FACTA UNIVERSITATIS
Series: **Electronics and Energetics** Vol. 29, N^o 1, March 2016, pp. 139 - 149
DOI: 10.2298/FUEE1601139S

THE IMPLEMENTATION OF SIGNAL ANALYSIS IN JAVA TO DETERMINE THE SOUND OF HUMAN VOICE AND ITS GRAPHICAL REPRESENTATION IN STANDARD MUSIC NOTATION

Patryk Solecki, Wojciech Zabierowski

Lodz University of Technology Department of Microelectronics and Computer Science,
Poland

Abstract. *The article presents the problems associated with signal processing in the human voice analysis. Based on the specific implementation of algorithms defining the human voice pitch, paper is shows the result in the form of a standard music notation with treble and bass keys on the stave. Particular attention is paid to the performance of the algorithms used for their implementation in Java. The Basic analysis of human voice signals is not a challenge, in general, but its implementation on mobile devices like smartphones with their limited hardware resources remains a challenge. The limitations of both, the CPU and the memory, affect the processing speed of the Java virtual machine. One should remember that the quality of microphone used in this type of mobile devices is low. From this point of view we have presented the new approach to the well-known problem of signal analysis implemented in computer applications such as Raven.*

Key words: *Signal processing, java programming, music notation, voice analysis, java*

1. INTRODUCTION

The sound pitch generated by the music instrument, was analyzed in Ref. [1] on the example of the smartphone application, which used phone's limited resources. In addition to the limitations associated with hardware (CPU, memory, bandwidth of the microphone) the selection and use of the appropriate programming language had a significant impact on the application. The problem with the choice of the language is that it is not just a matter of habit but it is very often determined by the selected hardware platform. It seems to be obvious that the most effective are the C + + type languages, especially in a situation where the sound analysis is done "online" on the data from the microphone.

Analysis of the sound pitch is one of the easiest issues related to the processing of acoustic signals. Therefore, an implementation of algorithms for the identification of the human voice or the sound of an instrument [1] is an often undertaken problem.

Received March 27, 2015; received in revised form October 21, 2015

Corresponding author: Wojciech Zabierowski
Lodz University of Technology Department of Microelectronics and Computer Science, Poland
(e-mail: wojciech.zabierowski@p.lodz.pl)

In the literature there are different approaches to the signal analysis. Various methods are used for speech recognition and the sound pitch or speech pitch determination. One example might be the use of adaptive algorithms in human speech segmentation [2]. The various approaches to the problem can be found in [3], where different methods of segmentation in automatic speech recognition are shown. One of the common problems in speech recognition is also a discrete wavelet transform used to identify the pitch and the segmentation of the signal [4]. An important aspect of speech recognition is to take into account the impact of individual features of the speakers and the signal transmission conditions on the issue of automatic speech recognition [5]. Another one is the Cepstral analysis, necessary for the continuation of the speech recognition not only in terms of its pitch, but also individual sounds [6].

The purpose of this publication is to show that for some simple problems of sound analysis, in particular its pitch determination, it is possible to create an effective implementation in Java, using the Java virtual machine and available input/output libraries.

The intention of the authors was not to propose commercial applications, having a possibility to create complete music notation from any of the songs. There are simple such applications, e.g. for an IPhone, as well as for desktops, like Raven. Some of them, especially those for smartphones, at the time of this research, were not yet available. The aim of this research was to show that using fairly simple, publicly available Java mechanisms, applicable to the various systems, including e.g. Symbian on mobile phones, it is possible, despite the hardware limitations (among others, of the phone's microphone) to create this type of application. The simplifications and consideration of only certain ranges in the field of signal analysis are intentional and aim to produce a good presentation of the implementation for human speech-signal analysis on the devices with limited hardware resources.

