Real-time QRS detector using Stationary Wavelet Transform for Automated ECG Analysis

Vignesh Kalidas, *Student Member, IEEE* and Lakshman S. Tamil, *Senior Member, IEEE*

***Abstract*— In this paper, we propose an online QRS detector algorithm using Stationary Wavelet Transforms (SWT) for real time beat detection from single-lead electrocardiogram (ECG) signals. Daubechies 3 ('db3') wavelet is chosen as the mother wavelet for SWT analysis. The information from the first ten seconds of the ECG signal is used as a learning template by the algorithm to initialize thresholds for beat detection. These thresholds are then modified every three seconds, thereby quickly adapting to changes in heart rate and signal quality. Hence false beat detections are vastly suppressed in this approach, while identifying true beats with a high degree of accuracy. Our algorithm yields a sensitivity (SE) of 99.88% and a positive predictive value (PPV) of 99.84% on the MIT-BIH Arrhythmia Database, SE of 99.80% and PPV of 99.91% on the AHA database and an SE of 99.97% and PPV of 99.90% on the QT database.**

I. INTRODUCTION

Beat detection is vital to the field of automated cardiac monitoring and acquires further importance in the context of real-time cardiac monitoring. Accurate beat detection algorithms using QRS complex information from Electrocardiogram (ECG) signals result in useful heart rate variability analysis that subsequently leads to accurate detection of cardiac arrhythmias and other abnormalities. This gains special significance in today's world dominated by non-invasive wearable ECG sensors for real-time cardiac monitoring outside of hospitals and other emergency care centers. These sensors are worn by individuals on a continual basis while performing their day-to-day activities. Therefore, there is a high probability that the signals obtained from these sensors are corrupted by external noise, thereby rendering the beat detection process error-prone and thus cumbersome. This external noise can be attributed to, but not limited to, artifacts due to movements necessitated by an individual's routine activities, sensor disconnects, wireless signal transmission interferences, baseline wander, powerline interference, muscle movements, etc., resulting in corruption of vital ECG information. Hence it is necessary to develop algorithms that perform well in such scenarios with minimal false detections. Secondly, it is extremely vital that the algorithm can adapt to varying heart rates exhibited by various cardiac arrhythmias when present. In the presence of arrhythmia, the heart rate is not constant and keeps fluctuating depending upon the type, severity and the number of concurrently occurring arrhythmias. Given these factors, it is imperative that beat detection algorithms are extremely robust to noise without compromising on detection accuracy, especially under arrhythmic conditions.

In literature, various approaches have been proposed for beat detection that generally incorporate a combination of one or more of techniques such as adaptive thresholding [1-4], Hilbert transform [5], multiscale morphological transformation [10], wavelet analysis [6-9], etc. Most the algorithms cited in literature are offline algorithms. Offline algorithms [5-7], have the inherent disadvantage in real-time analysis. Also, most of the offline algorithms work well on longer durations of the signal [8]. Longer segments are more stable to pre-processing steps such as bandpass filtering and normalization and as such, are less susceptible to transient noise and taller T-waves. At the same time, they are not always sensitive to transient changes in ECG morphologies that are characteristic of arrhythmic episodes. This could often result in skipping of arrhythmia or low-amplitude beats. On the other hand, online real-time algorithms are highly suitable for evolving ECG morphologies thus can detect arrhythmic beats more accurately. They have adaptive parameters that usually work well with most signals. The processing and response time is relatively very short as well, of the order of three to five seconds. There is a growing interest in the field of real-time beat detection and hence the need to constantly improve existing methods to continuously improve performance. The methods of [1] and [4] were the earliest published in the field of real-time QRS detection that demonstrated very high detection accuracy. Later, [2] proposed a real-time algorithm using combined adaptive thresholds that included back-search for missed beats, thus achieving improved detection performance. Also, wavelet-based techniques [6-9] started gaining attention for ECG delineation purposes. Wavelet transforms (WT) allow simultaneous time-frequency analysis of a non-stationary signal at different resolutions that makes them suitable for ECG signal analysis. Their ability to capture transient changes effectively makes them a good choice for arrhythmia detection. Most of the wavelet based techniques in literature such as [7-9] use Discrete Wavelet Transforms (DWT) due to their faster computation and decreased redundancy, compared to Continuous Wavelet Transforms [6]. A variant of DWT, namely Stationary Wavelet Transform (SWT) [11-12] has been recently gaining popularity in the field of ECG analysis [6, 13]. SWT is like Discrete Wavelet Transform (DWT) with the exception that there is no decimation in the time domain. Only a dyadic subsampling of scales (frequency domain) is performed. Hence there is translation invariance and lack of resolution loss at lower frequencies, which are

