A Practical Approach to the Chord Analysis in the Acoustical Recognition Process

Marcin Szlenk[1] and Wladyslaw Homenda[2]

[1] Institute of Control and Computation Engineering, Warsaw University of Technology, 00-665 Warsaw, Poland, m.szlenk@ia.pw.edu.pl
[2] Faculty of Mathematics and Information Science, Warsaw University of Technology, 00-661 Warsaw, Poland, homenda@alpha.mini.pw.edu.pl

Abstract: The identification of simultaneously sounding notes is one of the key problems for music recognition. In this paper we outline the localization of this problem in the whole process of music recognition and propose a practical solution.

Keywords: music recognition, acoustical recognition, chord recognition, automatic identification, spectrum analysis

1 Introduction

1.1 Problem Definition

Music recognition is the mathematical analysis of an audio signal generated by musical instruments and its conversion into musical notation. The input data for the system of music recognition is a digitally sampled signal representing the analogue sound waveform generated by musical instruments. A typical example of such data is the contents of a CD-Audio disc. From the user's point of view, the operation of an automated music recognition system is as follows: in response to digital sound data at the input, the system returns a musical score at the output.

The input sound data can be received as a piece of music performed by only one musician, playing one instrument (e.g. a piano sonata, a violin sonata etc.) or as multi-instrument music, i.e. as a piece of music performed by many musicians (e.g. a string quartet or chamber orchestra). Multi-instrument music can be seen as a union of single-instrument music and in this case the recognition system could be based on a system recognizing music performed on only one instrument. Such a solution would require the separation of input data into the parts played on single instruments. The implementation of this idea is, however, a long way off. The problem of single-instrument music recognition is still a challenge without, as yet, a satisfactory solution. The more complex the piece of multi-instrument music is, the more difficult it is to recognize solo instrument signals because they interfere with each other, which moves the problem of music recognition to a more complex class[1]. On the other hand, conversion of an acoustic signal into digital format distorts the music data, which raises another issues to be dealt with. These arguments allow for the conclusion that multi-instrument music recognition is still over the horizon of current technological development. Therefore, in this paper the discussion is restricted to the problem of single-instrument music only.

1.2 Music Recognition Stages

In the process of generating a musical score, using an input audio signal, there are two unique stages (see Fig. 1):

- *acoustical recognition*, and
- *music analysis*.

The aim of acoustical recognition is to determine the number of simultaneously sounding notes, to establish their pitches and time parameters, i.e. the start time and duration. The result of this stage gives sufficient information to generate a proper sequence of MIDI (*Musical Instrument Digital Interface*) commands. The MIDI standard is primarily used to control digital musical instruments [9]. The MIDI commands only contain the information about the occurrence of a certain event at a given moment (e.g. the Note On and Note Off events define the starting and ending time of a note) and they no longer contain any additional information about the real sound waveform.

Music analysis is engaged, on the contrary, in determining the information required to generate musical score from data received at the acoustical recognition stage (e.g. in the form of a MIDI file). It includes the recognition of tempo, tonality, note values, dynamics and other musical characteristics which can be reflected in printed music.

1.3 Acoustical Recognition

The music recognition stages above can be implemented completely independently because both solve different technological problems. This paper deals with acoustical recognition and, more precisely, the problem of simultaneously sounding note recognition. Regardless of the way in which the acoustical recognition stage is implemented, there is always the issue of how to recognize the number and determine the pitches of simultaneously sounding notes.

Let us consider the simplest schema of acoustical recognition. The input sequence of samples is divided into short time periods and then the notes employed

[1] The problem of multi-instrument music recognition is similar to speech recognition, where a number of people speak simultaneously.