2. SPECIFICATION OF THE SOUND PROCESSING PROBLEM

Assuming, that the current version of Java can deal with the issue of sound analysis, it was decided that a certain pitch, for a better visual effect and clarity, will be presented in the form of the traditional musical notation on a stave. From the algorithmic point of view, DFT (Discrete Fourier Transform) analysis should be used for the bands of the spectrum of the analyzed signal [7]. The algorithm should recognize the fundamental tone band number and transform calculated frequency into the corresponding notes in the standard musical notation. The implementation also considered the use of the FFT (Fast Fourier Transform) algorithm. With its use one can save a considerable amount of calculation in comparison to the direct implementation of the DFT. The complexity of the model is described first for the case of the FFT algorithm. The computational complexity for both algorithms is given by the equations:

$$k_{DFT} = \mathrm{O}(N^2) \tag{1}$$

$$k_{FFT} = \mathrm{O}(N * \log_2 N) \tag{2}$$

where N is the number of the input data.

With proper implementation, the memory load is not much larger than the amount of memory occupied by the input data. In the case of real samples, the process of FFT can be further improved by using a modification of the algorithm, the 2N-point Real FFT, which further reduces the amount of needed resources. Unfortunately, apart from the benefits, the applied algorithm has also a drawback: the number of samples increases to 2N, which, in turn, adversely affects the flexibility of the analysis of the sound. The final results are obtained only at the end of the execution of the algorithm. These advantages and disadvantages of considered solutions have a significant influence on implementation, in particular when considering the operation of the system in the "on-line" version.

Basic DFT algorithm has two main inherent advantages. In contrast to the FFT, the results for various bands are evenly spaced in time. The second advantage, not to be underestimated, is the flexible amount of the input samples. This means that by regulating the number of samples used in the algorithm one can control the resolution, identified as f_r (Equation 3).

$$f_r = \frac{f_s}{N} \tag{3}$$

where f_s is the adopted sampling frequency.

In the case of the FFT, a large number of input samples should be used, which has an effect on the application performance. During the online analysis, the signal is analyzed constantly. In the case of the offline analysis only specific samples are analyzed.

In the case of the "online" processing (samples are processed in real time and immediately presented by applications in standard music notation), the extended period of the sampling time increases the inertia of the system - the implemented application. There are delays in the graphical presentation of notes. Assuming the standard f_s = 44100 Hz, we chose the following numbers of samples - powers of two (Equation 4).

$$N_i = (\{8192,\ 16384,\ 32768,\ 65536\}) \tag{4}$$

Too small number of samples, as in the case of N = 8192, results in a very low resolution. In this case, f_r = 5.38 Hz, on the basis of Equation 3. Modified 2N-point FFT algorithm, that is used, determines the use of at least N = 16384 samples, which can significantly affect the data acquisition time for the next step of calculation.

Out of the above discussion, the following conclusions can be derived with respect to the applicability of specific algorithms for the problem under consideration. The FFT is a faster, but more complex implementation, which requires more resources. The DFT algorithm in the basic version with a simple implementation has fewer hardware requirements, which in turn, gives the programmer more possibility to adapt the implementation to the limited resources.

Calculating the FFT of the significant harmonics requires calculation of the whole FFT, or in other words, calculation of all the harmonics, which in the given example requires calculating at least 8820 samples every time (because f_s = 44.1kHz).

The experimentally used algorithm has the calculation complexity stated below:

$$O = N^2 \cdot \left(\frac{900}{f_s} \right) \tag{5}$$

Complexity graphs of the FFT, DFT and the experimental DFT for the analyzed ranges of 500Hz and 900Hz are shown in Figure 1.

