Wavelet-Based ECG Delineation on a Wearable Embedded Sensor Platform

Nicolas Boichat*, Nadia Khaled†, Francisco Rincon‡§, David Atienza‡
*Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland
†Dept. of Signal Theory and Communications (TSC)-Univ. Carlos III of Madrid, Spain
‡Embedded Systems Laboratory (ESL)-EPFL, Switzerland
§ DACYA-Complutense Univ. of Madrid, Spain.

***Abstract*—The analysis of the electrocardiogram (ECG) is widely used for diagnosing many cardiac diseases. Since most of the clinically useful information in the ECG is found in characteristic wave peaks and boundaries, a significant amount of research effort has been devoted to the development of accurate and robust algorithms for automatic detection of the major ECG characteristic waves (i.e., the QRS complex, P and T waves), so-called ECG wave delineation.**

One of the most salient ECG wave delineation algorithms is based on the wavelet transform (WT). This work is dedicated to the sensible optimization and porting of this WT-based ECG wave delineator to an actual wearable embedded sensor platform with limited processing and storage resources. The porting was successful and the implementation was extensively validated using a standard manually annotated database. Interestingly, our results show that, despite the limitations of the embedded sensor platform, careful optimization allows to achieve comparable or even better delineation results than the original offline algorithm.

I. Introduction

A significant amount of research effort has been devoted to the automated analysis of ECG signals, and the underlying detection of the major ECG characteristic waves, namely the QRS complex, P and T waves [1]. In fact, the performance of an automated ECG analysis system critically depends on the reliable detection of these fiducial waves, so-called *ECG delineation*. This has motivated the rich variety of state-of-the-art ECG delineation approaches [1]. One of the most salient approaches is based on the wavelet transform (WT) [2], [3], mainly because this transform provides a description in the time-scale domain that allows the suitable representation of signals having multiresolution characteristics such as the ECG signal. Indeed, the ECG signal is characterized by a cyclic occurence of patterns at different frequency content (QRS complex, P and T wave).

The WT-based ECG delineation algorithm, first introduced in [2] and further developed in [3], was extensively validated by simulation. Nevertheless, its translation into a robust, efficient and reliable automated analysis capability embedded in a wearable sensor node is unheard of. This translation calls for the porting and (non-straightforward) optimization of this algorithm to adapt it to the sensor node's limited processing resources. In general, this porting and optimization effort is key to translate the recent biomedical signal processing advances into autodiagnosis tools, and hence to enable pervasive healthcare. As a result, the main contributions of this work are the following ones: (1) The transformation of the offline WT-based ECG delineation algorithm of [3] into a real-time implementation, which entails several algorithmic modifications and parameter optimization; (2) The porting and sensible optimization of the new real-time algorithm on the state-of-the-art SHIMMER^{TM} commercial embedded sensor platform [4].

The rest of the paper is organized as follows. In Section II, we illustrate the principle of WT-based ECG delineation. Then, in Section III, we present our novel online WT-based ECG delineation algorithm, and highlight its contributions with respect to the baseline offline version of [3]. Next, in Section IV, we describe our implementation on the SHIMMER^{TM} embedded sensor platform, report the results of its validation on a standard database, and compare the results to the baseline offline algorithm. Finally, we summarize the conclusions of this work in Section V.

II. WT-Based ECG Delineation Principle

As aforementioned, this work is based on the offline multiscale WT-based ECG delineator extensively described and validated in [3]. As such, the therein detailed description and discrete-time implementation of the underlying *dyadic wavelet transform* is not reproduced in the present paper. For self-containment, it is sufficient to mention that the baseline offline ECG delineation algorithm of [3] performs the detection of all characteristic points (onset, peak and end) of the ECG waves using a quadratic spline wavelet transform, which produces *derivatives* of smoothed versions of the input ECG signal at *five dyadic scales* (i.e., 2^1 to 2^5). The choice of these scales was motivated by the observation that most of the energy of the ECG signals lies within these scales [3]. In particular, it was shown that the energy of the QRS complex is very low in scales higher than 2^4, and that the P and T waves have significant components at scale 2^5. These observations are confirmed by Figure 1, which shows an ECG-like signal together with the five first dyadic scales of its WT.

Figure 1 can also be used to illustrate the WT-based ECG delineation principle. As aforementioned, the WT at scale

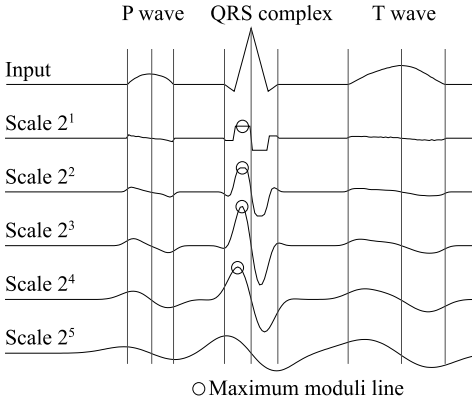

Fig. 1. WT at the first five scale of an ECG-like signal.

2^k is proportional to the derivative of the filtered version of the input ECG signal with a smoothing function at scale 2^k, the zero crossings of the WT correspond to the maxima or minima of the smoothed ECG signal at different scales, and the maximum absolute values of the WT are associated with maximum slopes in the smoothed ECG signal. This is verified in Figure 1, where monophasic waves (R, P and T waves) produce a positive maximum-negative minimum pair across scales, with a zero crossing in between. Moreover, each sharp change in the input ECG signal is associated with a line of maxima or minima across the scales.

Using this information of local maxima, minima and zero crossings at different scales, the WT-based algorithm identifies the fiducial points of the ECG signal, as thorougly explained in the following section.

III. Online WT-Based ECG Delineation

Based on the computed five scales of the discrete wavelet transform (DWT) of the input ECG signal, we perform a multiscale analysis to extract its fiducial points. We first detect the main wave of the QRS complex. Then, we delineate this complex, which consists of detecting its secondary peaks and finding its onset and end. Finally, we detect the P and T waves, along with their respective onset and end.

Our algorithm is based on [2] and [3]. However, we have introduced several modifications and new decision rules that improve its ECG delineation results. In particular, we have adapted the ECG detection thresholds, sampling sizes and search window to enable the implementation of the proposed algorithm on a state-of-the-art commercial embedded sensor platform. We subsequently describe these novel modifications and decision rules. Note that, for all the following, the ECG signal is sampled at a frequency of 250 Hz (i.e., sampling period of 4 ms), which corresponds to the sampling frequency of the used standard database (See Section IV).

A. QRS detection

Like [2] and [3], we detect the QRS main wave by looking for a pair of absolute maximum moduli lines of opposite signs present across scales 2^4 to 2^1, and identifying the main QRS wave as the zero crossing between the maximum and minimum at scale 2^1. More specifically, we search for a pair of maximum moduli lines across scales 2^4 to 2^1 respectively exceeding the predefined thresholds ϵ^4_{QRS} to ϵ^1_{QRS}. This corresponds to searching for a maximum modulus at scale 2^4, then for one in its neighbourhood at scale 2^3, and so on for lower scales. Note that a maximum modulus at scale 2^k is considered to be in the neighbourhood of one at scale 2^{k-1}, if the interval between them is less than 10 samples. In case more than one maximum modulus lies within this interval, we simply select the closest one. Furthermore, the rules in [2] are applied to eliminate redundant lines and to choose the most significant pair of maximum moduli lines. Both positive and negative R waves are allowed, and the peak of the main wave of the QRS complex is found at the zero crossing between the 2 maximum moduli at scale 2^1.