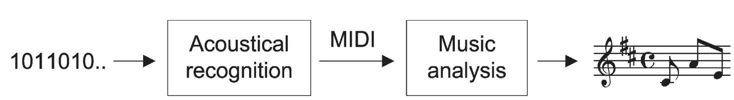

Fig. 1. Music recognition stages

in consecutive periods are recognized. Then, notes with the same pitch, recognized in successive periods of time, are glued together. Below, we present a more formal description of this solution:
The input samples of sound can be observed by moving them over a window with a certain width. Let us assume that Z *denotes the set of notes which are currently being played. The operation of inserting a note into the set* Z *is accompanied by generating a MIDI event indicating the starting time of a note as the moment corresponding to a current position of the window. By analogy, the operation of removing a note from the set* Z *is accompanied by generating an event indicating the ending time of it. With the initially empty set* Z*, the following steps are performed:*

1. *Place the window at the beginning of the sequence of samples.*
2. *Define the set* Z' *of notes occurring in the fragment visible in the window.*
3. *For all the notes in set* Z *perform:*
   - *If the note* n *currently being played is not among the identified notes* (n $\in$ Z *and* n $\notin$ Z') *then remove it from set* Z.
4. *For all the notes in set* Z' *perform:*
   - *If the recognized note* n' *is not among the notes currently being played* (n' $\in$ Z' *and* n' $\notin$ Z) *than insert it into set* Z.
5. *If the analyzed fragment was the last one then remove the remaining notes from set* Z *and finish. If not, shift the window to the next position* (*move at a distance equal to the window's width*) *and repeat the above steps, starting at* 2.

The solution to step 2 in the above algorithm is the main objective of this paper.

The above algorithm requires further explanation. We should take into account that the width of the window, which we move over the input data, determines the time resolution for the output MIDI information. As the window is narrowed we can identify start times and the duration of notes more accurately. In the case of most musical compositions the sufficient size is a width equal to 50 ms. Obviously, the narrower the window we use, the better resolution we get but the correct recognition of notes becomes more difficult.

2 Musical Instrument Sound

2.1 Introduction

Sound generated by musical instruments is an acoustic wave. Such a wave has a complex structure that does not have a precise mathematical model. In this paper we discuss selected aspects of sound generated by musical instruments. Mathematical modelling estimates aspects explored in the paper and provides tools for the analysis of selected features of the sound of musical instruments.

2.2 Tone and Frequency

The sound of musical instruments is an acoustic wave at a certain frequency or rather a series of such waves with differing frequencies interfering with each other. Analyzing a single sound generated by one instrument, e.g. the sound generated by a piano string in reaction to hitting one key of the keyboard, a sinusoidal wave of fundamental frequency f is generated (also called the first harmonic) and its higher harmonic frequencies, i.e. sinusoidal waves of frequencies equal to f, $2f$, $3f$, $4f$, etc. can be detected. All these waves are generated by a vibrating string since not only the whole string vibrates, but also its parts of lengths equal to 1/2, 1/3, 1/4, etc. do. So then we can interpret sound generated by a string as a collection of its *harmonic partials*. On the other hand, all harmonic partials aggregated create a periodic wave with a complex shape and frequency f. Thus, the frequency of the first harmonic is taken as the frequency of the sound. The set of values of the amplitudes of the harmonic partials is called the *harmonic spectrum* of sound. The spectrum decides about timbre and colour of sound and differentiates between the same tones generated by different instruments, e.g. violin, flute, piano, etc. For instance, the spectrum of the clarinet sound initially has odd harmonic partials much stronger than others [1] (see Fig. 2).

The spectrum of a given instrument may change depending on the fundamental frequency of a tone. It may even happen that the presence of a given harmonic partial depends on the fundamental frequency. However, changes are usually rather small and do not affect music recognition systems, in most cases.

2.3 Musical Scale

Frequency as a physical parameter of a sound corresponds to ear impression. We can experience lower and higher tones. Some tones are mapped to the musical scale. For instance, 440 hertz frequency corresponds to the note A_4 – the symbol A_4 denotes tone A in octave number 4. The 880 hertz frequency corresponds to the note A_5, i.e. to note A in the octave number 5. Thus we have the following sequence of tones A: A_1 – 55 hertz, A_2 – 110 hertz, A_3 – 220 hertz, A_4 – 440 hertz, A_5 – 880 hertz, A_6 – 1760 hertz, A_7 – 3520 hertz, A_8 – 7040 hertz, A_9 – 14080 hertz. All these tones are denoted by letter A since ear experience is very similar for all of them. Note that higher A tones are harmonic for lower A tones. The frequencies between two neighboring tones A are split into 12 intervals called *halftones*. Notes corresponding to consecutive halftones are: A, $A^\sharp$, B, C, $C^\sharp$, D, $D^\sharp$, E, F, $F^\sharp$, G, $G^\sharp$. The proportion of the frequencies corresponding to two consecutive halftones is equal to $\sqrt[12]{2} : 1$ (see Table 1).