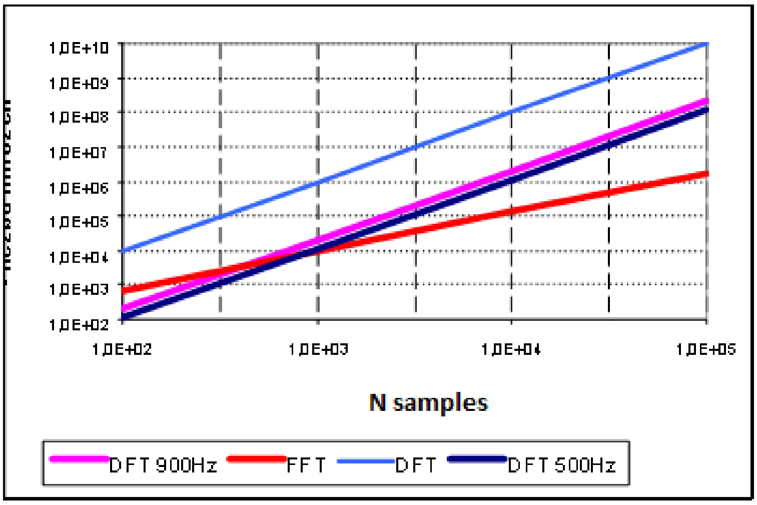


Fig. 1 Graph of the calculation complexity: computing time relative to the number of samples.

Specifying the fundamental pitch is a useful tool in the analysis of musical sounds. It allows you to specify the frequency of the fundamental tone, and on this basis determine the name of the sound. It also helps to examine such traits sound like *vibrato*. The frequency of the fundamental tone is in the intervals shown in Table 1. [5]:

Table 1 Frequency of the fundamental tones depending on the voice type.[10]

Voice name	Frequency [Hz]
Bass	8—320
Baritone	100-400
Tenor	120-480
Alto	160-640
Mezzo-soprano	200-800
Soprano	240-960

In addition, it is varied depending on the individual characteristics and is appropriate to the resonators: laryngeal, sinus, mouth and pectoral (chest), participating in creating a sound [10]. That is why, among other things, for the purpose of the research we decided to limit the frequency range to 1kHz. This restriction was introduced, because it was assumed that sounds will be read and written with the musical notation only in this range of frequencies. The human speech spectrum includes frequencies from 100 Hz to over 8 kHz, where the largest spectral density (energy) is in the vicinity of 500 Hz and gradually decreases with increasing frequency, which also supports our limitation of the analyzed interval.

The human ear receives signals in a much wider frequency range, but it is limited depending on individual human being. The typical range of signals recorded by the human ear covers frequencies from 20 Hz to 15 kHz (sometimes 20 kHz) and the highest sensitivity is from 1kHz to 3 kHz [16].

3. Solving the Problem – the DFT Analysis

Analysis of the human voice induces a lot of problems. The human voice is very complex in terms of number of parameters describing it [2,3,12,14]. This also results in a very complex set of harmonics visible in the spectrum of the signal. Changes in the voice can occur dynamically during the analysis, because of the conscious subject's voice modulation, but also by the impact of external factors that could affect the image of the spectrum of the voice of the tested person [10]. Although, the voice of every human is determined appropriately to the personal sound produced by vocal folds, but as a result of changes in the voice path it can vary considerably. This means that the voice of each person will be different due to the inter-individual characteristics, although it will still have the same pitch or the same character.

With the DFT analysis based on transformed expressions (Equation 6) one must be aware of certain characteristics of such signals, which facilitate further analysis and can prevent errors.

$$X(m) = \sum_{n=0}^{N-1} x(n) \cdot [\cos(2\pi \cdot n \cdot m / N) - j \cdot \sin(2\pi \cdot n \cdot m / N)] \tag{6}$$

Attention should be drawn to the following points:

- The fundamental tone is not always the most powerful component of the sound.
- Quality of used equipment is of fundamental importance and has an impact on the resolution and the possible disruption of the spectrum at low frequencies.
- If in the DFT without windowing the input data is used, fundamental tone may be disturbed with other bands, so-called "leaves".
- For a proper analysis of the signal and, in particular, of a human voice, the signal strength must be at the right level for the appropriate resolution of the harmonics, which allows for proper analysis.