Thresholds.: [3] computes threshold ϵ^k_{QRS} as the root mean square (RMS) value over 2^{16} samples of the DWT at scale 2^k (this RMS value is divided by 2 for scale 2^4). This may be acceptable for an offline implementation. However, for our real-time implementation on an embedded sensor platform (see Section IV-A for more details), only a limited number of samples can be processed at a time. Indeed, our current implementation only keeps in memory 512 samples of the input signal and the 512 samples of its WT-transform on each of the five scales at a time. This number of samples was carefully chosen to achieve the best ECG delineation accuracy, while allowing an online implementation despite the limited memory capacity of our embedded system (cf. Section IV-D). On each scale, the RMS value is computed on each block of 512 samples of the WT transform on that scale, and the thresholds are defined as the average over the last 8 blocks. We then divide the RMS value by 2 for ϵ^4_{QRS}.

A strong advantage of using a threshold computed over a limited amount of time is that the locality of the signal is considered, which is important for our implementation meant to be used in ambulatory conditions, as the noise, heart beat rate and other parameters may change quickly compared to the rather stationary records of the standard database we used (i.e., the QT database [5]).

RR interval.: For some of the following rules, we need to compute an estimate of the current RR interval (i.e. the interval between 2 consecutive R waves), denoted rr in the rest of this document. We use a moving average to perform this estimation:

$$rr_i = \frac{3}{4} \cdot rr_{i-1} + \frac{1}{4} \cdot \begin{cases} \frac{rr_{i-1}}{2} & \text{if } rr_m < \frac{rr_{i-1}}{2} \\ 2 \cdot rr_{i-1} & \text{if } rr_m > 2 \cdot rr_{i-1} \\ rr_m & \text{otherwise} \end{cases}$$

where rr_i and rr_{i-1} are respectively the updated and the old rr estimates, and rr_m denotes the last RR measurement. This allows the estimate to be sensitive to a change in the heart beat rate, without being too vulnerable to false detections, which would in turn produce an avalanche of other false detections due to the search back rule (see below). Additionally, we take into account that the heart beat rate cannot take values above

300 beats per minute ([1], p. 415), or below 20 beats per minute.

Search back.: If no QRS complex is detected for $1.5 \cdot rr$, we look backward for a pair of maximum moduli at scale 2^3, using the same threshold ϵ^3_{QRS}. We then directly look for the zero-crossing point at scale 2^1, without searching for maximum moduli at scales 2^2 and 2^1. This is somehow similar to [2], except for the fact that the threshold ϵ^3_{QRS} is not lowered. This procedure is less restrictive than the regular QRS detection, since we do not additionally check for maximum moduli at scales 2^1, 2^2 and 2^4. The latter modification proved successful to correct the wrong detections suffered by the regular procedure during the atrial fibrillation mode of our patient simulator [6].

B. QRS delineation

Once the main wave of the QRS complex has been detected, we delineate the secondary waves around it. Similarly to the QRS main wave, a secondary wave is characterised by a pair of maximum moduli with opposite signs across scales, and the wave peak is found at the zero crossing at scale 2^1 between these 2 maximum moduli. More specifically, in a search window starting 20 samples (i.e., 80 ms) before the main wave and ending 40 samples (i.e., 160 ms) after it, we look for maximum moduli at scale 2^2 before and after the main wave, respectively exceeding the thresholds $\gamma_{QRS_{pre}}$ and $\gamma_{QRS_{post}}$:

$$\gamma_{QRS_{pre}} = 0.06 \cdot \max(|W_{2^2}x[n]|) \quad (1)$$

$$\gamma_{QRS_{post}} = 0.20 \cdot \max(|W_{2^2}x[n]|) \quad (2)$$

with n within the aforementioned search window.

It is worthwhile mentioning that [3] uses a lower multiplicative factor for $\gamma_{QRS_{post}}$ (i.e., 0.09 instead of 0.20), which performs well with a narrower search window after the main wave. However, we found that our (wider) search window allows to properly delineate a larger variety of morphologies, but requires this threshold increase to avoid detection of insignificant waves. Additionally, [3] suggests that we only take 3 waves into account, so in this implementation, if more than 3 significant waves are detected (i.e., the main wave plus 2 secondary waves), we select the closest ones relatively to the main peak.

Onset and end detection.: The QRS onset and end detection rules of [3] have been completely reworked, since we were unable to reproduce the results therein based on the analysis of scale 2^2. Instead, the QRS onset and end are now detected at scale 2^4. This was chosen by inspecting graphs of the QTDB [5] records for which our first implementation using rules given in [3] performed the worst. We particularly capitalize on the fact that the QRS complex, at scale 2^4, is normally composed of a single pair of maximum moduli, as the signal gets smoothed down at higher scales.

Furthermore, the thresholds were then adjusted experimentally. The onset is detected using the threshold $\xi_{QRS_{on}}$ defined as follows:

$$\xi_{QRS_{on}} = 0.25 \cdot \max(|W_{2^4}x[n]|), \quad n \in [n_{first} - 10;\ R] \quad (3)$$

where n_{first} is the position of the first maximum modulus at scale 2^2 associated with any wave of the QRS complex (as in [3]), and R is the position of the main wave of the QRS complex. Then, we look backward at scale 2^4, starting 2 samples before n_{first}. The QRS onset corresponds to the first sample whose value is below $\xi_{QRS_{on}}$. The QRS end is detected similarly. We find the modulus maximum at scale 2^4 after n_{last} (n_{last} being the position of the last maximum modulus at scale 2^2 associated with any wave of the QRS complex [3]):

$$n_{max} = \arg\max_n(|W_{2^4}x[n]|), \quad n \in [n_{last} + 1;\ n_{last} + 10] \quad (4)$$

and then we define the threshold:

$$\xi_{QRS_{end}} = 0.7 \cdot |W_{2^4}x[n_{max}]| \quad (5)$$

The QRS end is found at the first sample after n_{max} whose value at scale 2^4 is below $\xi_{QRS_{end}}$.

C. P and T waves delineation.

P and T waves detection and delineation follow similar procedures as those of the QRS complex. As suggested in [3], we first look for maximum moduli at scale 2^4 within a search window. [3] also defines some thresholds $\epsilon^4_{P/T}$, which we did not use here. In fact, we realised that these thresholds were often close to 0, and that considering every positive maximum and negative minimum (which is logically equivalent to fixing these thresholds to 0) gives better results in most cases.

After extracting all the maximum moduli, we only consider those with amplitude greater than a threshold $\gamma_{P/T}$, defined in [3] as:

$$\gamma_{P/T} = 0.125 \cdot \max(|W_{2^4}x[n]|) \quad (6)$$

with n within a search window that will be later on defined for each wave specifically. As indicated in [3], if no suitable pair is found at scale 2^4, we repeat the above procedure at scale 2^5. Once the pair of maximum moduli is found at scale 2^k (k = 4 or 5), we select the zero crossing between them at scale 2^3 as the wave peak. If no such zero crossing is found, we use instead the zero crossing at scale 2^k. The onset and end are then detected as indicated in [3], using the same principles as for the QRS onset and end.

P detection.: The search window for the P wave relatively to the QRS complex was not mentioned in [3]. For the first search, we propose to use an initial search window (ISW):

$$ISW = [\text{QRS}_{\text{onset}} - \min(85, \frac{rr}{2});\ \text{QRS}_{\text{onset}} - 2] \quad (7)$$

For every detected P wave, the positions of the maximum moduli, relatively to the R peak, are recorded in 2 estimates pr_1 and pr_2. A moving average with parameters 0.75/0.25 is

used to smooth down these values, allowing us to be flexible but resilient to detection errors. For the subsequent searches, we look for the P wave at scale 2^4 using a narrowed search window (NSW):

$$NSW = ISW \cap [\mathrm{R} - 1.5 \cdot pr_1;\ \mathrm{R} - 0.6 \cdot pr_2] \quad (8)$$

If we do not detect the P wave at scale 2^4, we try again at scale 2^5, using the same window. If no wave is found, the search at scale 2^4 is repeated on a wider window (NSW'):

$$NSW' = ISW \cap [\mathrm{R} - 2.0 \cdot pr_1;\ \mathrm{R} - 0.5 \cdot pr_2] \quad (9)$$

If all the previous measures fail, we assume that the P wave is absent. If, however, multiple maximum moduli pairs are found within the search window, we take the one with the maximum slope

$$\frac{|W_{2^k}x[n_2] - W_{2^k}x[n_1]|}{n_2 - n_1}, \quad (10)$$

n_1 and n_2 being the 2 maximum moduli of the pair. The wave with the maximum slope usually corresponds to the one with the sharpest transition. These rules were found to significantly improve the delineation results when the P wave is faint and hard to distinguish.

T detection.: As for the P wave, the search window was not stated in [3]. We decided to use a search window SW defined as follows:

$$SW = \Big[\mathrm{QRS}_{\mathrm{end}} + 15,\ \mathrm{QRS}_{\mathrm{onset}} + (1.5 \cdot \mathrm{QTc}_{\mathrm{max}} \cdot \sqrt{rr})\Big] \quad (11)$$

Where the values for $\mathrm{QTc}_{\mathrm{max}}$ can be found [7], and correspond to the normal maximum values for the QT coefficient in Bazet's formula:

$$\mathrm{QTc} = \frac{\text{time interval from Q onset to T end}}{\sqrt{\text{Previous RR interval}}} \quad (12)$$

If, within the search window, there are more than one maximum modulus pair at scale 2^4, we choose the one with the maximum slope. However, if the largest slope is less that 1.8 times the second largest slope, we choose the pair with the most important modulus maximum at scale 2^5. This rule allows to discriminate the T wave when a U wave is present, as their impact on scale 2^4 is often very similar.

IV. Implementation and Validation on an Embedded Sensor Platform

In this section, we first present the embedded sensor platform used in this work, and then we descrive how we assess the quality of the different implementations of the ECG delineation algorithm we have developed. Next, we present the experimental results obtained from the validation of reference implementation of the algorithm running on a PC, as well as the results achieved after porting our algorithm on an embedded sensor platform.

A. The SHIMMER™ embedded platform

Our target embedded sensor system is the SHIMMER platform [4]. From the hardware viewpoint, this platform includes a low-power 16-bit microcontroller (i.e, Texas Instrument MSP430F1611), a low-power radio, and an extension module for ECG acquisition. The MSP430 microcontroller runs at 8 MHz, has 10 KB of RAM, 48 KB of Flash, and includes a fast hardware multiplier, but does not include a floating-point unit. Also, SHIMMER provides several I/O peripherals, i.e., a serial port, a Bluetooth radio, analog-to-digital converters (ADC) and ECG signal amplifiers connected to the ADC lines of the MSP430.

From the software viewpoint, we have used the open-source GCC 3.2.3 toolchain for the MSP430 [9] to generate the binaries of our WT-based ECG wave delineator algorithm. GCC uses automatically the hardware multiplier of the MSP430 for multiplication operations, but since the MSP430 does not include a floating-point unit, GCC has to replace many arithmetic operations of our algorithm by software emulation code, which implies a significant performance penalty if the proposed algorithm is simply compiled for the SHIMMER platform. Thus, we performed several implementation optimizations for low-power embedded sensor platforms.

B. Validation procedure

The commonly accepted way to evaluate the quality and precision of a delineation algorithm is to compare its results with manual annotations of cardiologists [1]. Thus, we used the QT database (QTDB) [5], which provides 105 2-lead records sampled at 250 Hz, with manual annotations of QRS complexes, and P and T waves. These annotations span a total of 3600 beats, and feature a wide variety of ECG morphologies. Moreover, as QTDB is widely used in the literature, the results using this database enable direct comparisons with state-of-the-art ECG delineation algorithms [3], [2], [1]. Also, using the ECG acquisition interface of SHIMMER, we have validated our delineation algorithm in ECG waves from a patient simulator [6].