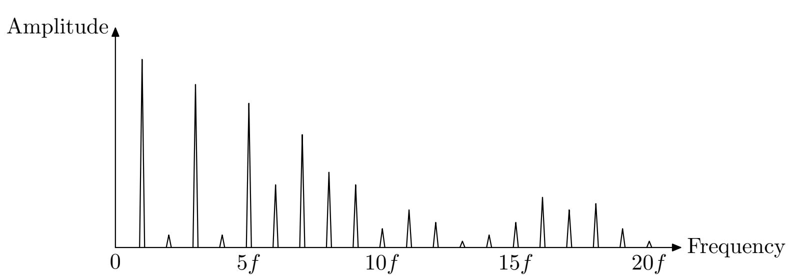

Fig. 2. Clarinet sound spectrum (f = 233.08 Hz)

Table 1. Notes and their frequencies (Hz) [1]

C_2	65.41	C_3	130.81	C_4	261.63	C_5	523.25
$D_2^\flat/C_2^\sharp$	69.30	$D_3^\flat/C_3^\sharp$	138.59	$D_4^\flat/C_4^\sharp$	277.18	$D_5^\flat/C_5^\sharp$	554.37
D_2	73.42	D_3	146.83	D_4	293.66	D_5	587.33
$E_2^\flat/D_2^\sharp$	77.78	$E_3^\flat/D_3^\sharp$	155.56	$E_4^\flat/D_4^\sharp$	311.13	$E_5^\flat/D_5^\sharp$	622.25
E_2	82.41	E_3	164.81	E_4	329.63	E_5	659.26
F_2	87.31	F_3	174.61	F_4	349.23	F_5	698.46
$G_2^\flat/F_2^\sharp$	92.50	$G_3^\flat/F_3^\sharp$	185.00	$G_4^\flat/F_4^\sharp$	369.99	$G_5^\flat/F_5^\sharp$	739.99
G_2	98.00	G_3	196.00	G_4	392.00	G_5	783.99
$A_2^\flat/G_2^\sharp$	103.83	$A_3^\flat/G_3^\sharp$	207.65	$A_4^\flat/G_4^\sharp$	415.30	$A_5^\flat/G_5^\sharp$	830.61
A_2	110.00	A_3	220.00	A_4	440.00	A_5	880.00
$B_2^\flat/A_2^\sharp$	116.54	$B_3^\flat/A_3^\sharp$	233.08	$B_4^\flat/A_4^\sharp$	466.16	$B_5^\flat/A_5^\sharp$	932.33
B_2	123.47	B_3	246.94	B_4	493.88	B_5	987.77

2.4 Monophony vs. Polyphony

Monophony is music having only one voice line. Roughly speaking this means that in any time at most one note can be played. By contrast, polyphony is a kind of music with many voice lines, which means that many notes may be played at a time. Recognition of monophonic music is much easier than recognition of polyphonic music.

In the case of monophonic music, the input sound signal can contain only a single harmonic structure (see Sect. 2.2). Excluding usually small inharmonic partials and additional noise, such a sound signal may be approximately described by

$$x(t) = \sum_k A_k(t)\sin(k\omega t + \phi_k), \tag{1}$$

where $x(t)$ is the sound signal in the time domain; ω is the fundamental frequency of the current note; $A_k(t)$ is the amplitude of the kth harmonic at time t; ϕ_k is the phase of the kth harmonic.

In this case acoustical recognition amounts to detecting the fundamental frequency, finding a note corresponding to it (having in mind that the detected fundamental frequency usually differs slightly from that defined in Table 1) and establishing note duration.

Recognition of polyphonic music poses a more complicated problem. Since many notes may sound at the same time, the analysis of the spectrum must consider many harmonic structures corresponding to different simultaneous notes. Harmonic partials of different notes may superimpose on each other, so then the analysis of such data is much more difficult than in the case of monophony.