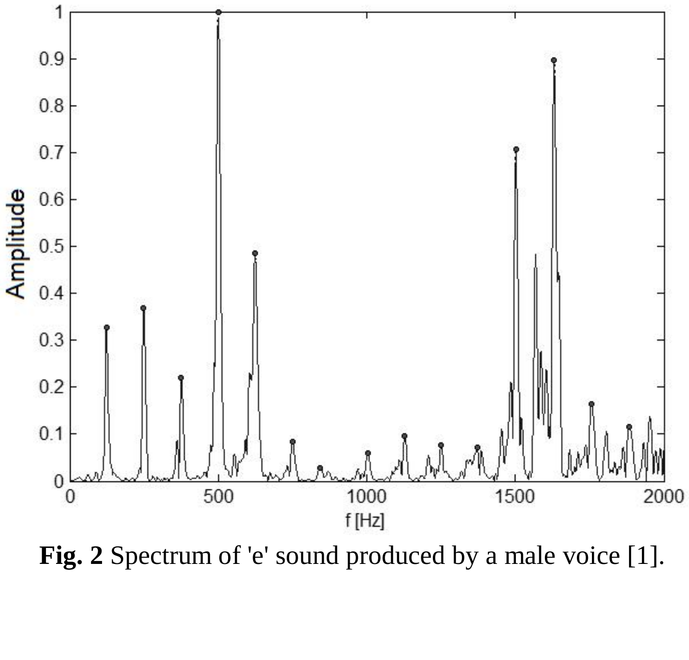


Fig. 2 Spectrum of 'e' sound produced by a male voice [1].

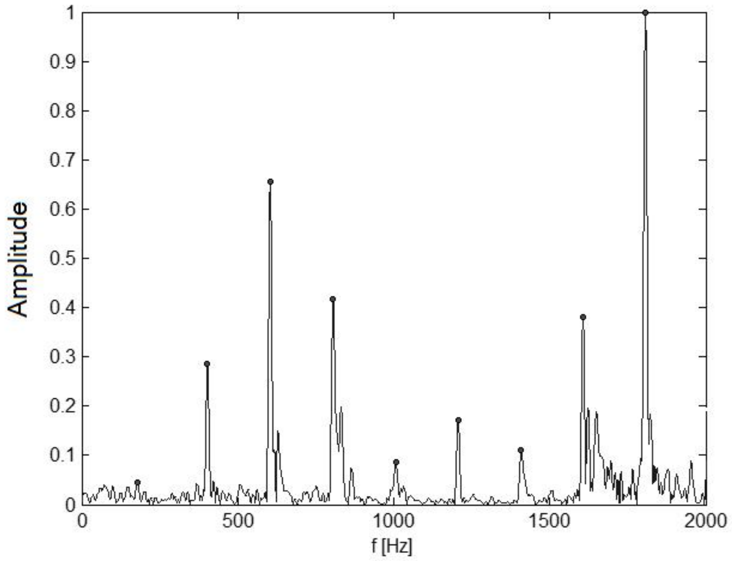


Fig. 3 Spectrum of 'e' sound produced by a female voice [1].

As shown in Figures 2 and 3, the frequency range of the human voice, in the sense of the fundamental tone, already starts in this particular case around 60-70Hz, and ends just over 1000 Hz. Taking into account the fact that the lowest and the highest frequencies are achieved only by a small percentage of the population for the tests and the described implementation, for the simplification, the narrowing of the scope was adopted in the range f = 98.00 Hz - 783.99 Hz. This is due to the need to ensure the appropriate resolution and frequency values assigned by equally tempered scale. The scale of the difference between the sounds was at the level df = 5.83 Hz. This means that the adoption of the resolution of f_r = 5 Hz would be sufficient for the most of the range. Unfortunately, the specificity of the mathematical properties of the Fourier transform can introduce errors for low frequencies, so-called "leaks". Correctly adopted resolution (see above) implicates a certain behavior of the variants of the algorithm. With such resolution of the DFT procedure components with the pitch similar to the observed spectral bands is transferred to adjacent spectral strips, which can result in having two neighboring strips of similar values. The interpretation of such strips depends on the applied algorithm. Either one chooses the strip of larger value as the basis for a sound diagnosis or approximating neighboring bands identifies the maximum to assign the sound pitch to it. At the resolution f_r = 5 Hz it is possible to recognize extreme low sound pitch on the basic level. In this way we limit also the DFT resolution and for the calculations, according to formula (1), 8820 samples may be used. With such resolution it is possible to reduce the number of samples while maintaining the basic, satisfactory sensitivity of the pitch markings. The algorithm adopted in the analysis was narrowed to five strips. The limitation was adopted on the basis of the signal analysis and the observation that for the purposes of pitch recognition this limitation gives such advantages in terms of utilization and load on system resources that we can tolerate the potential inaccuracies in the designation of the pitch.

