A Crossplatform ECG Compression Library for Mobile HealthCare Services

Alexander Borodin, Yuliya Zavyalova
Department of Computer Science
Petrozavodsk State University
Petrozavodsk, Russia
{aborod, yzavyalo}@cs.petrsu.ru

Abstract

Continuous monitoring of heart function can considerably decrease the percent of the sudden deaths caused by heart diseases since in this case there are a good chances that the help is rendered in time. Since in long-term heart examination huge amounts of data are generated a tool of effective data compression is needed. Recently several algorithms of biosignal compression were proposed by researchers. In this paper the results of adapting the algorithms based on wavelet transform to the constraints of mobile devices are presented. The crossplatform library implementing the ECG compression functions was constructed.

Index Terms: ECG, compression, mhealth.

I. Introduction

It is known that cardiovascular and ischemic heart diseases are the leading cause of death in the developed countries. Long-term continuous monitoring of heart function provides a way to discover the problem early and take timely measures to prevent sudden heart attack.

A variety of wearable devices are available in the market with the feature of electrocardiogram monitoring. These devices can be connected with smartphones and provide data for applications. According to forecasts of experts the smartphone applications will become the important part of mHealth solutions in the near future. The architecture of typical cardiomonitoring system based on smartphone use is presented in [1].

Nevertheless, long-term continuous monitoring leads to a problem of storing and transmitting huge amounts of electrocardiogram data. Recently a number of efficient compression algorithms have been developed for ECG signal processing. These algorithms need to be adapted for using in mobile applications.

This paper summarizes the project of compression library construction, which is suitable for electrocardiogram recordings encoding and at the same time convenient for use in mobile healthcare services.

II. ECG compression based on wavelet transform

Application of wavelet transform in compression algorithms of 1D and 2D signals has acquired a reputation for a good advantage, e. g. in JPEG 2000 [2] standard and in Dirac video codec [3].

Actually wavelet transform of a vector of values has the same size as the original one. But the distribution of values for the wavelet coefficients is usually centered close to zero and there are very few large coefficients. Consequently almost all the information is concentrated in a small fraction of the coefficients and can be efficiently compressed with an entropy coder.

Designing a computing algorithm for a small mobile device such as a smartphone we meet the following additional constraints.

1) An efficient implementation of the algorithm should be provided since huge and time-consuming computations can quickly discharge the battery.
2) Real arithmetics should be avoided if possible due to the lack of floating point unit on some devices.

The lifting scheme was proposed in [4] as an simple and efficient computation procedure for obtaining the wavelet coefficients. In short, the followings steps are implemented.

1) Splitting step: the original signal $x = \{x_i\}$ is splitted into two (odd and even) subsequences.

$$\begin{aligned} s^{(0)} &= \{x_{2i}\} \\ d^{(0)} &= \{x_{2i+1}\} \end{aligned} \quad (1)$$

2) Several repeats of lifting step: the odd and even subsequences are filtered by the prediction and update filters, $P_n(x)$ and $U_n(x)$.

$$\begin{aligned} d^{(j)} &= d^{(j-1)} - P_j\{s^{(j-1)}\} \\ s^{(j)} &= s^{(j-1)} + U_j\{d^{(j)}\} \end{aligned} \quad (2)$$

Diagram illustrating a lifting step is given in Fig. 1.

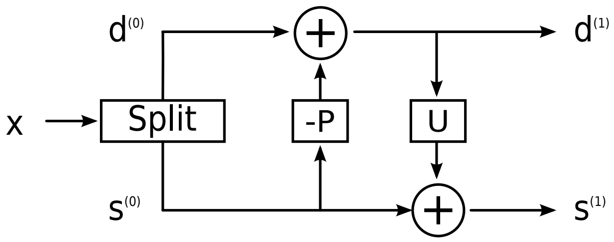

Fig. 1. Diagram of a lifting step

Denote the length of a vector as k. Then the filter have the following general form:

$$F_n\{x\} = \sum_{i=1}^{k} f_n(k) x_i \quad (3)$$

3) Normalization step: at last the coefficients are normalized. Denote the number of lifting steps as N. Then the normalization step can be expressed in the following form, where M_0 and M_1 are constant values:

$$\begin{aligned} d^{(N)} &= M_0 d^{(N)} \\ s^{(N)} &= M_1 s^{(N)} \end{aligned} \quad (4)$$

Since the ECG recording values are considered as integers then the wavelet transform that maps integers to integers is used. The integer lifting algorithm can be described with the following formulae:

$$\begin{aligned} d^{(j)} &= d^{(j-1)} - \lfloor P_j\{s^{(j-1)}\} + 0.5 \rfloor \\ s^{(j)} &= s^{(j-1)} + \lfloor U_j\{d^{(j)}\} + 0.5 \rfloor \end{aligned} \quad (5)$$

Despite a rounding, algorithm remains reversible.