3 Mathematical Apparatus

3.1 Frequency Analysis

One of the most popular tools for determining the harmonic contents of a signal is the *Fourier transform*. For analyzing the signal by virtue of its N input samples (sequence $x(n)$), a so-called *discrete Fourier transform* (DFT) is used and is defined as

$$S(m) = \sum_{n=0}^{N-1} x(n) e^{-i2\pi nm/N}, m = 0, 1, 2, \ldots, N-1. \tag{2}$$

For N input samples in the time domain, DFT establishes the harmonic contents of the input signal in N equally spaced points of the frequency axis. For a certain m, the value $S(m)$ (the value of the DFT) is the value of a spectrum at the frequency

$$f_{analysis}(m) = \frac{mf_s}{N}, \tag{3}$$

where f_s denotes the frequency used to sample the input signal. For example, in the case of data coming from CD-Audio this frequency is equal to 44.1 kHz.

When we examine signals in the frequency domain we are usually interested in the values of their power in comparison to the power of another signal. If we show the instantaneous power of signals, represented by successive values of the DFT, then the easiest way is to compare them with the partial of the highest power

$$S_{power}(m) = 20 \log_{10} \left(\frac{|S(m)|}{|S(m_{max})|} \right) \text{dB}. \tag{4}$$

where m_{max} is the index of the DFT value with the highest power. It is the so-called *normalized decibel scale*. Obviously the highest value on such a normalized scale is 0 dB.

3.2 Spectrum Analysis

As a result of the frequency analysis we obtain information about the harmonic contents of the input signal. By virtue of this information we would like to find out which note, or simultaneously sounding notes, correspond to the received harmonic structure. This problem appears to be very interesting especially if we do not know the number of simultaneously sounding notes and we do not know anything about the instrument from which the input signal comes. Tanguiane [7] reduced this problem to the search of the appropriate *deconvolution* of the chord spectrum. Only an outline of this approach will be presented here.

In practice spectrum analysis is restricted to a certain frequency range. The lower limit is defined by the fundamental frequency of the lowest note which is to be recognized and the upper limit is a consequence of the frequency used to sample the input signal[2].

Let us divide the frequency range which we analyze into bands wherein the signal power is represented by one value. By *discrete spectrum* we understand the expression of the form

$$S = S(x) = \sum_{n=0}^{N-1} S_n \delta(x - n) = \sum_{n=0}^{N-1} S_n \delta_n, \tag{5}$$

[2] The spectrum of a discrete signal is a periodic function with the period equal to the sampling rate.

where N is the total number of frequency bands; n is the index of a frequency band; S_n is a value interpreted as the signal power in the nth frequency band; δ_n is the Dirac delta function, i.e. the unit impulse at the nth frequency. For the given discrete spectrum S we can introduce the conception of *Boolean spectrum* (associated with S) defined by

$$s = \bigvee_n s(n)\delta_n, \quad s(n) = \begin{cases} 0 \text{ if } S_n = 0, \\ 1 \text{ if } S_n \neq 0. \end{cases} \tag{6}$$

Each successive note on the musical scale has the frequency $\sqrt[12]{2}$ times larger than the preceding one (see Sect. 2.3). It means that equal distances of notes' pitches (e.g. the interval equal to one octave) do not correspond to equal distances on the frequency axis. For example, the musical distances between the notes A_3–A_4 and A_4–A_5 are equal to an octave but, in terms of their fundamental frequencies, these distances are respectively 220 and 440 Hz. The corresponding distances on both scales may be achieved by rescaling the frequency axis with the $\log_2$ function. The index of the frequency band wherein falls a frequency f is defined by the formula

$$n = \left\lfloor C \log_2 \frac{f}{f_0} + 0{,}5 \right\rfloor , \tag{7}$$

where C is the constant, equal to the number of frequency bands per octave; f_0 is the middle of the frequency band with the index 0.

For $\log_2$-scaled frequency axis a Boolean spectrum s of a sound of several simultaneously sounding notes can be treated as generated by multiple translations of a Boolean spectrum of one note. The number of notes and their musical interval (precisely, the number of frequency bands) in relation to a note with the spectrum t is defined by the *interval distribution* i. Thus the spectrum s has the form of the following

$$s = \bigvee_n s(n)\delta_n = t * i, \tag{8}$$

where both t and i are Boolean spectra.