For the proper operation of the described algorithm, the following assumptions were made:

- Two adjacent bands on both sides must have a smaller value.
- The analyzed band must exceed the value established in advance.

Please be aware of the issue of simplifying assumptions. One can, of course, expand the algorithm described, but these changes will affect the computational complexity, and this, in turn, will decrease the processing performance of the solution. The presented algorithm is based on a single pass, which implies that the time needed to find a correct tone varies for different tones. The analysis continues from low to high frequencies and, therefore, the lower sound is detected earlier and it will be marked sooner.

4. The Application

A Multi-threaded application was written, using [8], so that the described algorithms work efficiently. The project was split into groups of classes responsible for the analysis of samples and class group presentation, as well as the input/output operation to receive samples. Control group classes provide communication between the groups and between the threads. Collection and processing of data is carried out and can be controlled by the user. Data are collected directly from the buffered sound card stream, and then are subjected to normalization.

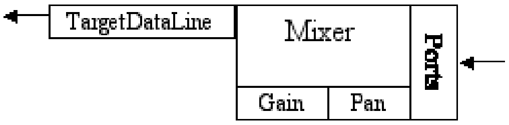


Fig. 4 A simple line-in configuration [11].

Java libraries provide a mechanism for downloading directly from the line-in standard audio mixer of the operating system (Figure 4).

The analysis described in the previous section is carried out with the prepared data to search for the fundamental tone. The effect of this thread is delivered to the thread responsible for the presentation by means of the music notation (Figure 5). Saving the marked tone should be taken into consideration as well as special characters like flat and sharp symbols, which lay down the increase and decrease of the displayed note one halftone. Although the frequency range of the human voice is not large, for the proper presentation of tones the key treble and bass should be used. The application has several features to enable various options for sound analysis.

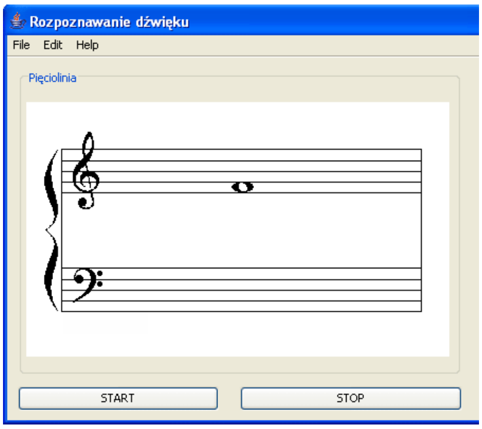
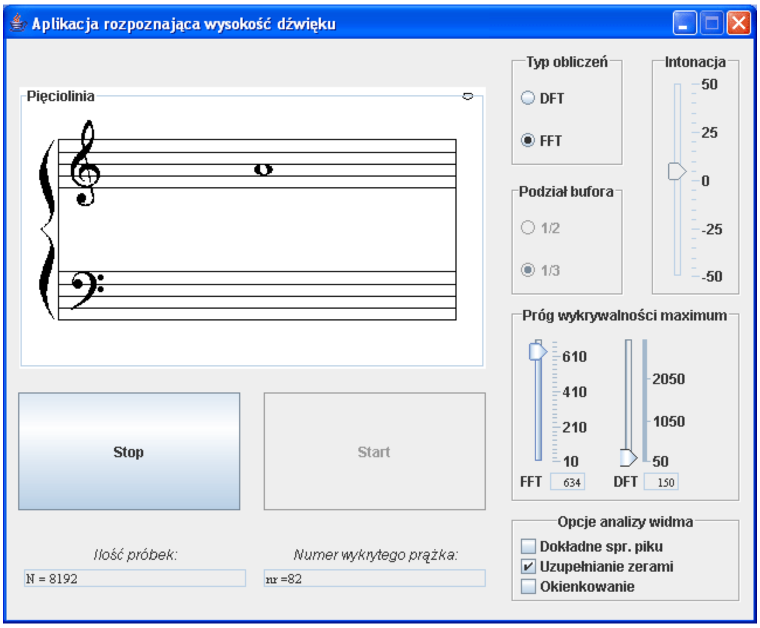


