Cross-platform ECG Compression Library
for Mobile HealthCare Services

Alexander Borodin, Yulia Zavyalova

Petrozavodsk State University
Department of Computer Science

This research is supported by grant KA179 of Karelia ENPI - joint program of the European Union, Russian Federation and the Republic of Finland

11th FRUCT conference
April 23–27, Saint-Petersburg, Russia
Heart Attack — the leading cause of sudden death in developed countries and Russia

Long-term continuous monitoring — is the way to alarm a doctor timely

Impossible in hospital circumstances and with standard 12-lead ECG monitors

We need simple personal-use devices
Common Mobile HealthCare System Architecture

Cellular Networks and Internet

Emergency tracking

Telemedicine server

Cross-platform ECG Compression Library
Why compression?

Cellular Networks and Internet

Cross-platform ECG Compression Library
The CardioZip library

- Lossless and lossy ECG data compression based on wavelet transform
- The pilot project aimed at growing the competences in Mobile HealthCare area in PetrSU
### The Team

<table>
<thead>
<tr>
<th></th>
<th>07</th>
<th>08</th>
<th>09</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>01</th>
<th>02</th>
<th>03</th>
<th>04</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alexander Borodin</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Yulia Zavyalova</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maxim Obryadin</td>
<td></td>
<td></td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Anna Sapankevich</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Cross-platform ECG Compression Library
Limitations of mobile platforms

- An efficient implementation of the algorithm should be provided since huge and time-consuming computations can quickly discharge the battery.
- Real arithmetics should be avoided if possible due to the lack of floating point unit on some devices.
Preprocessing ECG data with wavelet transform

- Good reputation in 1D and 2D signal compression
  - JPEG 2000
  - Dirac video codec
- Distribution of values for the wavelet coefficients centered close to zero with very few large coefficients
- Effective computation algorithm (lifting schemes)
- Integer-to-integer can be computed with no FPU
Lifting scheme of DWT computation

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

\[
\begin{align*}
  s^{(0)} &= \{x_{2i}\} \\
  d^{(0)} &= \{x_{2i+1}\}
\end{align*}
\]  

(1)

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

\[
\begin{align*}
  d^{(j)} &= d^{(j-1)} - P_j\{s^{(j-1)}\} \\
  s^{(j)} &= s^{(j-1)} + U_j\{d^{(j)}\}
\end{align*}
\]  

(2)
Lifting scheme of DWT computation

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

\[
F_n\{x\} = \sum_{i=1}^{k} f_n(k) x_i
\] (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:

\[
d^{(N)} = M_0 d^{(N)}
\]

\[
s^{(N)} = M_1 s^{(N)}
\] (4)
Lifting scheme for Le Gall 5/3 DWT

\[ s_i = d_i^{(1)} = d_i^{(0)} - \frac{1}{2} (s_i^{(0)} + s_{i+1}^{(0)}) \]

\[ d_i = s_i^{(1)} = s_i^{(0)} + \frac{1}{4} (d_i^{(1)} + d_{i-1}^{(1)}) \]

\[ = -\frac{1}{8} s_{i-1}^{(0)} + \frac{1}{4} d_{i-1}^{(0)} + \frac{3}{4} s_i^{(0)} + \frac{1}{4} d_i^{(0)} - \frac{1}{8} s_{i+1}^{(0)} \]
Lifting scheme for Daubechies 9/7 DWT

\[
\begin{align*}
    d_i^{(1)} &= d_i^{(0)} + \alpha(s_i^{(0)} + s_{i+1}^{(0)}) \\
    s_i^{(1)} &= s_i^{(0)} + \beta(d_i^{(1)} + d_{i-1}^{(1)}) \\
    d_i^{(2)} &= d_i^{(1)} + \gamma(s_i^{(1)} + s_{i+1}^{(1)}) \\
    s_i^{(2)} &= s_i^{(1)} + \delta(d_i^{(2)} + d_{i-1}^{(2)}) \\
    s_i &= \zeta s_i^{(2)} \\
    d_i &= \zeta^{-1} d_i^{(2)}
\end{align*}
\]  

(6)
### Quantized values of Daubechies 9/7 DWT

<table>
<thead>
<tr>
<th>Coefficient</th>
<th>Irrational</th>
<th>Quantized rational</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha$</td>
<td>$-1.58613434\ldots$</td>
<td>$-1.5$</td>
</tr>
<tr>
<td>$\beta$</td>
<td>$-0.0529801185\ldots$</td>
<td>$-0.0625$</td>
</tr>
<tr>
<td>$\gamma$</td>
<td>$0.882911076\ldots$</td>
<td>$0.46875$</td>
</tr>
<tr>
<td>$\delta$</td>
<td>$0.443506852\ldots$</td>
<td>$0.7998046875$</td>
</tr>
<tr>
<td>$\zeta$</td>
<td>$1.14960439\ldots$</td>
<td>$-1.25$</td>
</tr>
</tbody>
</table>
Wavelet transform with no thresholding
Wavelet transform with thresholding
Compressing the coefficients of wavelet transform

- **Dynamic Markov Compression**
  - Moderate speed
  - Good for DWT coefficients
  - There are scalability problems

- **Range Arithmetic Coding**
  - Fast
  - Good compression ratio
Compressing the coefficients of wavelet transform

- Dynamic Markov Compression
  - Moderate speed
  - Good for DWT coefficients
  - There are scalability problems

- Range Arithmetic Coding
  - Fast
  - Good compression ratio
## Lossless compression results

<table>
<thead>
<tr>
<th>Input packet size, bytes</th>
<th>Average output size, bytes</th>
<th>Average CR</th>
</tr>
</thead>
<tbody>
<tr>
<td>300</td>
<td>201</td>
<td>1.49</td>
</tr>
<tr>
<td>1000</td>
<td>562</td>
<td>1.78</td>
</tr>
<tr>
<td>10000</td>
<td>3534</td>
<td>2.83</td>
</tr>
<tr>
<td>25000</td>
<td>8211</td>
<td>3.04</td>
</tr>
<tr>
<td>50000</td>
<td>16120</td>
<td>3.10</td>
</tr>
</tbody>
</table>
Library architecture and overview

- Application
- Qt wrapper classes (optional)
- Input format drivers
- Direct compression API
- Core compression classes
Compressed file format

- Zero Header (4 bytes)
  - CZF in ASCII (3 bytes)
  - Size of memory block in bytes (1 byte)

- File Header
- Integrity Header
- Security Header
- Patient Header
- Algorithms Header
- Compressed ECG Recordings