Generally, each Boolean spectrum s can be represented as

$$s = t * i + \epsilon - \lambda. \tag{9}$$

In [7] Tanguiane shows how to find this representation for any Boolean spectrum s minimizing, at the same time, the components ϵ and λ. This issue, however, is too wide to be presented here in detail.

4 Chord Recognition Process

We assume that the input data for the recognition process is a sequence of signal samples representing a sound of one or several notes played on any musical instrument. The identification of notes occurring in the signal will be realized in several stages (see Fig. 3):

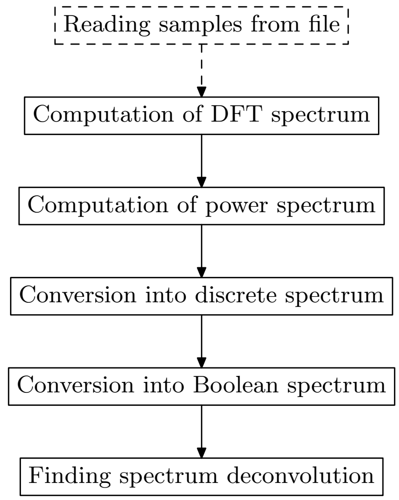

Fig. 3. Chord recognition stages

1. The first step of the recognition process is the frequency analysis. For computing the DFT spectrum of the input signal we use the *fast Fourier transform* (FFT) algorithm [3, 4, 5]. The computation of the N-point DFT directly from the definition (see Eq. 2) requires $\mathrm{O}(N)$ complex multiplications whereas the usage of the FFT algorithm reduces the complexity of computation to $\mathrm{O}(N \log_2 N)$. The only disadvantage of the FFT algorithm in comparison with the classical DFT algorithm is the requirement that the number of input samples must be the whole power two.
2. For the obtained DFT spectrum we compute the corresponding power spectrum and represent it using a normalized decibel scale (see Eq. 4).
3. The next step is to convert the power spectrum into the form of a discrete spectrum with the simultaneous rescaling of the frequency axis (see Eq. 7). According to the applied resolution of the discrete spectrum its bands correspond to wider or shorter frequency ranges. As the value of the discrete spectrum in the given band the highest value of the power spectrum belonging to this frequency range is taken[3]. As the lowest note which is to be recognized we took the note C_2. Thus the middle of the band with the index 0 is equal to the fundamental frequency of the note C_2 (65.406 Hz).
4. The discrete spectrum[4] of the signal power is then converted into a Boolean spectrum. From the definition of the Boolean spectrum s associated with the discrete spectrum S (see Eq. 6) we have

[3] The other approach may be based on estimation of the signal power in the given band.

[4] Because of the applied normalized decibel scale this spectrum does not completely satisfy the definition of the discrete spectrum where the value 0 denotes the lack of signal partial in the given band. Conformability to the definition may be achieved by an appropriate rescaling of the spectrum values.

$$s(n) = \begin{cases} 0 \text{ if } S_n = 0, \\ 1 \text{ if } S_n \neq 0. \end{cases}$$

In practice, the Fourier transform signals the presence of partials (which are possibly weak but always with power above zero) in each frequency band. This fact does not allow us to apply the above definition directly. The simplest way of overcoming this inconvenience is to remove all spectrum partials which lie below a certain threshold. After this operation, according to the definition of Boolean spectrum, we replace the remaining partials with unit values.

5. The last step is to find the deconvolution $t*i+\epsilon-\lambda$ of the Boolean spectrum. The interval distribution i, which we obtain as a result, is the answer to the question about the number and pitches of notes employed in a chord. Determining the notes' pitches we interpret interval values in the spectrum i in relation to the first partial of the spectrum t.