Regarding the specific validation procedure used in our experiments, our algorithm is designed to operate on one lead at a time to compare it with other state-of-the-art ECG delineation algorithms [2], [3]. Therefore, since the manual annotations of the QTDB were performed using information of the 2 leads, for each manual annotation, we selected the automatic annotation on the channel producing less error, as done in [3]. As a result, a manual annotation is considered to be related to an automatic one if their time interval is smaller than 320 ms. Hence, this pair of annotations is a true positive (TP). Then, when 2 annotations on different leads fall within that interval, the closest one is chosen to compute the error, and the other one is not counted as a false positive. Finally, each manual annotation that has no corresponding automatic annotation is counted as a false negative (FN), and automatic annotations without manual annotations count as false positive (FP). Thus, we define the sensitivity (Se) and the positive

TABLE I
RESULTS OF FLOATING POINT, 16-BIT INTEGERS AND REFERENCE PAPER [3] IMPLEMENTATIONS.

Method	Param	P_{on}	P_{peak}	P_{end}	QRS_{on}	QRS_{end}	T_{peak}	T_{end}
this work	Se (%)	99.94	99.91	99.94	100.00	100.00	99.97	99.97
double	P^+_{min} (%)	91.54	92.20	91.54	96.87	96.92	98.91	98.66
PC	$m \pm s$ (ms)	8.1 ± 11.1	10.4 ± 9.0	1.6 ± 10.2	3.9 ± 6.9	3.5 ± 8.3	4.1 ± 12.5	-0.5 ± 16.2
this work	Se (%)	99.87	99.87	99.91	99.97	99.97	99.97	99.97
16-bit int	P^+_{min} (%)	91.98	92.46	91.70	98.61	98.72	98.91	98.50
SHIMMER™	$m \pm s$ (ms)	8.6 ± 11.2	10.1 ± 8.9	0.9 ± 10.1	3.4 ± 7.0	3.5 ± 8.3	3.7 ± 13.0	-2.4 ± 16.9
Martinez [3]	Se (%)	98.87	98.87	98.75	99.97	99.97	99.77	99.77
	P^+_{min} (%)	91.03	91.03	91.03	N/A	N/A	97.79	97.79
	$m \pm s$ (ms)	2.0 ± 14.8	3.6 ± 13.2	1.9 ± 12.8	4.6 ± 7.7	0.8 ± 8.7	0.2 ± 13.9	-1.6 ± 18.1
Tolerances [8]		10.2	-	12.7	6.5	11.6	-	30.6

predictivity (P^+) as:

$$Se = \frac{TP}{TP + FN}, \quad P^+ = \frac{TP}{TP + FP} \tag{13}$$

In addition, due to the QTDB format, it is impossible to know if the lack of an annotation means that the wave is not present or that the cardiologist did not annotate the point with confidence for some reason. In the first case, an automatic detection would count as a false positive, but in the second case, it would not necessarily mean that the detection is erroneous. As a result of this overestimation of false positives, we show in Table I the value of P^+_{min}, which is a lower bound on the real value of P^+.

In our results, the mean of the errors (m) is computed as a summation of the errors on all records, while the standard deviation of the error (s) is defined as the average of the standard deviation on each record.

C. Reference implementation on a PC

The algorithm was first validated on a x86 personal computer (PC), using double-precision floating-point numbers. The obtained results are reported in the first row of Table I, and show that our implementation performs within the standard deviation tolerances indicated by the CSE working party [8] for all points, except the QRS_{on} and P_{on}, for which we are only slightly above the agreed tolerances (less than 7% on average). Furthermore, Table I shows that this implementation of our proposed algorithm improves the standard deviation by up to 4 ms (i.e., one sample) in comparison to the reference state-of-the-art off-line algorithm proposed in [3], while preserving a high sensitivity and predictivity.