Nevertheless this algorithm is still using floating point processing in P and Q operators computation. In order to completely implement a lifting scheme with integer arithmetic only the FIAWT algorithm proposed in [5] for image compression in nuclear physics applications is adopted.

In this approach each constant rational multiplier is represented by bit vector. Each multiplication is embodied with a series of bit shifts and additions.

One should notice that besides refusal of floating point computations, this algorithm also is extremely fast. Thereby both additional requirements are satisfied.

The popular Le Gall 5/3 [6] and Daubechies 9/7 [4] filter pairs used in JPEG 2000 compression are chosen to be implemented by the library core.

Lifting scheme for Le Gall 5/3 DWT is simple:

$$\begin{array}{rcccl} s_i & = & d_i^{(1)} & = & d_i^{(0)} - \frac{1}{2}(s_i^{0)} + s_{i+1}^{(0)}) \\ d_i & = & s_i^{(1)} & = & s_i^{(0)} + \frac{1}{4}(d_i^{1)} + d_{i-1}^{(1)}) = -\frac{1}{8}s_{i-1}^{(0)} + \frac{1}{4}d_{i-1}^{0)} + \frac{3}{4}s_i^{(0)} + \frac{1}{4}d_i^{(0)} - \frac{1}{8}s_{i+1}^{(0)} \end{array} \quad (6)$$

It is necessary to note that Le Gall 5/3 filter coefficients are rationals, moreover they are represented as binary numbers of finite length.

Recall lifting scheme for Daubechies 9/7 DWT:

$$\begin{array}{rcl} d_i^{(1)} & = & d_i^{(0)} + \alpha(s_i^{0)} + s_{i+1}^{(0)}) \\ s_i^{(1)} & = & s_i^{(0)} + \beta(d_i^{1)} + d_{i-1}^{(1)}) \\ d_i^{(2)} & = & d_i^{(1)} + \gamma(s_i^{1)} + s_{i+1}^{(1)}) \\ s_i^{(2)} & = & s_i^{(1)} + \delta(d_i^{2)} + d_{i-1}^{(2)}) \\ s_i & = & \zeta s_i(2) \\ d_i & = & \zeta^{-1} d_i(2) \end{array} \quad (7)$$

Daubechies 9/7 filter coefficients $\alpha, \beta, \gamma, \delta, \zeta$ are irrational numbers, therefore in implementation quantized values of the coefficients proposed in [7] are used:

TABLE I
QUANTIZED VALUES OF DAUB 9/7 DWT

Coefficient	Irrational	Quantized rational
α	$-1.58613434\ldots$	-1.5
β	$-0.0529801185\ldots$	-0.0625
γ	$0.882911076\ldots$	0.46875
δ	$0.443506852\ldots$	0.7998046875
ζ	$1.14960439\ldots$	-1.25

The actual compression is done by predictive arithmetic coder based on Markov chain [8]. The input is predicted one bit a time. The predicted bit is then coded using arithmetic coding. This algorithm is known to have a good compression ratio. Nevertheless, it has middling speed and also requires a lot of memory. Hence this algorithm will be replaced in future releases.

III. LIBRARY ARCHITECTURE AND OVERVIEW

The CardioZip library implements the ECG compression algorithms described above. The library is represented by the set of classes written in C++.

The structure of the library is shown on Fig. 2.

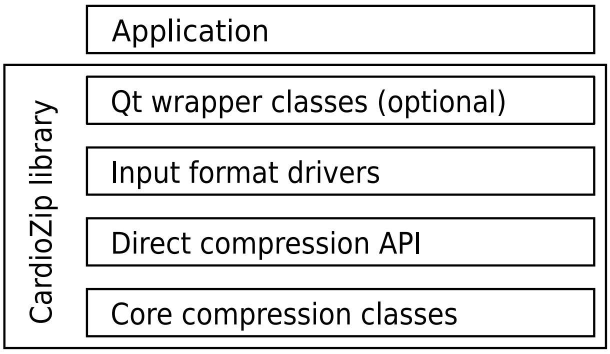

Fig. 2. CardioZip library architecture

At the bottom level core compression algorithms are presented. Implementations will probably be changed in next releases, hence they are not accessible directly from application.

The set of wrapper classes for public direct API is located above. It is not stable yet, but hopefully will be stabilized in release.