5 Examples of Recognition

Tables 2, 3 and 4 show the results of the recognition of three chords: (C_4, E_4, G_4), (C_4, E_4, G_4, $A_4^\sharp$), (C_4, E_4, G_4, A_4, D_4), played on the piano, organ and flute respectively. The recording parameters were the same as in the case of CD-Audio, i.e. 16-bit samples were taken at 44,100 samples/s. The length of the input sequence was equal to 2048 samples which corresponds to about 50 ms. Before the calculation of the FFT (see step 1 in Sect. 4) the input samples were first *centered* and *windowed* using the Hanning window [3, 4, 5]. The tables show the answer of our chord recognition system only for those threshold values (see step 4 in Sect. 4) for which the result of the recognition of one of the chords had changed. The misrecognized notes are marked in bold.

The results shown allow us to make some vital observations. A simple prediction to make was that the correctness of recognition significantly depends on the number of notes employed in a chord. As the number of simultaneously sounding notes

Table 2. Example of chord recognition (piano)

dB	C_4, E_4, G_4	C_4, E_4, G_4, $A_4^\sharp$	C_4, E_4, G_4, A_4, D_5
0	G_4	G_4	D_5
-1	G_4	G_4	G_4
-2	G_4	G_4	G_4
-5	C_4	G_4, $\mathbf{A_4}$	G_4, D_5
-6	C_4, C_5	C_4, C_5	G_4, D_5
-7	C_4, C_5	E_4, $\mathbf{A_4}$, E_5	G_4, D_5
-8	E_4, $\mathbf{F_4}$, E_5	C_4, $\mathbf{F_4}$, C_5	E_4, G_4, D_5
-9	*correct*	C_4, $\mathbf{C_4^\sharp}$, C_5	$\mathbf{F_4^\sharp}$, G_4, D_5, G_5
-10	*correct*	C_4, $\mathbf{F_4}$, $\mathbf{G_4^\sharp}$	C_4, $\mathbf{F_4^\sharp}$, A_4, G_4, D_5
-12	*correct*	C_4, $\mathbf{F_4}$, $\mathbf{G_4^\sharp}$	E_4, G_4, C_5
-13	*correct*	*correct*	E_4, G_4, C_5
-14	C_4, $\mathbf{C_4^\sharp}$, C_5	C_4, E_4, $\mathbf{F_4^\sharp}$, $A_4^\sharp$	E_4, G_4, C_5

Table 3. Example of chord recognition (organ)

dB	C_4, E_4, G_4	C_4, E_4, G_4, $A_4^\sharp$	C_4, E_4, G_4, A_4, D_5
0	G_4	G_4	A_5
-1	C_4	C_4	E_4, G_4, D_5
-2	E_4,E_5	E_4, E_5	*correct*
-3	E_4,$\mathbf{F_4}$,E_5	C_4, C_5	C_4, E_4, D_5
-4	E_4,$\mathbf{F_4}$,E_5	C_4, C_5	C_4, A_4, D_5
-5	C_4,$\mathbf{C_4^\sharp}$,C_5	$\mathbf{C_4^\sharp}$, G_4, $\mathbf{A_4}$	C_4, A_4, C_5, D_5
-6	*correct*	C_4, E_4, G_4	C_4, E_4, G_4, A_4
-8	*correct*	*correct*	C_4, E_4, G_4, A_4
-9	*correct*	*correct*	*correct*
-10	*correct*	*correct*	$\mathbf{C_4^\sharp}$, E_4, $\mathbf{C_5^\sharp}$
-12	$\mathbf{A_3^\sharp}$, $\mathbf{C_4^\sharp}$, $\mathbf{F_4^\sharp}$	$\mathbf{A_3^\sharp}$, D_4, $\mathbf{F_4}$, $\mathbf{G_4^\sharp}$	$\mathbf{C_4^\sharp}$, E_4, $\mathbf{C_5^\sharp}$
-14	$\mathbf{A_3^\sharp}$, $\mathbf{D_4}$, $\mathbf{F_4}$	C_4, E_4, $\mathbf{F_4^\sharp}$, $A_4^\sharp$	C_4, $\mathbf{C_4^\sharp}$, A_4, C_5, D_5

