Patryk SOLECKI, Wojciech ZABIEROWSKI

Technical University of Łódź, Department of Microelectronic and Computer Science, Wólczanska 221/223 90-924 Łódź

The signal analysis of sound based on the application of guitar tabulatures for mobile devices

Abstract. *These The paper presents application for mobile devices which provides real time guitar tunning and analysis of tabulatures. This functionality requires algorithms to extract harmonic components of sound. This application is destinated for Symbian operating system with S60 interface. It consists of two important parts: application core created in Symbian C++ and user interface dedicated for Symbian Qt which provides support for this platform with integration to the S60 framework.*

Streszczenie. *Artykuł prezentuje aplikację przeznaczoną dla urządzeń mobilnych, która zapewnia strojenie gitary w czasie rzeczywistym oraz zapis tabulaturowy. Aplikacja przeznaczona jest dla systemu operacyjnego Symbian S60. Składa się z dwóch zasadniczych części: rdzenia aplikacji zrealizowanego w technologii Symbian C++ oraz interfejsu graficznego dedykowanego dla Symbian Qt, który zapewnia wsparcie dla tej platformy i integrację z frameworkiem S60. Narzędzie to przesuwa granicę zastosowań smartfonów i stanowi alternatywę dla komercyjnych stroików gitarowych.* ***Analiza sygnałowa dźwięku na przykładzie aplikacji zapisu tabulatur gitarowych dla urządzeń mobilnych.***

Keywords: signal analysis, music, Qt, Symbian S60, guitar
Słowa kluczowe: analiza sygnałowa, muzyka, Qt, Symbian S60, gitara

Introduction

Guitar is one of the most known string instrument on the world. There are many applications which get the hang of guitar play. Nowadays very important thing is mobility and ergonomic usage of applications. Tool described in this paper includes these features. This application provides tunning and tabulatures analysis for electroacustic guitars. Tunning panel includes information to set appropriate frequency for all strings. Pictures give information about too low, high or correct frequency of currently chosen string. Tabulature analysis gives graphical information about pushed strings and guitar frets for all of them. Main task of application is digital signal processing with usage of embedded microphone.

Project specification

The Electroacustic guitar includes six steel strings with generated sound frequency from about 80Hz to 330Hz:

E6 – 82,4Hz,
A5 – 110,0Hz,
D4 – 146,8Hz,
G3 – 196,0Hz,
H2 – 246,9 Hz,
E1 – 329,6Hz.

Guitar tabulature is a method to describe guitar sound through the six lines which represents guitar strings.

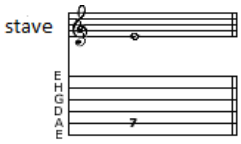

Fig. 1. Stave and guitar tabulature.

Number on the line should represent guitar fret (tone of currently pushed string).

Analysis of sound piece should have at least 0.5Hz accuracy in order to receive satisfying result of tunning. Mobile device should provide sample rating with 8kHz frequency or higher. Furthermore input signal frequency should be lower than 4kHz to avoid aliasing occurrence. Aliasing is a diversity of meaning input digital signal.

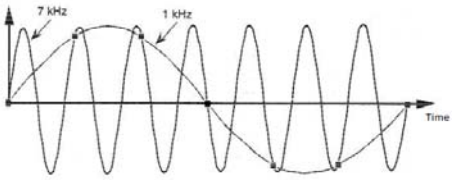

Fig. 2. Diversity of meaning input digital signal. Original input signal – 7kHz, sample rating f_s=6kHz.[1]

Symbian API provides many available sound formats. Main sound format applied to application is signed PCM, 16-bits. Symbian is dedicated to Little Endian architecture systems, commonly with ARM11 processors family. Main establishment is to implement few functionalities shown in Figure 3.

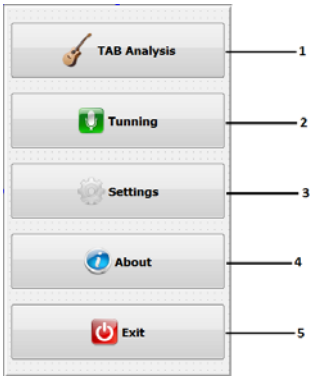

Fig. 3. View of main window. Available options: (1) analysis of guitar tabulatures; (2) guitar tunning; (3) settings of tunning and guitar tabulatures; (4) application description; (5) clousure main thread, application exit.

Tunning mode is a signal analysis which gives information about first harmonic. Tabulatures mode is an analysis of basic and next harmonics of sound. Everyone string has an unique collection of harmonics. Apart from that every sound of currently pushed string for all frets includes this collection (more info described in [1] position). That can be use to obtain numbers of guitar frets showed by dot screen of mobile device. Interaction with user runs by

[1] Richard G. Lyons, „Understanding Digital Signal Processing”

dot screen of small mobile device. There are two types of screen orientation – landscape and portrait. Tabulatures view is created only as a landscape view. It makes usage of application more ergonomic. Project consists of two main parts. First part is an application core created as loaded static library responsible for digital signal analysis. Core provides also data transmition between embedded microphone and algorithms. Application core is written in Symbian C++ language. This part should be effectively designed to provide quick responses and interaction with user interface. User interface is created by well known cross over Qt framework. UI part includes window manager which is responsible for view management and creating of dynamic animations.

Algorithms

Real time digital signal analysis of sound requires fast and efficient algorithms. Tunning process requires sample rating grather than standard 8kHz applied for mobile devices. In this case solution is digital signal increase of sample rating by interpolation of input signal. Other important operations are:

- 2N Fast Fourier Transform (2N-FFT). Fast Fourier Transform provides frequency analysis of input signal. 2N-FFT is a very efficient algorithm. In this case 2048 input samples gives resolution about 4Hz. Classic FFT gives redundand information if input data is real samples. In case of complex input samples Fourier transform it returns useful information on whole range of result. First step is to provide proper sorting of real input data and create complex samples. Afterwards we can calculate classic FFT. In the end to retrieve frequency analysis we should use additional activities, widely described in [4]. In this case 2N-FFT algorithm is more efficient than classic calculations of FFT.
- Interpolation of input signal - finite impulse response filter (FIR) as a interpolating filter to increase digital sample rating. Main operation of data filtering is convolution (1).

$$(1)\quad y(n)=\sum_{k=0}^{M}h(k)x(n-k)$$

M – amount of input samples

Figure 10 shows Interpolation's example of digital input signal. Thanks to this operation we can obtain better resolution of guitar tunning. Dolph-Chebyshev window - also applied to the Final Inpuls Response Filter - is very useful. Window corrects characteristic of window. Furthermore, characteristic of filter and window (such as gamma parameter) can be set by the user.

- Discrete Fourier Transform of data piece (DFT of small range area). DFT is much more slower operation than FFT but it is independent to the number of input data. DFT can be very useful if we want to receive piece of frequency analysis. Firstly FFT can give us information about whole range of frequency function and secondly DFT can describe particularly required piece. DFT operation is described by:

$$(2)\quad X(m)=\sum_{n=0}^{N-1}x(n)e^{-j2\pi mn/N}$$

Other equivalent figure is:

$$(3)\qquad X(m)=\sum_{n=0}^{N-1}x(n)[\cos(2\pi mn/N)-j\sin(2\pi mn/N)]$$

In both formulas "N" is amount of input samples and "j" is imaginary unit.

- Analysis of autocorrelation function described below:

$$(4)\quad r(n)=\sum_{m}x(m)\cdot x(m+n)$$

m – sample from input range

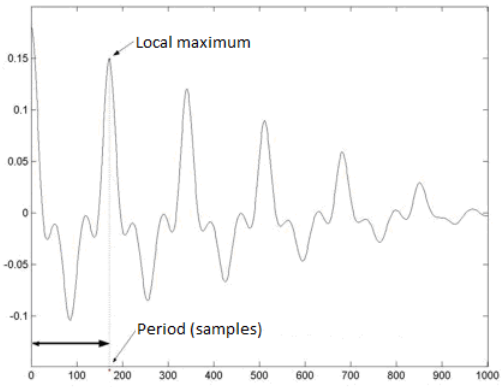

Fig. 4. Analysis of autocorrelation function. Extraction of basic harmonic

This algorithm provides good resolution but it's not proof of noises. There can occur also problems if input signal doesn't include basic harmonic;

- Analysis of spectral function. Main destination of spectrum function analysis is to find peak of function and ascertain current basic frequency of input signal;
- Adaptation of Dolph-Chebyshev window (FIR filter). This type of window is very useful in case of creating FIR filter. More info about Dolph-Chebyshev window can be obtained from appropriate websites[2].

- Approximation of complex vector module.

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

$$(5)\quad |V|=\sqrt{I^2+Q^2}$$

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

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

$$(6)\quad |V|=\alpha Max+\beta Min$$

where Max number is greater part of complex value and Min is lower part. Alpha and beta are the parameters which are chosen from appropriate table (more info described in [4] position).

Guitar sound can be considered in context of string characteristic. Different string deflection is commonly named as string mode (detailed info at [2]).

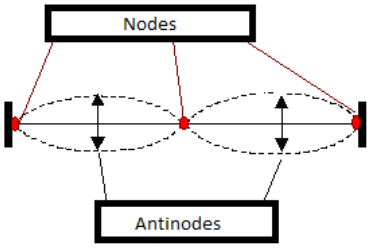

Fig. 5. String mode.

[2] http://www.dsprelated.com/dspbooks/sasp/Chebyshev_Polynomials.html

Successive length determines which harmonics components can be included in guitar sound.

Sometimes qt application is not able to process all events. This situation occurs in case of high costs of others operations. Events will be arranged in order but not considered by application. It can be resolved by QCoreApplication class and static processEvents method which causes triggering of mentioned events.

One of the most important issues is sound recognition. Standard sound sampling for mobile devices amounts to 8kHz. This frequency is typical for voice calls and simple voice recording but sometimes it can cause some difficulties if we want to analyze the digital signal (sometimes the digital increase of sample rating is required). S60 platform provides sound API (position [6]) which gives ability to operate embedded microphone of mobile devices with Symbian S60 on board. There are also many others available features such as sound codecs or sound formats. Quick review showed on chart below.

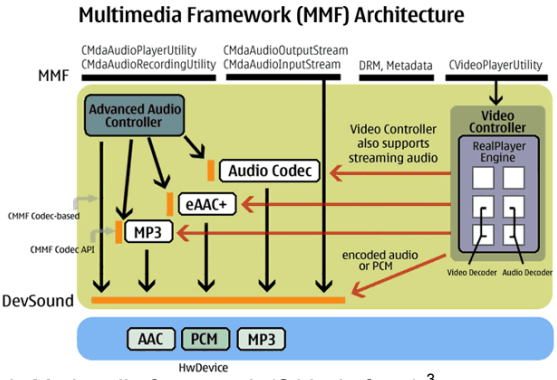

Fig. 6. Mutimedia framework (S60 platform).[3]

Main class which provides raw sound samples is CMdaAudioInputStream (typical supported sample rate for many devices is 8kHz, mono channel) – a part of Symbian Multimedia Framework [3].

Tools and technologies

This application is dedicated to Symbian S60 operating system. There are few available SDK (Software Development Kit). In this case S60 5th Edition SDK and Qt for Symbian by Nokia v4.7.1 (S60 Open source) is used. Carbide C++ environment which supports developer's work is also very useful. Provides wide range of plugins such as build-in Qt Designer to create windows of user interface.

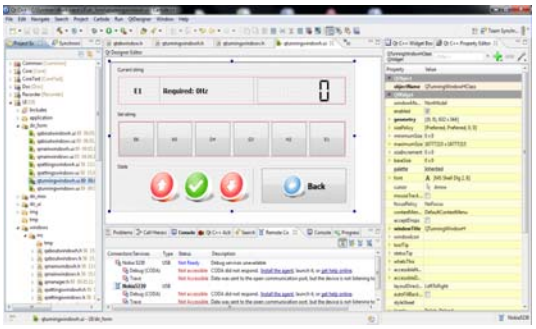

Fig. 7. Carbide C++ and build-in Qt Designer as plugin.

Each window can be created by drag and drop method. Thanks to Qt Designer we can automatically generate needed files such as main window view (*.ui, XML template), header, source files and resources file (*.qrc) which give access to local icons and pictures which can be added to main project.

```
<RCC>
  <qresource prefix="/res">
    <file>plus.png</file>
  </qresource>
</RCC>
```

Listing 1. Qt resources file. Example XML template.

There are also many other programs (see [5]) which support build process and creating of project:

uic (User Interface Compiler) – compiler of *.ui files. UI file describes appearance and content of view,

moc (Meta Object Compiler) – compiler which generates source files on example of header files content,

qmake – programme that manages compilation process, creates and updates „makefile" on example main project file (*.pro),

Qt Designer – build-in tool, Carbide plugin,

Qt Assistant – application which provides wide help and describes all Qt classes, supports work of developers, includes wide description of libraries, tools and examples,

others (Qt Creator, Qt Linguist).

Other useful program is StarUML which can be use by creating class diagrams. Thanks to StarUML[4] we can design complex and detailed schemas which supports implementation of applications.

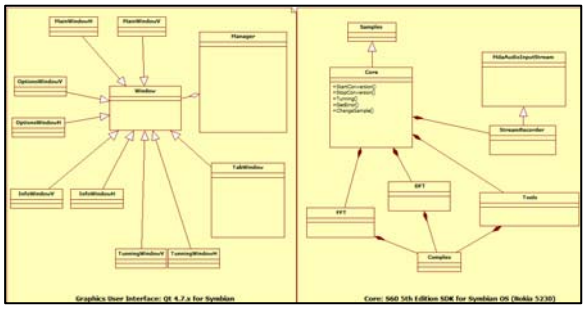

Fig. 8. Class diagram with usage StarUML.

Testing and development

Very important part of building the application is testing and improvement. Empirical knowledge gives ability to find the best solution in many cases.

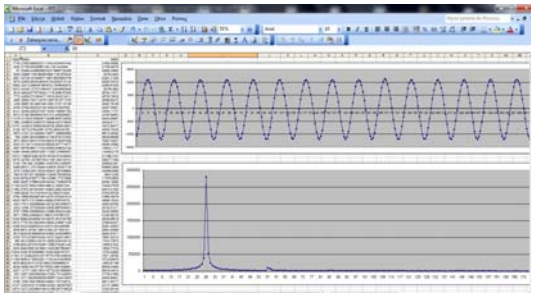

Fig. 9. Extraction of base frequency – Ms Excel

This application can be compiled in two independent modes. First release mode is dedicated for common user. The second one works slower but provides many informations about workflow, operation results and raw sound samples. This mode gives ability to save all the data

[3] http://www.developer.nokia.com/Community/Wiki/MMF

[4] http://staruml.sourceforge.net/en/

to file. All the informations can be imported i.e. to Ms Excel datasheet. This approach to problem provides possibility to design i.e. autocorrelation method, DFT, FFT or FIR solutions separately from implementation code.

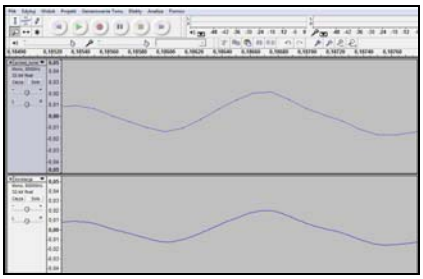

Fig. 10. Audacity tool. Example of two signals: 8kHz digital input signal and interpolated signal (digital increase of sample rate to 80kHz).

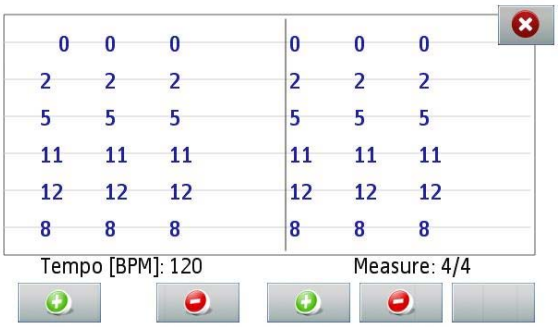

Fig. 11. Example of guitar frets (first version of UI, launched on emulator with “stub” core).

This method provides quick testing and improvements. We can show immediately the results of improved algorithms on real input data and adjust parameters. Visual Basic scripts can be also very useful tools to simulate all the data flow. At the end of design process we can implement the best solution in C++ code and show the result of output raw data i.e. in Audacity application.

Summary

Tabulatures record is a commonly used method to record guitar sound which provides ability to learn musical compositions without knowledge of score. Tabulatures record is applied to many commercial applications, such as Guitar Pro which supports work of musicians. This smart tool gives ability to acquaint user with basics of guitar tabulatures and provides fast tunning.

One of the most important disadvantages are low hardware and system resources of mobile devices thus application should implement increase of digital sample rating. Main advantage is also mobility of usage. Obtained result of guitar tunning is comparable with commercial tuners so this application can be very useful tool for all guitar players.

Wydanie publikacji zrealizowano przy udziale środków finansowych otrzymanych z budżetu Województwa Zachodniopomorskiego.

REFERENCES

[1] Musicology: The Key Concepts, David Beard and Kenneth Gloag
[2] http://www.physicsclassroom.com/class/sound
[3] http://wiki.forum.nokia.com/index.php/MMF
[4] Understanding Digital Signal Processing, Richard G. Lyons,
[5] C++ GUI Programming with Qt 4, Jasmin Blanchette and Mark Summerfield
[6] Quick Recipes on Symbian OS: Mastering C++ Smartphone Development, Michael Aubert.

***Authors**:*
dr inż. Wojciech Zabierowski, Politechnika Łódzka, Katedra Mikroelektroniki i Technik Informatycznych, E-mail: wojtekz@dmcs.pl; mgr inż. Solecki Patryk, E-mail: soleckipatryk@gmail.com