Fig. 5 The fragment of the dialog window (simplified) - presentation of the sound [11]

To generate the appropriate notes in musical notation JavaSwing package was used. In addition, applications have been introduced to allow control and algorithm modification by the user (Figure 6). The user can change parameters and select the method of analysis. It is possible to choose how the data acquisition and data analysis should be performed. These additional features allow one to show a different aspect of the application functioning.


Aplikacja rozpoznająca wysokość dźwięku
Pięciolinia
Stop
Start
Intonacja
50
25
0
-25
-50
Próg wykrywalności maximum
610
410
210
10
2050
1050
50

Typ obliczeń
○ DFT
● FFT

Podział bufora
○ 1/2
● 1/3

FFT 634
DFT 150

Opcje analizy widma
□ Dokładne spr. piku
☑ Uzupełnianie zerami
□ Okienkowanie

Ilość próbek: N = 8192

Numer wykrytego prążka: nr =82

Fig. 6 The fragment of the dialog window - presentation of the sound [11].

Before beginning the sound pitch analysis, the size of the incoming data buffer (samples) being sent to the analyzing functions must be set in the application window. This setting determines the number (*N*) of the samples analyzed in one process. This number is shown in the lower-left corner of the application's dialog window. The number of currently being identified stripe is shown on the right. This allows fast real time check of correctness of the results. There is also the intonation indicator in this corner. It shows if the identified pitch is below or above the standard frequency of the identified sound. During the calculation process the type of the calculations can be changed at any time by choosing between the FFT or the narrowed DFT. To assure correct identification of the tone of the sound by the program the calibration of the traceability threshold is necessary. The possibility to control this parameter allows adjusting the program to the level and timbre of the analyzed sound. This means scaling the levels of the analyzed parameters in order to avoid false identification of the tones. The use of the FFT, avoids leaflets (dicribed in the DFT section). While using the FFT, user has the possibility to enable the option of the thorough peak checking, filling with zeros in order to increase the resolution of the spectrum and windowing, in order to eliminate leaks (the, so called, side leafs or leaflets).

5. Smartphone Application

As already mentioned, the real challenge is to write the application for the mobile device, such as a smartphone, which is now already a very popular device accessible to everyone. Having experience in setting up sound for recording guitar tablature and guitar tuner implementations on a mobile phone [1], we decided to face a more complex challenge. On a smartphone we tested the results obtained for a relatively powerful machines: a desktop computer and a laptop. To increase the challenge, the application has not been tested on the latest models of smartphones, but on those 2-3 years old.

The implementation of the user interface especially required considerable changes in comparison to the desktop version. The touch screen instead of a mouse and a keyboard gives significant expansion of the functionality and usability of the application. However, as it was with the guitar tabulature creator solution [1], a serious problem was the mobile device's microphone. During the implementation of the application, the following additional assumptions have been adopted. First, the FIR (Finite Impulse Response) filter has been used, in addition, as an interpolating filter to increase digital sample rating. The main operation of data filtering is the convolution (Equation 7).

$$y(n) = \sum_{k=0}^{M} h(k)x(n-k) \tag{7}$$

where, M – number of input samples.