D. Optimized implementation on the SHIMMER embedded platform

According to the amount of available memory (10 KB) in SHIMMER, we have restricted the maximum buffer size for the input data and each of the 5 wavelet transforms to 512 samples. Since each sample uses a 16-bit integer (2 bytes), the total amount of data used is 6 KB of RAM. Overall, only 6.5 KB of RAM are used for temporary data in our implementation of the proposed algorithm in SHIMMER, and the instructions require about 15 KB of Flash memory. Thus, making the algorithm suitable for current low-power sensor node platforms.

Since the MSP430 does not have a floating-point unit, the first adaptation we performed to port and run in real-time our ECG delineation algorithm in SHIMMER was the conversion of all the operations from the reference implementation on the PC to use 16-bit signed integers instead of floating-point numbers. Then, the second transformation we applied into the reference implementation was related to the computation of the ϵ thresholds. In the proposed ECG wave delineation algorithm, these thresholds are computed using the RMS of the data blocks, which requires a square root operation. However, the C library `sqrt` function used by the GCC compiler to implement the RMS requires many floating-point operations. Thus, we replaced this function with our own implementation of the Longhand square root algorithm of [10], which performs well on platforms without hardware division support. Overall, we fully avoided all time-consuming software implementations of floating-point operations at the compiler level for our algorithm.

Then, in the third optimization step for the SHIMMER platform, we improve further the real-time performance of our implementation by a simple change in the generated code by the GCC compiler for the MSP430. In particular, the most time-consuming parts of our proposed delineation algorithm are the low-pass and high-pass filters required for the computation of the five scales of the DWT [3], which are called 4 and 5 times, respectively, for each input data sample. However, the multiplications and divisions needed for these filters can be replaced by fast shift and addition operations. Therefore, we generated few lines of assembly code to replace the generated code by GCC from the original C code of our algorithm. This final modification of few lines of code achieved more than 20% improvement in processing time with respect to the default compiled version.

After applying the three previous optimization steps, we used the serial port of the SHIMMER platform to transfer all the records of the QDTB to the platform, and retrieve the delineation results using this modified implementation. The obtained experimental results are shown in the second row of Table I. As this table indicates, our transformations of the original implementation of the algorithm on a PC to run in

TABLE II
REAL-TIME PERFORMANCE OF OUR IMPLEMENTATION.

	Data length	Processing time	Ratio
Worst case	37 s	2161 ms	1/17.1
Best case	67 s	3106 ms	1/21.6
Average	-	-	1/19.5

the SHIMMER platform produce a negligible loss of precision (less than 0.7 ms in terms of standard deviation, i.e, less than 0.2 samples), while enabling an efficient implementation in a low-power embedded microcontroller running only at 8 MHz, and utilizing less than 7 KB of RAM.

In addition, we measured the real-time performance of our algorithm implementation on the SHIMMER platform. To this end, we used an internal timer of the MSP430 to compute the total processing time. This timer is suspended when data transfers are performed through the serial port, and resumed when the data is being processed, as if it was in real-time. Thus, we can extract a ratio between the amount of data to process and the actual time needed to process the data of the ECG waves of the QDTB. Table II shows the average ratio on all records, and the ratio required in the worst and best case, which occured on the first lead of records `sel43` and `sel33` of QDTB, respectively. Interestingly, the obtained processing ratio results using the proposed algorithm and embedded implementation prove that we can perform real-time delineation while keeping the MSP430 microcontroller in sleep mode about 95% of the time on average. Thus, enabling a very low-power embedded ECG wave delineation system.

Finally, to visualize our previous results, we have created a Java application with a graphical user interface (GUI) to control our proposed real-time ECG wave delineation system running on the SHIMMER platform. This interface is able to show on a PC the raw ECG signal received by electrodes placed on a subject's skin or an a patient simulator, as well as the delineation results of our algorithm, which are both sent to the PC using the included Bluetooth support in the SHIMMER platform. A screen-shot of the output of our Java-based GUI with the raw data and the delineation results of our algorithm on the SHIMMER platform are shown in Figure 2.