Table 4. Example of chord recognition (flute)

dB	C_4, E_4, G_4	C_4, E_4, G_4, $A_4^\sharp$	C_4, E_4, G_4, A_4, D_5
0	G_4	G_4	D_6
-1	G_4	$A_4^\sharp$	D_6
-3	G_4	G_4	D_6
-4	G_4	G_4, G_5	A_5
-5	G_4	G_4, G_5	C_4
-6	E_4, E_5	G_4, $\mathbf{A_4}$, G_5	E_4, E_5
-7	E_4, $\mathbf{F_4}$, E_5	$\mathbf{F_4}$, G_4, $A_4^\sharp$,E_5	E_4, E_5
-8	E_4, $\mathbf{F_4}$, E_5	E_4, G_4	E_4, E_5
-10	E_4, G_4	E_4, G_4	C_4, $\mathbf{C_4^\sharp}$, C_5
-11	E_4, G_4	E_4, $\mathbf{F_4}$, $\mathbf{B_4}$	C_4, A_4, D_5
-12	E_4, G_4	E_4, $\mathbf{F_4}$, $\mathbf{G_4^\sharp}$, E_5	C_4, A_4, C_5, D_5
-13	E_4, G_4	E_4, $\mathbf{F_4}$, $\mathbf{G_4^\sharp}$, E_5	C_4, A_4, C_5, D_5

is increased, it becomes more difficult to recognize them accurately. There are two main reasons for this. Firstly, the increased number of strong harmonic partials in a sound signal makes the problem of spectral leakage in FFT-based spectrum analysis more noticeable [3, 4]. This problem could be reduced by calculating the FFT for more samples of the signal, but as previously mentioned in Sect. 1.3, it would decrease the time resolution in the whole process of acoustical recognition. Secondly, the increased number of harmonic partials in the signal causes an increase of the *complexity* (number of partials) of the Boolean spectrum (see step 4 in Sect. 4). Consequently, this reduces the effectiveness of the algorithm for finding the deconvolution of the Boolean spectrum (see step 5 in Sect. 4). Briefly, with the increased complexity of the Boolean spectrum it becomes increasingly likely that one or a subset of its partials fit more than one note at the same time.

The analysis of the above results leads us to make another observation. The correctness of recognition depends not only on the number of simultaneously sounding notes but also on the kind of instrument on which they were played. The origin for this phenomena is, however, the same. Musical instruments differ in the number and the relative amplitudes of the harmonic partials in their sound. This is particularly apparent with the sound of the organ. Here the only harmonic partials which appear have a distance of one octave between them [6]. Thus, the complexity of its spectrum is much smaller in comparison to other musical instruments researched, which visibly makes recognition easier.

6 Conclusion

Analyzing the properties of the proposed solution to the problem of simultaneously sounding note identification we performed a number of tests taking account of different initial conditions such as the kind of musical instrument, the number of simultaneously sounding notes and the number of signal samples. The issue which still requires a solution is an appropriate choice of the threshold value (or values) used to generate the Boolean spectrum. As a result of analyzing the answers of the recognition process for several threshold values, it is possible to accurately identify the whole chord even if we did not get the correct answer for any of these values. In view of this the results obtained, thus far, seem to be very promising, indeed.

References

1. Berg RE, Stork DG (1995) The Physics of Sound. Prentice-Hall Inc., New Jersey
2. Bregman AS (1990) Auditory Scene Analysis: The Perceptual Organization of Sound. M.I.T Press, Massachusetts
3. Lyons RG (1997) Understanding Digital Signal Processing. Addison Wesley Longman Inc.
4. Oppenheim AV, Schafer RW (1989) Discrete-Time Signal Processing. Prentice-Hall Inc., New Jersey
5. Oppenheim AV, Willsky AS with Nawab SH (1983) Signals & Systems. Prentice-Hall Inc., New Jersey
6. Szlenk M (2000) The Automatic Identification of the Properties of Acoustical Space from the Sound Generated by Musical Instruments (in Polish). MS Thesis, Warsaw University of Technology, Warsaw
7. Tanguiane AS (1993) Artificial Perception and Music Recognition. Springer-Verlag, Berlin
8. Intel Integrated Performance Primitives, Reference Manual, vol 1: Signal Processing. http://developer.intel.com
9. The Complete MIDI 1.0 Detailed Specification. http://www.midi.org