Thanks to this operation one can obtain better resolution of the guitar tuning. Dolph-Chebyshev window - also applied to the Final Impulse Response Filter - is very useful and corrects characteristic of window. Furthermore, characteristics of the filter and the window (such as the gamma parameter) can be set by the user. In addition, the autocorrelation function were introduced. Analysis of the autocorrelation function described below has been done:

$$r(n) = \sum x(m) \cdot x(m+n) \tag{8}$$

m – sample from the input range, n- sample number;

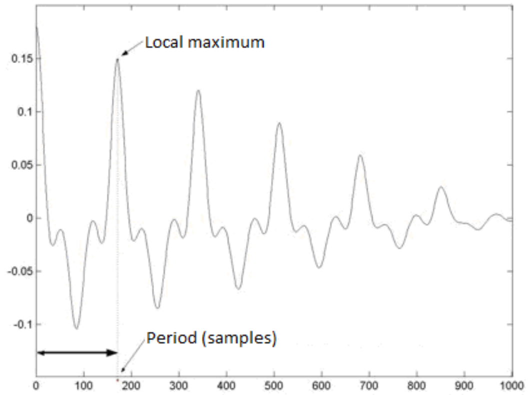


Fig. 7 Analysis of the autocorrelation function. Extraction of the fundamental harmonic [6].

This algorithm provides a good resolution, but it does not eliminate the noise. There can occur also problems if input signal does not include the fundamental harmonic:

- Analysis of the spectral function. The main goal of the spectrum function analysis is to find the peak of the function and establish current fundamental frequency of the input signal;
- Adaptation of Dolph-Chebyshev window (FIR filter). This type of window is very useful in case of creating the FIR filter;
- Approximation of the complex vector module.

Commonly used operation is arithmetic with complex numbers i.e.:

$$|V| = \sqrt{I^2 + Q^2} \tag{9}$$

I – real part of a complex number
Q – imaginary part of a complex number

It can be replaced with the simple low cost operation, which gives comparable result:

$$|V| = \alpha Max + \beta Min \tag{10}$$

where *Max* number is a larger part of the complex value and *Min* is a smaller part. Alpha and beta are the parameters, which are chosen from the appropriate table [7]. A standard sound sampling for mobile devices amounts to 8 kHz. This frequency is typical for voice calls and simple voice recording but sometimes it can cause difficulties if one wants to analyze the digital signal (sometimes the digital increase of the sample rating is required). For the purposes of this implementation, input signal frequency has been increased from 8 kHz to 80 kHz with digital interpolation process (sampling frequency has been increased). The disadvantage is the additional interpolation process to be executed by smartphone's resources but, on the other hand, we profit from the possibility of using anti-alias filter of lower order, which needs less processing power. The disadvantage is the possibility of the appearance of the noise and artifacts, because interpolation process is never able to reproduce the original signal accurately. The main profit is, that it increases the signal-to-noise ratio.

Summing up, by applying this procedure it became possible to obtain a good quality of the signal for further analysis and, finally, satisfactory results.

6. SUMMARY

It has been shown that Java programming language and the Java virtual machine, despite its limitations is able to process signals "online" in a satisfactory manner. New versions of the Java platform, as well as newer computers, significantly improve the comfort of the programmer, allowing more accurate analysis of increasingly complex computing problems. However, we must remember that today's challenge is not a desktop computer or even a laptop, but a mobile device [1]. Therefore, the issue of signal analysis for a variety of platforms, including Java is still valid.

This study may also have a practical aspect. Application of this type can be used for educational purposes, e.g. for learning signal analysis and related issues. It can also serve as an interesting addition to learning to recognize a sound pitch, which is a basic exercise for students learning to play an instrument or to sing.

The signal was collected online from a microphone, and analyzed according to the presented algorithm. No tests were carried out on the external databases, but only a few people (musicians) evaluated the application by listening, in terms of the quality of the voice recognition. Indeed, one could think of some more systematic way of checking the application, but it is worth noting that apart from checking by a few musicians, the application is used during learning, as an aid to teaching at the music school. This is good practical stress test. A detailed comparison with commercial applications has not been done because it was not the purpose of authors to compete with commercial applications for desktops, like Raven [17].

For these tests, the effect of recognition was satisfying. The obtained results show that the effect of our work may be useful for people studying playing on musical instruments, tuning the instruments, etc. The program was used with good results as a teaching aid for children learning music at music school.

REFERENCES

[1] P. Solecki, W. Zabierowski, *The signal analysis of sound based on the application of guitar tabulatures for mobile devices.* PRZEGLĄD ELEKTROTECHNICZNY, 2012, rocznik 88, nr 10b, p. 239-242.

[2] V. A. Petrushin, A*daptive Algorithms for Pitch-synchronous Speech Signal Segmentation*, SPECOM'2004: 9th Conference Speech and Computer St. Petersburg, Russia September 20-22, 2004.

[3] A. S. Spanias, "Speech coding: A tutorial review," Proc. IEEE, 82, 1541–1575, October 1994.

[4] Ch.Wendt, Athina P. Petropulu, *Pitch determination and speech segmentation using the discrete wavelet transform*, Electrical and Computer Engineering Department, Drexel University, Philadelphia PA 19104.

[5] P. Mrowka, *Algorytmy kompensacji warunków transmisyjnych i cech osobniczych mówcy w systemach automatycznego rozpoznawania mowy*, Politechnika Wrocławska, Instytut Telekomunikacji, Teleinformatyki i Akustyki, Raport Nr I28/PRE-001/07, phd dissertation, Wrocław 2007.

[6] A.P. Dobrowolski, E. Majda, *Analiza cepstralna w systemach rozpoznania mówców*, No 6/2012, Instytut Logistyki i Magazynowania, 2012.

[7] R. G. Lyons, Understanding Digital Signal Processing, PEARSON, 2010.

[8] B. Eckel, , 2003. *Thinking in Java*. Wydawnictwo "Helion", Gliwice.

[9] K. Demuynck, T. Laureys, A Comparison of Different Approaches to Automatic Speech Segmentation, http://www.esat.kuleuven.ac.be/#spch 2013.

[10] W. P. Morozow, *Isskustwo Rezonansnawo Pienija*, Iskusstwo i nauka. Instytut Psychologii Rosyjskiej Akademii Nauk, Państwowe Konserwatorium im. P.I. Czajkowskiego w Moskwie, Moskwa 2002.

[11] M. Dybowski, W. Zabierowski, 2005. *Aplikacja rozpoznająca wysokość dźwięków głosu ludzkiego JAVA – w mgnieniu oka*, XIII Konferencja SIS - Sieci i Systemy Informatyczne – teoria, projekty, wdrożenia, aplikacje, Łódź, p. 421-426, t. 2, Piątek Trzynastego Wydawnictwo 2005, ISBN 837415-069-6, 711 s., 2 t., 23,5 cm

[12] A. Gersho, "Advances in speech and audio compression", Proc. IEEE, 82, June 1994.

[13] Lawreace R.Rabiner, Ronald W. Schafer, *Digital Processing of Speech singlas*, Prentice-Hall, Inc.Englewood Cliffs, New Jersey 07632, Bell Laboratories 1978

[14] T. Robinson, *Speech Analysis*, Lent Term 1998

[15] W. Hess. Pitch Determination of Speech Signals. Springer-Verlag, 1983.

[16] D. Gerhard. *Pitch extraction and fundamental frequency: History and currenttechniques.* Technical Report TR-CS 2003-06, Department of Computer ScienceUniversity of Regina, Regina, Saskatchewan, CANADA S4S 0A2, november 2003

[17] http://www.birds.cornell.edu/brp/raven/RavenTestimonials.html 2013