Since the library supports raw data channels with 1, 2 or 4 byte signed integer samples hence a set of input format converters are developed. Currently Alive stream data format produced by Alive Heart Monitor and some WFDB-compatible formats used in PhysioBank [**?**] archive of ECG recordings are supported. Custom input format converters can be implemented using direct API.

Qt wrapper classes are on top level of library architecture. They form an optional high-level API that can be used when Qt is supported. This API makes possible organizing the interaction between application and library in Qt style manner, e. g. using signals and slots concept.

IV. Compressed file format

Despite considerable interest to a problem of compression of biosignals, attention of researchers it is given exclusively to algorithms and their prototype implementations. At the same time the issues of software solution design, constructed on the based of these algorithms will be left in the basket.

For the purposes of storing and further processing compressed ECG recordings CardioZip file format is developed. It was designed to be easily extendable for possible implementation of new features.

File produced by CardioZip library consists of a set of headers and compressed ecg recordings section. File start with a four byte "zero header", containing a file signature "CZF" in ASCII and a one byte unsigned representing a number of bytes used to store a size of some memory block or an offset further referred to as *asize*.

All numbers which are more than one byte long are written in little endian byte order.

Next $2 \times asize$ bytes are allocated to a file header. This header stores two numbers: the size of the current file and the offset from the beginning of the file to the coded electrocardiogram data. In other words the second number represents the size of all headers.

The file header is followed by information header, containing technical details: number and types of channels, sample rate, bits per sample, algorithms used. Note, that data channels are not limited by ECG recordings only and a support of other kinds of biosignals can be added in the future. The information header starts with *asize* bytes field, containing the length of header itself.

All other headers are optional. If present the header starts with four byte signature ”CZF” plus header letter. The signature field is followed by *asize* bytes length of current header. Currently supported headers are listed in table II in appearance order.

TABLE II
OPTIONAL HEADERS OF CARDIOZIP FORMAT

Letter	Header	Comments
I	Integrity	Error correcting method and data, e. g. CRC
S	Security	Security related data, e. g. encryption method
P	Patient	Patient personal data, e. g. recording date and duration
A	Algorithms	Algotithms related data, e. g. Huffman coder table

Patient and algorithms header can be encrypted along with ECG recordings, the particular behavior is defined by security header.

V. CONCLUSION

The CardioZip library that can be used for the purposes of ECG signal compression on the base of wavelet compression is constructed. The library is aware of the constraints of mobile computing. Three levels of object-oriented API are avaiable: raw stream compression classes, common file format interfaces and QObject-based high-level interface.

Currently the project is in the testing phase. The library is released under the terms of the GNU Lesser General Public License 2.0 (LGPL 2.0).

ACKNOWLEDGMENT

The authors would like to thank Maxim Yatskovsky and Prof. Oleg Medvedev from Moscow State University for problem statement and helpful comments. We also would like to thank all members of FRUCT Mobile Healthcare WG for fruitful discussion.

REFERENCES

[1] Q. Fang, F. Sufi and I. Cosic, ”A Mobile Device Based ECG Analysis System”, *Data Mining in Medical and Biological Research*, 2008, pp. 209–226.

[2] M. Unser and T. Blu, ”Mathematical Properties of the JPEG2000 Wavelet Filters”, *IEEE Transactions on Image Processing*, Vol. 12, No. 9, 2003, pp. 1080–1090.

[3] Dirac Specification, *http://diracvideo.org/download/specification/dirac-spec-latest.pdf.*

[4] I. Daubechies and W. Sweldens, ”Factoring Wavelet Transforms into Lifting Steps”, *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2004.

[5] W. Polchlopek, W. Maj and W. Padee ”Fast Integer Arithmetic Wavelet Transform Properties and Application in FPGA/DSP System”, *Proceedings of EUSIPCO 2006.*

[6] D. Le Gall, A. Tabatabai, ”Sub-band Coding of Digital Images Using Symmetric Short Kernel Filters and Arithmetic Coding Technique”, *IEEE International Conference in Acoustic, Speech and Signal Processing*, 1988, pp. 761–765.

[7] S. Barua et. al., ”Optimal Quantized Lifting Coefficients for the 9/7 Wavelet”, *Journal of Fourier Analysis and Applications*, Vol. 4, 1998, pp. 247–269.

[8] G. V. Cormack and R. N. S. Horspool ”Data Compression Using Dynamic Markov Modelling”, *The Computer Journal*, Vol. 30, No.6, 1987.

[9] PhysioBank archive of ECG recordings, *http://www.physionet.org/physiobank/database/#ecg.*