V. Kalidas is with the Electrical & Computer Engineering Department, The University of Texas at Dallas, Richardson, TX 75080 USA (phone: 512-484-1214; email: vignesh.kalidas@utdallas.edu).

L. S. Tamil is with the Electrical & Computer Engineering Department, The University of Texas at Dallas, Richardson, TX 75080 USA, where he is the director of Quality of Life Technology Lab (email: laxman@utdallas.edu).

2471-7819/17/31.00 ©2017 IEEE
DOI 10.1109/BIBE.2017.00083

major bottlenecks of DWT [6, 12]. This increases redundancy in coefficients but the advantage is that the additional artifacts induced due to time-domain subsampling at higher scales are avoided. In this paper, we propose a real-time QRS detector using SWT and adaptive thresholding. Combining SWT with a simpler number of adaptive parameters enables our algorithm to detect QRS complexes with less computational complexity.

We evaluated the performance of our algorithms on the MIT-BIH Arrhythmia database, the QT database [14-16] and the American Heart Association (AHA) database. Our algorithm was written and tested using MATLAB R2017a software package. The run time of our algorithm was less than 10ms per three-second segment when executed on an Intel Core i7-6500HQ processor running at 2.6G Hz.

The paper is organized as follows. Section II provides a brief overview of our approach. Section III explains our algorithm in detail with the necessary equations and threshold settings. The results of our algorithm are reported in section IV. We conclude with a brief discussion on the scope and future improvements of our algorithm in section V.

II. METHOD OVERVIEW

Our algorithm considers the first ten seconds of the signal as the learning stage and successive three-seconds as the detection stages. In our approach, the incoming signal is first resampled to 80 Hz and 2-level SWT is applied to this resampled signal. Daubechies 3 ('db3') wavelet is chosen to be the mother wavelet for computing the SWT coefficients. We found that using a three-second detection stage, along with 'db3' mother wavelet and resampling at 80 Hz for the SWT computation yielded the best performance across multiple signals (obtained from different ECG databases) with varying sampling frequencies.

After computing the SWT, detail coefficients from the second level are extracted. These coefficients are squared and moving window averaging is applied to these coefficients. This step is like the one described in [1]. The resultant signal is normalized between [0, 1]. This normalization enhances the peaks around QRS complexes and reduces the amplitude to almost zero at other locations. The locations of these normalized peaks, therefore, correspond to R-peak locations in the original signal, but with a finite delay. These normalized peaks are used to compute the thresholds for the beat-to-beat intervals and the R-peak amplitudes. These thresholds are then used in the beat detection of the subsequent three-second ECG segments. The algorithm also keeps track of the six most recent beats to compute the standard deviation of the five most recent RR intervals. This information helps in deciding if the signal is currently exhibiting varying heart rate activity or not. This step enables the algorithm to appropriately modify thresholds so that premature beats such as premature ventricular beats (PVCs), atrial fibrillation beats, suddenly on setting tachy-arrhythmias such as ventricular tachycardia, etc. are detected.

The following section describes our algorithm in detail. It should be noted that our algorithm does not detect ventricular fibrillation beats and therefore, such beats have been excluded while evaluating the performance of our method.

III. ALGORITHM DESCRIPTION

The following subsections explain our algorithm along with necessary equations and suitable explanations for different parameter settings. Also, wherever applicable, distinction is made between parameter settings for the learning stage and the detection stage. The first ten seconds of the signal is used as the learning template for initial setting of heart rate and QRS peak amplitudes. Following this, the ECG signal is analyzed periodically every three seconds (detection stages) using thresholds from the preceding stage (can be either the learning segment or the previous three-second detection segment).

A. *Signal Pre-processing*

Prior to performing SWT, the incoming signal is resampled to 80 Hz. This limits the maximum frequency component to 40 Hz per Nyquist criterion. (We assume the signal has an original sampling frequency of at least 80 Hz). We chose 80 Hz as the resampling rate as most of the useful information in an ECG is present in the 0.05Hz to 40Hz range. This choice helps to attenuate the amplitudes of high frequency noise components, thus making the SWT less sensitive to voltage spikes in the ECG. Also, this step reduces the number of sample points for the SWT stage, thereby reducing computation time, making the system more optimal for real-time applications.

B. *Stationary Wavelet Transform computation*

We compute 2-level SWT of the resampled signal using 'db3' as the mother wavelet and extract the level 2 detail coefficients. Level 2 coefficients correspond to the [10, 20] Hz frequency band, which is known to have maximal QRS energy and hence is most suitable for QRS detection [1, 17]. Note that the level-2 detail coefficients thus obtained have a sampling frequency of 80 Hz. We resample these coefficients back to the original sampling frequency of the input signal for further analysis.

C. *Squaring and Moving Window Averaging*

After resampling the detail coefficients back to the original signal frequency, they are squared sample-wise. Following this, moving window averaging (MWA) is performed to enhance peaks around QRS complex locations and to attenuate rest of the sample points. These steps are same as the squaring function and moving-window integration steps described in [1]. For the MWA step, the window duration is the preceding 0.15s of the squared data. The MWA signal is then normalized between 0 and 1.

D. *R-peak Detection*

The normalized MWA signal is analyzed for determining the locations of the QRS complexes. They are explained in detail in the following subsections.

Initialization and Definitions

1. *Fs:* Original Sampling frequency of the incoming ECG signal (e.g. 360 Hz for MITDB).

   pk_mwa: Initially empty vector for storing peak locations from normalized MWA signal.

amp_vec: Initially empty vector for storing peak amplitudes from normalized MWA signal.

ppi_vec: Initially empty vector for storing peak-to-peak intervals from normalized MWA signal.

amp_thr: Peak amplitude threshold for identifying peaks in the MWA normalized signal. Initialized to 0.25 for learning stage.

ppi_thr: Minimum separation threshold between two consecutive peaks. Initialized to (0.2*Fs) samples i.e., 200 milliseconds for learning stage since physiological constraints require two heartbeats to be spaced at least 200 milliseconds from each other temporally.

missed_ppi: Minimum separation threshold between consecutive peaks for identifying missed beats.

sd_rr5: Standard deviation of five most recent RR intervals. Initialized to 0.00 milliseconds for learning stage.

r_vec: Initially empty vector for storing actual R-peak locations.

Threshold-based peak detection

2. The normalized MWA signal is scanned to identify peaks with minimum amplitude of *amp_thr* units and separated by at least *ppi_thr* samples. These peaks are added to *pk_mwa* vector.
3. Update *amp_vec* with amplitudes of the above peaks.
4. Update *ppi_vec* with peak-to-peak interval (PPI) values computed from these peaks.

Missed Beat Detection

5. Determine intervals from *ppi_vec* that exceed a pre-defined number of samples, *missed_thr*, computed as follows:

   if ($sd\_rr5 > 0.1*Fs$)

   $$missed\_thr = (1.25 * ppi\_thr) \quad (1a)$$

   else

   $$missed\_thr = (1.5 * ppi\_thr) \quad (1b)$$

   Equation (1a) corresponds to a scenario where the current heart rate is not constant and therefore the standard deviation of the five most recent RR intervals is greater than 100 milliseconds. This can be mostly attributed to the presence of recurring PVCs or other premature beats in the preceding segments. Hence the *missed_thr* is lower than when the heart rate variation is significantly less as in (1b). The value of 100 milliseconds was chosen experimentally for this step.
6. Scan each interval found in step 5 for peaks in a manner like step 1, but with a minimum amplitude of 0.1 units and a minimum separation *missed_ppi*, computed as follows:

   $$missed\_ppi = \max(0.75*ppi\_thr, (0.2*Fs)) \quad (2)$$

7. Update *pk_mwa, amp_vec* and *ppi_vec* vectors appropriately with the new peaks found in step 6

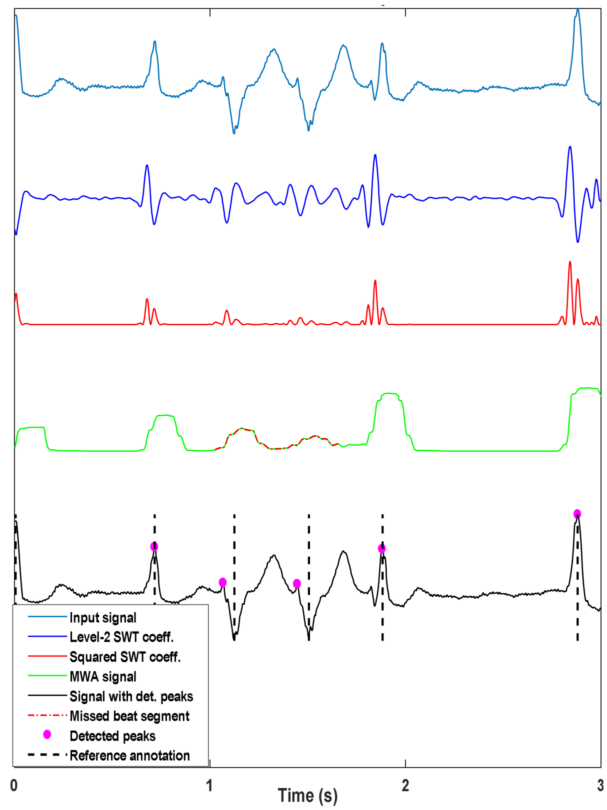

Figure 1. Detection of Ventricular Tachycardia (segment from 1s to 2s) from record 203 (MITDB) because of missed beat detection steps. The missed segment is denoted by dotted red line on the MWA signal.

Update thresholds

8. Update thresholds to be used for the subsequent detection stages as follows:

   Learning stage update

   $$amp\_thr = 0.4*(amp\_thr + \min(amp\_vec)) \quad (3a)$$

   $$ppi\_thr = 0.75*(ppi\_thr) + 0.25*(\mathrm{median}(ppi\_vec)) \quad (3b)$$

   Detection stage update

   $$amp\_thr = 0.4*(amp\_thr + \min(amp\_vec)) \quad (3c)$$

   $$ppi\_thr = 0.4*(ppi\_thr + \min(ppi\_vec)) \quad (3d)$$

   We use different settings for changing the PPI thresholds in the learning and detection stages respectively. This is done since the detection stage has a shorter duration and hence using the minimum of PPI values instead of their median for threshold computation resulted in better detection of ventricular tachycardia and other premature beats.

Actual R-peak location determination

9. For each peak location pk_i in *pk_mwa,* we compute the corresponding R-peak's actual location as the

index of the maximum peak in the normalized ECG signal within the previous 0.10s, measured from pk_i.

10. If no peaks are found in this window, we choose the index with the maximum amplitude in the ECG signal as the actual R-peak location.
11. Update *r_vec* with the actual R-peak locations.
12. Determine the RR interval values from the six most recent R-peak locations in *r_vec.*
13. Update *sd_rr5* using the RR intervals from step 12.
14. Use the updated parameters from step 8 and step 13 to detect R-peaks from the next three seconds of ECG data.

These above steps summarize our algorithm. These steps are implemented for successive three-second segments to determine their corresponding QRS complex locations.

IV. RESULTS

We evaluated our algorithm on the MIT-BIH Arrhythmia database (MITDB), AHA database and QT Database (QTDB). We achieved a sensitivity of 99.88% and a positive predictive value (PPV) of 99.84% on the MITDB database, 99.80% sensitivity and PPV of 99.91% on the AHA database and a sensitivity of 99.97% and PPV of 99.90% on the QTDB database. There are 48 records in the MITDB database, with the ECG signals sampled at 360 Hz. From the AHA database, we tested our algorithm on the first 70 records, sampled at 250 Hz. The QT database is also sampled at 250 Hz and has 65 records with reference annotations.

TABLE I. PERFORMANCE OF OUR ALGORITHM

Database (total beats)	*Sensitivity*	*PPV*	*Accuracy*	*Mean Error (# of samples)*
MITDB (109494)	99.88%	99.84%	99.73%	5.55ms (~ 2 samples)
AHA (161190)	99.80%	99.91%	99.70%	30.00ms (< 8 samples)
QTDB (86995)	99.97%	99.90%	99.87%	21.04ms (< 6 samples)

We used the signals from Lead I in each of the three databases, resulting in a total beat count of 109494, 161190 and 86995 beats for the MITDB, AHA and QT databases respectively. The results are tabulated in Table I. The table displays the sensitivity, PPV and accuracy. It also shows the mean-error which is the average difference between the beat-annotations of our algorithm and the reference beat annotations. Table II compares the performance of our method with some of the other published methods for beat detection. Our algorithm shows improved performance when compared to other real-time methods and comparable performance when compared to off-line methods.

TABLE II. COMPARISON WITH OTHER METHODS (MITDB DATABASE)

Algorithm	*Se (%)*	*PPV (%)*	*Remarks*
Our method	99.88	99.84	Real-time, use SWT
J.Pan et.al [1]	99.75	99.54	Real-time, digital filters, adaptive thresholds, no WT
P.S.Hamilton et.al [4]	99.69	99.77	Real-time, digital filters, adaptive thresholds, no WT
I.I.Christov [2]	99.74	99.65	Real-time, adaptive thresholds, no WT
M.Merah et. al [6]	99.84	99.88	Offline, use SWT
C.Li et al [8]	99.90	99.84	Offline, use WT, excluded records 214 and 215
J.P.Martinez et. al [7]	99.80	99.86	Offline, use DWT

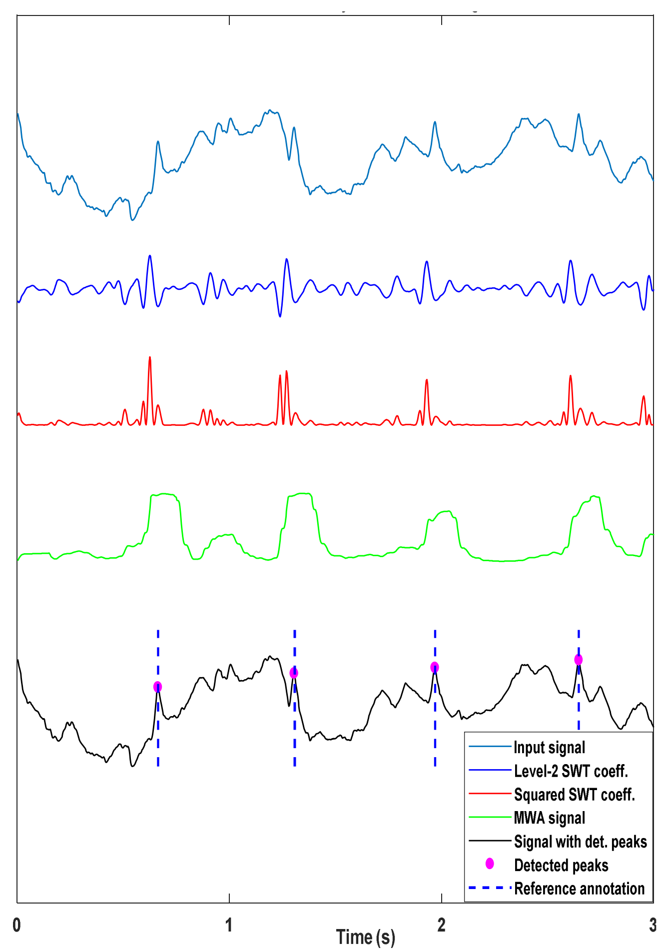

Figure 2. Detection of QRS complexes in a 3s noisy segment. (Record 105 from MITDB database).

V. CONCLUSION

In this paper, we have proposed an algorithm based on stationary wavelet transform for real-time detection of QRS complexes in an ECG signal. Our algorithm performs well in the presence of noise as well as under different arrhythmic conditions. We have used three different databases containing records with multiple arrhythmias to demonstrate this. Fig. 1 shows detection of ventricular tachycardia because of missed beat detection steps described in Section III. Fig. 2 shows the detection of beats in a noisy environment and Fig. 3 shows detection of low amplitude beats in the presence of unusually large ventricular tachycardia beats.

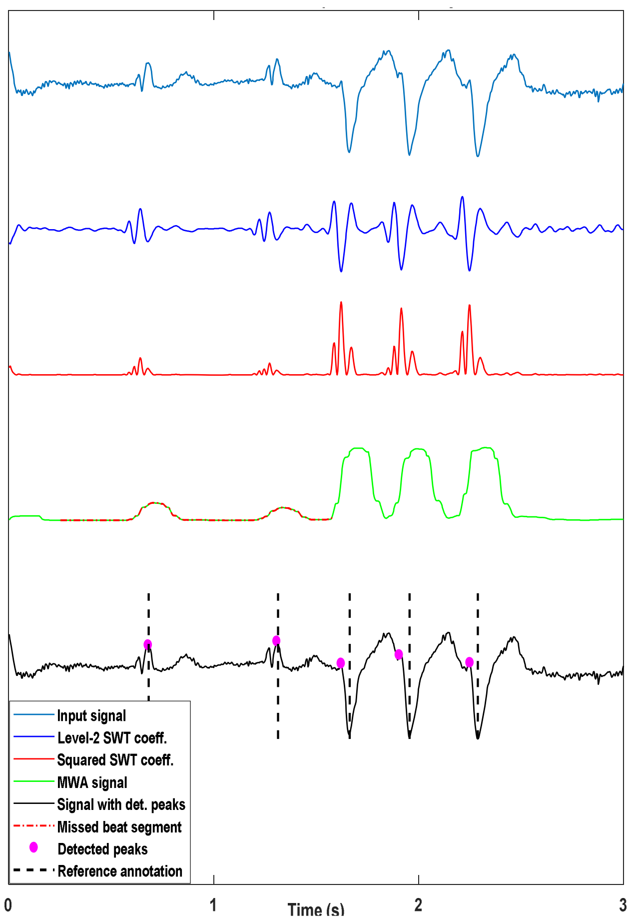

Figure 3. Detection of low amplitude QRS complexes (from 0s to 1.5s) from record 203 in the presence of unusually large ventricular tachycardia beats (from 1.5s to 2.5s).

Our approach requires the first ten seconds of a new signal to be fairly clean as it is used as the learning template. Otherwise, there could be a slight degradation in the algorithm's performance. To comply with this constraint, we should measure the quality of the signal and pick only those that are good for the ten seconds learning segment. The algorithm described in [6] uses SWT and shows similar performance to our approach but our method has the advantage that it can perform real-time online QRS detection with the same degree of accuracy. This is an indication that SWT can be a useful tool in the future for improving real-time ECG beat detection performance.

REFERENCES

[1] J. Pan and W. J. Tompkins, "A real-time QRS detection algorithm," *IEEE Transactions on Biomedical Engineering*, vol. BME-32, no. 3, pp. 230-236, Mar. 1985.

[2] I. I. Christov, "Real time electrocardiogram QRS detection using combined adaptive threshold ," *BioMedical Engineering OnLine,* vol. 3(1):28, Aug. 2004.

[3] D. Lai, F. Zhang and C. Wang, "A real-time QRS complex detection algorithm based on differential threshold method," *2015 IEEE International Conference on Digital Signal Processing (DSP)*, Singapore, 2015, pp. 129-133.

[4] P. S. Hamilton and W. J. Tompkins, "Quantitative investigation of QRS detection rules using the MIT/BIH Arrhythmia Database," *IEEE Transactions on Biomedical Engineering*, vol. BME-33, no. 12, pp. 1157-1165, Dec. 1986.

[5] D. S. Benitez, P. A. Gaydecki, A. Zaidi and A. P. Fitzpatrick, "A new QRS detection algorithm based on the Hilbert transform," *Computers in Cardiology 2000. vol.27 (Cat. 00CH37163)*, Cambridge, MA, 2000, pp. 379-382.

[6] M. Merah, T. A. Abdelmalik and B. H. Larbi, "R-peaks detection based on stationary wavelet transform," *Computer Methods and Programs in Biomedicine,* vol. 121, no. 3, pp. 149-160, Oct. 2016.

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

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

[9] M. Bahoura, M. Hassani and M. Hubin, "DSP implementation of wavelet transform for real time ECG wave forms detection and heart rate analysis," *Computer Methods and Programs in Biomedicine,* vol. 52, pp. 35-44, 1997.

[10] Y. Sun, K. L. Chan and S. M. Krishnan, "Characteristic wave detection in ECG signal using morphological transform," *BMC Cardiovascular Disorders,* vol. 5(1):28, Sep. 2005.

[11] J. C. Pesquet, H. Krim and H. Carfantan, "Time-invariant orthonormal wavelet representations," *IEEE Transactions on Signal Processing*, vol. 44, no. 8, pp. 1964-1970, Aug 1996.

[12] G. P. Nason and B. W. Silverman, "The stationary wavelet transform and some statistical applications," Springer-Verlag, 1995, pp. 281–300.

[13] S. Li and J. Lin, "The Optimal De-noising Algorithm for ECG Using Stationary Wavelet Transform," *2009 WRI World Congress on Computer Science and Information Engineering*, Los Angeles, CA, 2009, pp. 469-473.

[14] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "Physiobank, physiotoolkit, and physionet," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000. [Online]. Available: http://-circ.ahajournals.org/content/101/23/e215

[15] G. B. Moody and R. G. Mark, "The impact of the MIT-BIH Arrhythmia Database," *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45-50, May-June 2001.

[16] P. Laguna, R. G. Mark, A. L. Goldberger and G. B. Moody, "A database for evaluation of algorithms for measurement of QT and other waveform intervals in the ECG," *Computers in Cardiology 1997*, vol. 24, Piscataway, NJ, 1997, pp. 673-676H.

[17] M. Elgendi, M. Jonkman and F. De Boer, "Frequency bands effects on QRS detection," *Proc. Internaltional Joint Conference on Biomedical Engineering Systems and Technologies, BIOSIGNALS 2010,* Valencia, Spain, 2010, pp. 428-431.