V. CONCLUSIONS

In this paper, we have demonstrated the feasibility of exploiting the limited processing and storage resources of a state-of-the-art embedded sensor platform to deploy a real-time automated ECG delineator. This goal was achieved through judicious application-level optimizations, taking into account the underlying hardware architecture and the real-time constraints of the automated analysis application. The excellent accuracy of our online WT-based ECG delineator was established using a commonly used standard database. These excellent results show the potential of embedding advanced processing and intelligence in sensor nodes to reduce the amount of wirelessly streamed data, which would ultimately extend the lifetime of the personal ECG monitoring system in which they will be integrated.

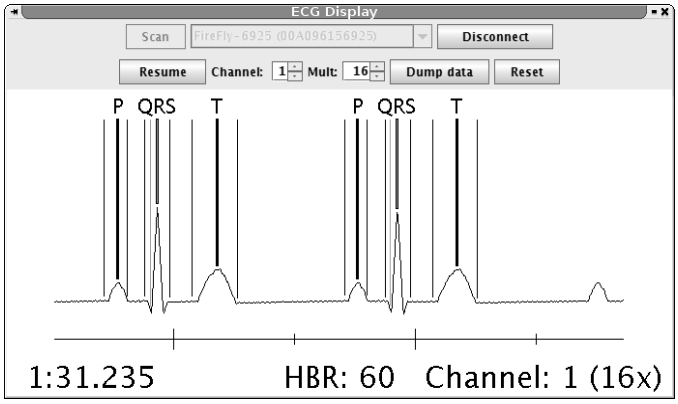

Fig. 2. Graphical interface on host PC displaying the correct ECG delineation results wirelessly streamed from the SHIMMER platform with as input an input signal generated by a patient simulator.

REFERENCES

[1] L. Sörnmo and P. Laguna, *Bioelectrical Signal Processing in Cardiac and Neurological Applications*. Elsevier Academic Press, 2005.

[2] C. Li, C. Zheng, and C. Tai, "Detection of ecg characteristic points using wavelet transforms," *Biomedical Engineering, IEEE Transactions on*, vol. 42, no. 1, pp. 21–28, Jan. 1995.

[3] J. Martinez, R. Almeida, S. Olmos, A. Rocha, and P. Laguna, "A wavelet-based ecg delineator: evaluation on standard databases," *Biomedical Engineering, IEEE Transactions on*, vol. 51, no. 4, pp. 570–581, April 2004.

[4] SHIMMER Research, "Shimmer Small Wireless Sensor Platform Designed to Support Wearable Applications." [Online]. Available: http://shimmer-research.com

[5] P. Laguna, R. Mark, A. Goldberg, and G. Moody, "A database for evaluation of algorithms for measurement of qt and other waveform intervals in the ecg," *Computers in Cardiology 1997*, pp. 673–676, Sep 1997. [Online]. Available: http://www.physionet.org/physiobank/database/qtdb/doc/

[6] Fluke, "PS400 ECG/Arrhythmia Simulator." [Online]. Available: http://global.flukebiomedical.com/busen/products/PS400.htm

[7] L. Schamroth, *The disorders of cardiac rhythm*. Oxford: Blackwell Scientific Publications, 1971.

[8] C. Party, "Recommendations for measurement standards in quantitative electrocardiography," *Eur. Heart J*, vol. 6, pp. 815–825, 1985.

[9] "The GCC toolchain for the Texas Instruments MSP430 MCUs." [Online]. Available: http://mspgcc.sourceforge.net/

[10] T. J. Rolfe, "On a fast integer square root algorithm," *SIGNUM Newsl.*, vol. 22, no. 4, pp. 6–11, 1987.