomposed in an infinite sum of harmonics. Since the range of perceptible frequencies is approximately limited to the band 20-20000 Hz, the infinite sum can be always reduced to a finite one.

In the present work we refer to the audio spectra, i.e. we assume that the frequency axis is log2-scaled, implying equal distances correspond to equal musical intervals. Since man is not sensitive to the phase of the signal, we restrict our consideration to power spectra, where the phase of the overtones is ignored. Finally, the spectra to be considered are discrete, i.e. it is assumed that the frequency range is divided into bands wherein the signal amplitude is measured by positive numbers.

In chord recognition, we often refer to the “Chord Spectrum”, which is the spectrum of the chord in the frequency domain. It can be considered as generated by the sum of “Tone Spectra” in a logarithmic spaced frequency.

3 Proposed Algorithm and Experimental results

A frequency analysis is performed in order to detect the harmonic content of musical signal. For this aim, we perform a Q-constant Fourier Transform (QFT), described in [1], which returns the logarithmic spectrum of the signal. In this way, the QFT channels correspond to the musical harmonics of the chord tones. Moreover, with the QFT, we obtain a higher resolution for high notes and a lower resolution for low notes: it suits better for musical notes that are based on a logarithmic scale. We introduced a threshold, to avoid all the frequency components under a fixed level, so obtaining that we called the Discrete Chord Spectrum (D.C.S.).

Then, we perform an autocorrelation over the D.C.S.: the peaks of the autocorrelation function correspond to the intervals (i.e. the distances in semitones) between each chord note and the chord fundamental, easily found through the QFT [5,6].

4 Experimental results

A huge amount of computer simulations have been performed to ascertain the behavior of the proposed algorithm. We tried chord with two, three and four notes. Moreover, we used different instruments to show how the algorithm behaves according to the peculiarity of their sound spectrum. We considered for the experiments the following instruments: clarinet, violin, flute, and sax. These instruments present very different spectral characteristics: for example, the spectrum of the clarinet has the odd harmonics higher than the even ones, while the spectrum of the flute has a number of harmonics lower than other instruments (therefore, we have less partials to analyze and so we make greater errors). We also made a lot of proofs with groups of instruments, in which each note of the chord was played by a different instrument. In all experiments, we chose a threshold of –30 dB to compute the D. C. S.

In the following the results of the experiments are illustrated. As a first example of application of the cord recognition algorithm, we consider the two-notes chord (c4;b4). We choose the following parameter values:

- number of octaves to scan = 5
- lowest octave to analyze = 4
- threshold value = -30dB
So the number of frequencies of interest is $12 \times 5 + 1 = 61$ (12 are the semitones of an octave, and +1 is introduced to consider the note of the higher octave). Tab.1 shows the results:

Tab.1 shows the QFT autocorrelation values for this chord. We can observe that the highest autocorrelation value is 0.21467, corresponding to a distance of 11 semitones. Once calculated the chord fundamental c4 by the QFT, we conclude that the analyzed chord is (c4;b4).

<table>
<thead>
<tr>
<th>Autocorrelation values</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 0.0138</td>
</tr>
<tr>
<td>2 0.01045</td>
</tr>
<tr>
<td>3 0.01457</td>
</tr>
<tr>
<td>4 0.0142</td>
</tr>
<tr>
<td>5 0.01673</td>
</tr>
<tr>
<td>6 0.00914</td>
</tr>
<tr>
<td>7 0.01451</td>
</tr>
<tr>
<td>8 0.05617</td>
</tr>
<tr>
<td>9 0.03207</td>
</tr>
<tr>
<td>10 0.01183</td>
</tr>
<tr>
<td>11 0.21467</td>
</tr>
</tbody>
</table>

Tab. 1 – Autocorrelation values of chord (c4;b4).

The same procedure can be extended also to chords with a higher number of notes and played by more instruments. The results are summarized in Tab.2.

<table>
<thead>
<tr>
<th>Instruments</th>
<th>Two-notes chords</th>
<th>Three-notes chords</th>
<th>Four-notes chords</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clarinet</td>
<td>98 %</td>
<td>97 %</td>
<td>95 %</td>
</tr>
<tr>
<td>Violin</td>
<td>96 %</td>
<td>95 %</td>
<td>93 %</td>
</tr>
<tr>
<td>Flute</td>
<td>92 %</td>
<td>88 %</td>
<td>81 %</td>
</tr>
<tr>
<td>Sax</td>
<td>93 %</td>
<td>92 %</td>
<td>85 %</td>
</tr>
<tr>
<td>Group</td>
<td>93 %</td>
<td>90 %</td>
<td>83 %</td>
</tr>
</tbody>
</table>

Tab. 2 – Resume table of chord recognition tests.

5 Conclusion

We developed a new algorithm for chord recognition, which finds the intervals in a musical chord examining the autocorrelation of the chord spectrum. We obtain the chord spectrum through the QFT, which gives us a ready logarithmic spectrum, suitable for musical notes that are based on a logarithmic scale. Then, we consider the autocorrelation of the chord spectrum to identify the chord interval. The developed algorithm allows us to yield good results in recognition of two, three and four notes chords.

References:


